STRUCTURE LEARNING FRAMEWORK BASED ON MUTUAL COHERENCE IN BELIEF NETWORK

by

Muhammad Naeem
PC103002

A PhD Dissertation submitted to the
Department of Computer Sciences
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

Faculty of Computer Science
Mohammad Ali Jinnah University
Islamabad
May, 2013

ACKNOWLEDGMENT

The period I used up on my PhD work was the most challenging, exigent and demanding, but at the same time very appealing and rewarding time during research work. I dealt with a range of research problem in the field of structure learning under the umbrella of data mining intrinsic solution approaches. Through the arduous voyage towards the pursuit of highest educational career in my life, it would have been unimaginable without the assistance of many individuals. I would like to express my gratitude for my advisor Dr. Sohail Asghar for providing me full freedom to follow up on my research interests. He has imparted to me the skill of thinking deeply around the selection of problems worth solving. He taught me how to illustrate the ideas cogently and concisely. In fact, I chose data mining as a research area because of his lectures during my MS program. I particularly thank him for disbursing his wealth of knowledge to furnish me with brilliant ideas during our weekly grouped meetings at the platform of the Center of Research in Data Engineering (CORDE). His way of promptness in providing me cynical yet useful comments on research articles will always be much appreciated. If I have the privilege to involve in teaching profession, I will let myself strive to emulate him.

There are also some others who have shown keen concern in my intellectual growth, particularly Dr. Abdul Qadir. He has been a terrific graduate counselor, providing me with awesome advice during course work and for that, I am extremely thankful. Furthermore, I am thankful for the financial support provided by Higher Education Commission Islamabad Pakistan under the program of indigenous scholarship scheme for MS leading to PhD. I am thankful to my mother, brother and sisters. In the last but not least, I would express my heartiest gratitude towards my beloved wife who always stand by me during the pursuit of my educational objective; without her continuous support, it was not possible for me to achieve a lot of milestones in my graduate career.

ABSTRACT

In classification, structure prediction from the Bayesian inference model is a highly symbolic formalism for the purpose of retrieving hidden rules in pragmatic situations. Although there are numerous classes of models representing uncertainties; however Bayesian Belief Network (BBN) is the only model explicitly dealing with direct statistical cause-effect relationship based on an established theory of probability. This process comprises of two sequential steps broadly. First step deals with the construction of best suitable structure. The second step is oriented towards parameter learning for the sake of the inference drawn from this structure. In this study, the focus is on the completeness of the first part. We have highlighted some issues related to the hyper-parameters encompassing feature selection versus feature ordering and scoring function which rests at the heart of structure learning. The originality and contribution of this study is bifurcated into two phases.

In the first phase, we have introduced parameter free, decomposable, penalty less factor Non Parametric Factorized Likelihood Metric (NP-FiLM). The data fitting of some of existing scoring metrics are characterized by parameter of external penalty factor; where unfortunately, it is not possible to correctly identify most appropriate penalty factors a prior. On the other hand, some scoring metrics are not potent enough to exhibit balance between overfitting and underfitting of the learnt model. The proposed scoring metric has its root in information theoretic elucidation. The metric is devised to maximize the discriminant function for query variables with respect to the class and other non class variables. We empirically evaluated the proposed metric over an abundant number of natural datasets (fifty UCI dataset). The comparison is made with respect to ten tree classifiers, one regression model and two neural network system. Furthermore, the scoring metric has been examined to six peers scoring metric within the greedy search and hill climbing searching mechanism as well. NP-FiLM oriented BBN have been satisfactory found with significant results in a paradigm of classification accuracy with the capability of illustrating the best possible data fitting model in context of hyper-parameters described above.

In the second phase, we have presented an information theoretic criterion Polarity Measure (PM) which is quite useful for feature order sensitive classifiers such as

Random Forest (RF) and BBN employing greedy algorithm K2. Both of these classification systems have been shown to be sensitive to the initial ordering of the features. We have illustrated that improvement in classification can be obtained even without ceding variables in case of feature (attribute) ranking sensitive classifiers. We also performed a comparison between BBN and RF classification approaches in the well known feature subset selection and feature ranking problem. The PM measure is devised directly from well renown objective function: conditional likelihood. It posses the capability to discover the degree of explanation made by one feature (attribute)'s state to explain the other feature's state. The technique has significantly better well performed in BBN and better in RF in comparison to five feature ranking techniques and three well established feature subset selection techniques. The proposed measure PM is quite tractable over large dimensional search spaces with low computational complexity.

Another contribution of this study includes a practical application of structure learning to decision support system for settlements in labor negotiations system and identification of genotype in HCV sequences, where a model learned from the dataset is used to yield swift approximation to counting queries, surpassing in certain aspects other peer state-of-the-art techniques to the same problem.

AUTHOR'S PUBLICATIONS

RELEVANT TO THESIS

1 Naeem M. and Asghar S., A Novel Mutual Dependence Measure for Structure Learning, *Journal of the National Science Foundation of Sri Lanka*, Impact Factor Impact Factor 0.232 (accepted on 26 April 2013)

2 Naeem M. and Asghar S., Scientific Study of Religion in Vexillology, *European Journal of Science and Theology*, Impact Factor Impact Factor 0.6 (accepted on 3 May 2013)

3 Naeem M. and Asghar S., An Information Theoretic Scoring Function in Belief Network, *The International Arab Journal of Information Technology*, (Accepted, In press for vol.11 (5)) (ISI Impact Factor 0.127)

4 Naeem M. and Asghar S., A Novel Feature Selection Technique For Feature Order Sensitive Classifiers, Anale. Seria Informatica. Annals. Computer Science Series, Tome 11, Fasc. 1 (To be published in June 2013 Issue) http://anale-informatica.tibiscus.ro/?page=11_numarcurent&lang=en

SUBMITTED

5 Naeem M. and Asghar S., A Review of Feature Selection Techniques in Structure Learning, Journal of Engineering & Applied Science, ISSN: 1023-962X. (1st revision) (HEC X-Category)

6 Naeem M. and Asghar S., Parameer Free and Non Penalized Scoring Metric for Bayesian Belief Network, Journal of Applied Research and Technology, (1st Revision) Impact Factor: 0.14

OTHER PUBLICATIONS

7 Naeem M. Gillani S. Qadir M. A. and Asghar S., gSemSim: Semantic Similarity Measure for Intra Gene Ontology Terms, *International Journal of Information Technology and Computer Science (IJITCS)*, ISSN (print): 2074-9007, ISSN (online): 2074-9015, Vol. 5, No. 6, May 2013, Pp. 32-40.

8 Kanwal A., Fazal S., Asghar S. and Naeem M., Linkage Analysis and Subgroup Discovery of Mody Genes from Text Documents, *The Professional Medical Journal*, ISSN: 1024-8919, eISSN: 2071-7733. (Online) http://www.theprofesional.com/ (Paper accepted and In Print), *(HEC Y-Category)*

9 Naeem M., Naeem M. and Asghar S., Knowledge Discovery in Metabolic Pathways, *International Journal of Bio-Science and Bio-Technology*, Accepted for Vol. 5, No. 3, June, 2013. (indexed in scopus)

10 Naeem M., Gillani S. and Asghar S., Application of Subset Theory towards Solution of Functional Diversity Paradox, *International Journal of Hybrid Information Technology*,Vol. 6, No. 2, March, 2013, Pp. 107-116.

11 Naeem M. and Asghar S., "Knowledge Discovery in Endangered Species Diversification" *International Journal of Information Technology and Computer Science (IJITCS)* 5, no. 2 (2013). 57-65.

12 Naeem M., Gillani S. and Naz S., MfWMA: A Novel Web Mining Architecture for Expert Discovery, *International Journal of Advanced Science and Technology,*Vol. 52, March, 2013, Pp. 45-60.

13 Kanwal A., Fazal S., Asghar S. and Naeem M., Subgroup discovery of the MODY genes from text documents, The Professional Medical Journal, ISSN: 1024-8919, eISSN: 2071-7733. (Online) http://www.theprofesional.com/ (Paper accepted and In Print) *(HEC X-Category)*

14 Naz S., Naeem M. and Qayyum A., "Performance Evaluation of Index Schemes for Semantic Cache" *International Journal of Information Technology and Computer Science (IJITCS)* 5 , no 4 (2013). 40-46.

15 Naeem M., Naz S. and Gillani S., "A Framework for Real-Time Resource Allocation in IP Multimedia Subsystem Network" *International Journal of Computer Network and Information Security (IJCNIS)* 5, no. 3 (2013): 32-38.

16 Naeem M., Khan M. B. and Afzal M. T., Expert Discovery: A web mining approach, Journal of Artificial Intelligence & Data Mining, Vol. 1 (2013), Pp. 35-47.

17 Gillani S., Naeem M., and Habibullah R. and Qayyum A., Semantic Schema Matching Using DBpedia, *I.J.Intelligent Systems and Applications,* 2013, Volume 5, Issue 04, pp. 72-80.

18 Naeem M., Naz S. and Gillani S., QoS Guarantee for VOIP over Wireless LANs, *International Journal of Hybrid Information Technology,* Accepted and in press for Vol. 6, No. 3, May, 2013.Pp.xxx-xxx

19 Naz S., Naeem M., Afza M. T., Qayyum A., "Expertise identification and visualization", *8th International Conference on Computing and Networking Technology (ICCNT), 2012,* pp. 16-20. IEEE, 2012.

20 Gillani S., Naz S., Naeem M., Afzal M. T., Qayyum A., "ERRAGMap: Visualization Tool" *8th International Conference on Information Science and Digital Content Technology (ICIDT), 2012,* vol. 3, pp. 736-740. IEEE, 2012.

21 Naeem M. and Asghar S., KDSSF: A Graph Modeling Approach. *International Journal of Computer Applications* 33(4):31-37, November 2011. Published by Foundation of Computer Science, New York, USA

22 Naeem M., Asghar S., Fong S., Hiding Sensitive Association Rules Using Central Tendency, *2nd International Conference on Data Mining and Intelligent Information Technology Applications (ICMIA 2010),* 30 Nov - 2 Dec 2010, Seoul, Korea, pp.478-484 (IEEE-Xplore)

23 Naeem M., Sohail A., Irfan S. R. and Simon F., "Multilevel classification scheme for AGV perception" In *Advanced Information Management and Service (IMS), 2010 6th International Conference on,* pp. 485-489. IEEE, 2010.

24 Naeem M., Asghar S., A Novel Architecture for Hiding Sensitive Association Rules. Proceedings of the 2010 International Conference on Data Mining (DMIN 2010). Las Vegas, Nevada, USA. July 12-15, 2010. CSREA Press 2010. ISBN: 1-60132-138-4, Robert Stahlbock and Sven Crone (Eds.)

25 Naeem M. and Jamal T., Remote Eye of LAN, Journal of Natural Sciences & Mathematics, Government College University Lahore Pakistan, PK ISSN 0022-2941 CODEN JNSMAC Vol. 43, No.2 (October-2003) Pp 187-192

SHORT/POSTER PAPER

- Naeem M., and Asghar S., gMean: Graph Clustering in Metabolic Pathways, National Science Conference: Univ. of Arid Agricluture Rawalpindi Pakistan, Jan 10-12, 2012. Pp 146 (Short Paper)
- Naeem M., Asghar S., Endangered Species Diversification to Integration: A Graph Modeling Approach, National Science Conference: Univ. of Arid Agricluture Rawalpindi Pakistan, Jan 10-12, 2012. Pp 143 (Short Paper)
- Naeem M. And Asghar S., Bayesian Inference from Metabolic Pathways, National Science Conference: Univ. of Arid Agricluture Rawalpindi Pakistan, Jan 10-12, 2012. Pp 302 (Poster Presentation)
- Naeem M., and Asghar S., KEGG Metabolic Reaction Network (Undirected), UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/KEGG+Metabolic+Reaction+Network+%28Undirected%29, (Dec, 2011)
- Naeem M., “Statistical View of Data in a Clustered bar chart”, 2004, http://www.devarticles.com/c/a/PHP/Statistical_View_of_Data/

TABLE OF CONTENTS

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BBN	Bayesian Belief Network
BF	Best First
BIC	Bayesian Information Criteron
BDeu	Bayesian Dirichlet for likelihood-equivalence for uniform joint distribution
CFS	Correlation based Feature Subset
CLS	Classifier Subset
CNS	Consistency Subset
fCLL	Factorized Conditional Log Likelihood
FSS	Feature Subset Selection
GS	Greedy Search
HCL	Hill Climbing
K2	Kutato Two
MDL	Minimum Description Length
NP-FiLM	Non Parametric Factorized Likelihood Metric
PM	Polarization Measure

Chapter 1

STRUCTURE LEARNING

1.1 Overview

For the last two decades, the evolution of the internet and web enabling technologies have generated massive amounts of electronic data which is a significant advancement in the field of information technology. In fact, the data generation never goes at eternal rest or even slumbering state. In every second, large amount of data is emerging out of application across the web. Amidst this data generation, a simple query arises; is these piles of data convertibility to useful information model. Researchers realized that there are intelligent techniques which can lead to the discovery of useful information (pattern) from within this data. Although statistical techniques were already in practice but there was a need to go beyond the conventional statistical measures. Robust heuristics were required to distill patterns automatically from an avalanche of data. This leads to the emergence of machine learning and data mining as a branch of computer science. This study investigates a notable machine learning technique known as a Bayesian Belief Network (BBN) which has its notion towards structure learning of the data. This study analyzes and evaluates the core functionality and components of BBN over the benchmark dataset with a solid theoretical contribution towards the field of structure learning. Through the whole of this study, we have used the terms variable/feature and scoring function/scoring metric as interchangeable with each other.

A great deal of research has been observed with its focus on structure learning from data (Buntine, 1996; Heckerman, 2008). Bayes belief networks (BBN) have proven their robustness and efficiency in decision and reasoning under uncertainty for inference tasks in the notion of their carrier of structural and qualitative information about the domain (Guo and Schuurmans, 2012). In BBN, structure learning has been addressed in two approaches; constrained based and scoring function inspired approaches. The latter technique is more popular and intractable as compared to the first one (Guo and Schuurmans, 2012). The scoring function based approach which is essentially based on well established statistical principles, the whole structure is evaluated in terms of a score, the better the score, and the more reliable the network structure is. The score of the network in other words reflects how well the structure fits the underlying data; thus scoring function provides a pivot towards optimized structure learning. We can group Naïve Bayes classifiers from various dimensions.

- Firstly, some techniques are aimed towards building “correct structure”. Here the correct structure indicates the structure built on domain expert knowledge establishing the actual connections between node variables. Simple restrictive assumption such as NBC also falls under this category. It is reported that restrictive network accentuate the quality of the learnt network where the final structure is in high agreement to the drawing of domain experts. This approach is typically restricted in the complex domain with a disclosure of the high degree of dependence among data attributes. Lamma, Riguzzi and Storari (2004; 2005) presented two system K2-Chi squares and K2-Lift. They used three benchmark network, Asia, Alarm and Boelarge92. They show that their system can draw the structure with “minimum” error of connection between nodes in the structure. A serious concern about this approach is its biasness and time consuming manual effort.
- Contrary to this approach, the other approach is to let the intelligent heuristics decide on the basis of probability or possibility theory to place the positions of the nodes and setting the nodes amongst them. There are vantages to structure learning a Bayesian Network straight from the dataset. Employing an initial ordering of the query variables, a search algorithm is chosen such as greedy search algorithm, then a metric (also known as scoring function) is used to maximize the overall value of the complete network till all of the variables in input bin are exhausted. K2 algorithm is a notable example in this regard. This approach is primarily focused on improvement of the class imbalance characteristics of learnt structure during parameter learning. Second dimension is feature selection which can be further trifurcated into three sections. Feature Subset selection, Feature Ordering and clustering. The third dimension is improvement in searching algorithm. The fourth dimension is the introduction of a new scoring function which is thrived during the whole searching mechanism.

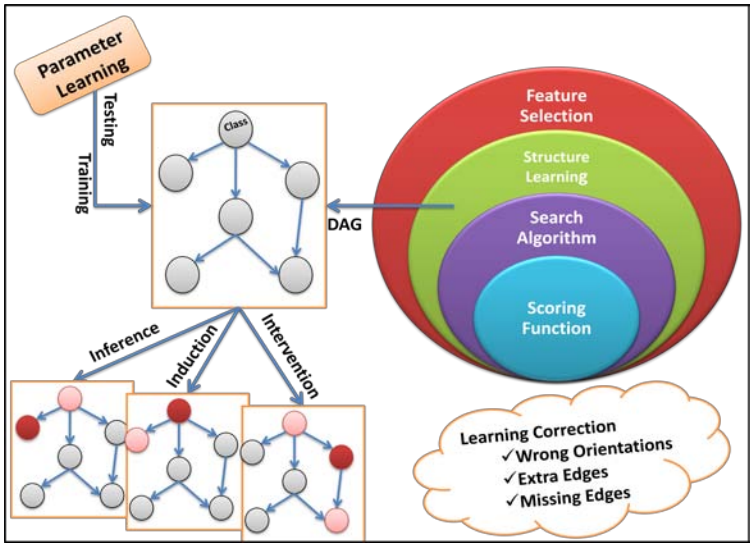


Figure 1.1: Structure Learning: A Broader Picture

The figure 1.1 is delineating the whole picture of the structure learning for which we shall describe each and every component of this framework shown in the figure 1.1. The remainder of chapter one is structured into the following sections. Section 1.2 to section 1.7 introduces the core part of structure learning concept within the framework of machine learning. These concepts are quite useful for understanding the functionality of BBN. Section 1.8 and 1.9 are concerned with the usability of the learnt structure whereas the remaining sections highlight optimization tactics of structure learning. In the last sections the motivation and objectives of this study have been forwarded.

1.2 Expert-Based Structure

Expert based structure learning has its withstanding in its natural tendency to be adapted with an intuitive way to build a structure. The domain experts relied on marking the variables followed by identification based on the degree of their mutual dependence and independence. However, the expert knowledge is an implicit characteristics; thus the identification of degree of dependence always varies from expert to expert due the level of domain knowledge an expert can have. This surely arises some issues such as difficulty to hire the services of an expert, inaccuracy of the structure due to the underlying misrepresentation of the data in perspective of a continuously changing environmental conditions. Moreover the last but not least, it is time consuming due to its manual nature of collection of meta information for the dataset. These reasons pushed back the application of expert based knowledge towards small scale dataset involving a few variables. Another limitation in expert based structure is the target of achieving high accuracy. In building of the expert based structure, It is not essential to obtain a better classification accuracy in comparison to learned from data structure technique. de Campos and Castellano (2007) presented a study in which structural restrictions affecting the learning BN was carried out. These restrictions include existence of arcs between nodes, absence of arcs between nodes, and variable prior ordering restrictions. The study demonstrated whether the prior knowledge can codify expert knowledge for a particular dataset. de Campos and Castellano (2007) analyzed these restrictions not only empirically but theoretically as well. Moreover, they also demonstrated potential relationship between these three restrictions within the framework of score and search paradigms and conditional independence test heuristics.

1.3 Learned-From-Data Structure

Consider the situations where domain expertise is not reliable enough or expert based structure is limited by the large count of query variables given in the dataset. These situations lead to the structure learning using intelligent heuristics. A BN built out of data is prone to alleviate the domain expert. The structure suggested by an intelligent technique can be considered as a preliminary structure for which a domain expert may render his/her services for its improvement in the form of adding, removing or reversing the direction of the arc. This provides a pivot for judgments between the “pure” and implied structure. Learning structure from data by means of intelligent heuristics has the potential to achieve the probabilistic insight about variable mutual dependence within a domain of interest; moreover, it can be used to execute casual inference in decision making procedure. This dimension of learning structure can be split up into two types of techniques. Constraint-based techniques and search and score based technique. The former is described by (Pearl and Verma, 1991; Cheng, Bell and Liu, 1997; Spirtes, Glymour and Scheines, 2000; Yehezkel and Lerner, 2009) whereas search & score based technique is described in (Heckerman, Geiger and Chickering, 1995; Cooper and Herskovits, 1992; Heckerman, 1995; 2008; Keogh and Pazzani, 2002) techniques. The constraint based techniques employ statistical tests including mutual information test or chi-squared to determine conditional independence relationships among the query variables involved. This relationship is exercised to develop causality driven orientation rules which eventually evolved in construction the structure in the form of Bayesian Network (Pearl and Verma, 1991). Some notable techniques include

TPDA introduced by Cheng, Bell and Liu (1997), PC by Spirtes, Glymour and Scheines (2000), RAI by Yehezkel and Lerner (2009).

Score and search techniques are composed of two essential components. A searching algorithm and a scoring metric (also known as scoring function). The scoring function rests at the heart of any structure learning as it evaluates the best possible structure. In a simple brute force searching mechanism, every candidate of BN is passed to evaluate its scoring function after which the BN with highest scoring function is chosen.

Although brute force provides a Gold Standard BN, however it is restricted to dataset with only a very small number of features as otherwise generating a score for each possible candidate is NP hard with the increasing count of nodes in structures. A Solution to tackle this NP-hard issue is to restrict the number of potential candidates for a parent and employing a heuristic searching algorithm such as greedy algorithm (Heckerman, 1995; Heckerman, Geiger and Chickering, 1995). In the greedy search algorithm, the search is started from a specific structure which initially takes the input variables in a predefined order. The obtained structure is analyzed by the scoring function which results in adding, deleting or reversing the direction of the arc between two nodes. The ordering of the query nodes is characterized by the prior knowledge or by means of sophisticated techniques such as defined by Naeem and Asghar, (2013a), Hruschka and Ebecken (2007). The search continues to the adjacent structure reaching to the maximum value of a score if this value is greater as compared to the current structure. This procedure, which is known as hill-climbing search halts when culminating to a local maxima. One way of escaping local maxima is to employ greedy search. While employing greedy search, random perturbation of the structure is the way through which local maximum can avoid off. Apart from this approach, there are alternate approaches escaping of local maxima problem. These include simulated annealing introduced by Kirkpatrick, Gelatt and Vecchi (1983) and best-first search (Heckerman, 1995). In other words, if one describe the procedure of K2 in simple words then pursuit of an optimal structure is more or less tantamount to selecting the best set of parents for every variable but avoiding any circular dependency. This is the basic concept behind the K2 algorithm.

1.4 Formal Notation of a Bayesian Network

A Bayesian Network (BN) which is also known in an alternate name of Belief Network or Bayesian Belief Network (BBN) is a graphical model representing a process of an arbitrary nature. We can describe it by a triplet <D, G, R>. The first component of this triplet denotes the underlying dataset. The second component indicates a graph whereas the last component is set of parameters representing the underlying network. It is useful to explain each of them in detail in a formal notation. The second component G belongs to the family of Directed Acyclic Graph (DAG). Every query node in this DAG is a representation of query variables of the underlying objects or process. It is inscribed as a set of independence conditions; which means each query variable does not depend on its corresponding parent in the DAG. The component R holds parameters $\Theta[\overset{i}{Z}(pa(\overset{i}{Z})] = P(\overset{i}{Z} \mid pa(\overset{i}{Z})$. For each possible value of $\overset{ij}{z} \in \overset{i}{Z}$ and $pa\,(\overset{ij}{z}) \in pa\,(\overset{i}{Z})$ where i denotes ith variable Z and j denotes the jth state of ith variable Z. $pa\,(\overset{i}{Z})$ indicates the set of potential parents of the variables $\overset{i}{Z} \in G.$ Each query variable $\overset{i}{Z} \in G$ is denoted as a vertex or node in a DAG. Keeping in view of the best structure. As the number of graphs in structured learning are not limited to single graph during the searching process, so it is useful if we consider more than one graph in our consideration given that $\overset{G}{pa}(\overset{i}{Z})$ which shows the parents of the variable $\overset{i}{Z}$ in the DAG. The cumulative joint probability of a single DAG can be calculated by the equation.

$$P_b(\overset{1}{Z},....,\overset{N}{Z}) = \prod_{i=1}^{N} P_b(\overset{i}{Z} \mid pa(\overset{i}{Z})).....(Eq....)$$

The set of data which is to be learnt can be formally described as $O = \{o_1,......,o_n\}$ where

$o_i = \{\overset{1}{z}_i, \overset{2}{z}_i ......, \overset{N}{z}_i\}$. Note down that subscript points out the number of observation and the superscript is the indication of number of query variables or column in the data set. The value of N is the total count of instances in the dataset in which each instance covers all of the variables. We set forth a compulsion that there must exist at least 2 instance below which although the network may be built but the division of training and test data set requires this value to be $N \geq 2$. Each query node has varying number of distinct states such that we can express $\overset{i}{\underset{j}{Z}}$ indicates the counts of ith variables with jth states. Each structure $g \in G$ of the Bayesian Network can be denoted by N number sets of parents $\overset{1}{\prod},...\overset{N}{\prod}$. In simple words, we can state that for each node $j = 1,....N$ the set $\overset{j}{\prod}$ is a set of parent nodes in which a node has no self loop or close loop. Formally it can be represented such that $\overset{j}{\prod} \subseteq \{\overset{1}{Z},....\overset{N}{Z}\} \setminus \{\overset{j}{Z}\}$.

1.5 Graphical Representation of Bayesian Network Structure

The evolution of a Bayesian network can be traced back into a simple structure where a tree is also a kind of graph. In simplical way, we can divide the Bayesian Network in three categories.

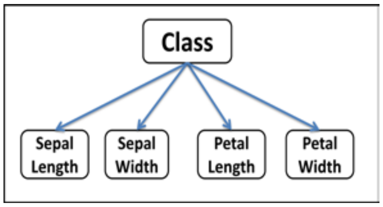


Figure 1.2: A simple BN Tree

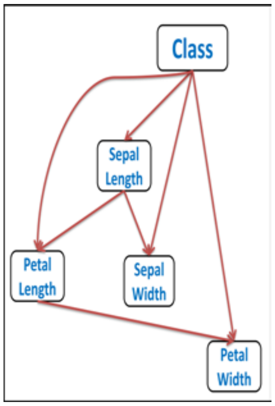


Figure 1.3: A Polytree

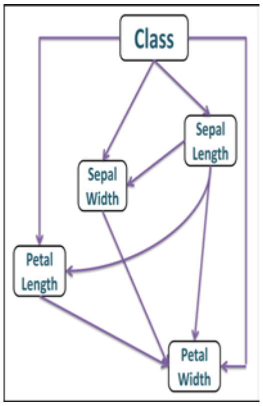


Figure 1.4: General Bayesian Network Graph

A Tree data structure where every non class query variable is represented by a node attached to a single parent node whereas this single parent node is either a class variable or non class variable. However, as we are interested in inference from class variable, the class variable has no parent node. A sample BN tree is shown in the figure 1.2. The second category is a polite (see figure 1.3) Bayesian Network structure in which any node can be linked to more than one parental node but with a restriction. The restriction implies that any two nodes in the learnt structure must not have at least and at most a single path connecting both of them. Three figures 1.2 to 1.4 have been drawn from the iris dataset using entropy as scoring function. The third category which is shown in figure 1.4 is a general belief network. The first two categories can be termed as simply connected networks whereas the general belief or Bayesian network has no such restriction as placed in the other two categories. By general Bayesian network, any node can be linked to more than one parent node; moreover such paths are allowed in redundancy where any of two nodes can be reached.

1.6 Algorithms of Structure Learning in BBN

In 1968, Chow and Liu presented a notion for building Bayesian Network (BN). This BN was a simple Tree BN. The measure used for linking the node was Mutual Information (MI). The outcome of this technique delivers the joint probability distribution which is assumed to fit the data in an optimized way. The algorithmic complexity for Tree BN is $O(N^2)$ given N is the number of variables; although the time complexity for general BN is far more than it.

Reuben and Pearl (1988) modify the basic Chow Liu algorithm. The modified algorithm was demonstrated to construct a more complicated network which is a polytree. Later on, Herskovits and Cooper (1990) proposed the Kutato algorithm which was in fact the earlier version of the notable K2 algorithm. This algorithm was meant for the generalized Bayesian network. As the complexity of this BN was more than exponential. Cooper and Herskovits (1992) reduced its space of possible DAG by means of introducing the concept of initial ordering. Another assumption during the structure learning is about the independence of the variables with respect to each other. At the initial step, the entropy of the BN is found. Keeping in view of the minimum entropy of the whole network, the arc between nodes are established. The final outcome is a network with minimum entropy.

1.7 Scoring Function

When we discuss about the scoring metric or scoring function then mutual dependence and correlation between two attributes of a dataset attains an essential notion in the sphere of structure learning. Numerous pair wise measures have been introduced explaining a particular or general relationship (Wasserman, 2007; Bagdonavicius, Kruopis and Nikulin, 2011; Gibbons and Chakraborti, 2003; Corder and Foreman, 2009). However, it was described that correlation and dependence, both are intrinsically different phenomenon. Albeit wide application of correlation in various domains of interest has been reported but a careful examination of the correlation measure delivers two problems in structured learning. The first issue is related to its incapability of describing the nonlinear structure between the random variables. It has been pointed out that two unrelated variables does not suggest their independence to each other (Grimmett and Stirzaker, 2001). The second problem is the inability of providing circumscribed knowledge about the underlying true dependence nature (Grimmett and Stirzaker, 2001). This arises a dictum "*correlation is unable to imply causation*" entailing that correlation is not ideally well suited in classification problems for the sake of establishing causal relationships between variables (Aldrich, 1995).

There are some certain characteristics associated with scoring function which are quite useful in speeding up the structural learning procedure (Liu and Han, 2010; Jensen and Nielsen, 2007). The first characteristic is the ability of any scoring metric to balance the accuracy of a structure keeping in view of the structure complexity. The second characteristic is computational tractability of any scoring function (metric). The most notable and worthy property is the ability of a scoring function to be decomposed into factors or terms in such a way that the node can be involved in its parents for enumeration. It results into operation of insertion, deletion or edge reversal, the total score of the DAG can be computed easily by means of updating the local score of one or two node's scores towards their parent. This fundamental characteristic leads to the introduction of other essential characteristics of scoring function which is caching techniques. It is observed that most scoring function changes remain non variant in their nature after completion of every graphical operation. Friedman, Geiger and Goldszmidt (1997) points out that this caching is useful in reducing the time complexity attached to structure learning, however it is also indicated that it does not necessarily lead to improvement in the precision of the learnt results. This exciting theoretical advancement has its roots in the verity that the scoring function is decomposable. In brief, learning methods can utilize this specific structural property of scoring functions in order to design effective dynamic programming heuristics to avoid repeated computations. The same was presented by Silander and Myllymäki, (2006) that there are relatively simple technique available for construction of the exact maximal scoring network.

Bayes (Cooper and Herskovits, 1992), BDeu (Buntine, 1991), AIC (Akaike, 1974), Entropy and MDL (Lam and Bacchus, 1994; Suzuki, 1999) and fCLL (Carvalho et al, 2011) have been reported to satisfy these characteristics. Among these scoring functions, AIC, BDeu and MDL are based on Log Likelihood (LL) as given below:

$$LL(G\mid D)=\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right)..................(1.1)$$

Where G denotes directed acyclic graph given dataset D. Other three counter include n, qi and ri indicates number of cases, number of distinct states of a feature variable and number of distinct states of parent of a ith feature variable. The log likelihood tends to promote its value as the number of features increases. The phenomenon occurs because additions of every edge is prone to pay contribution in the resultant log likelihood of final structure. This process can be controlled somewhat by means of introduction of some penalty factor or otherwise restricting the number of parents for every node in the graph.

1.7.1 Bayes

We earlier mentioned that Cooper and Herskovits (1992) introduced an algorithm K2 in which greedy search was employed while a scoring metric of Bayes was used. It was described that the structure with highest value of Bayes metric was considered the best representative of the underlying dataset. It motivates us to describe Bayes metric formally expressing in mathematical notations.

Let there is a sequence of n instances such that $\overset{n}{z}=d_1d_2d_3.....d_n$ the Bayes scoring function of structure $g\in G$ can be formulated in form of the equation.

$$P_b(g,\overset{n}{z})=P_b(g).\coprod_{j\in J}\left(\prod_{s\in S(j,g)}\frac{(\alpha-1)!\cdot\prod^{j}_{q\in\overset{j}{A}}n[a,s,j,g]!)}{(\overset{j}{n}[s.j.g]+\alpha-1)!}\right).................(1.2)$$

Where P$_b$ (g) is the prior probability of full network $g\in G$. The prior probability can be omitted in the computation. The notation $j\in J=\{1,....,N\}$ is the count of the variable of the network g, and $s\in S(j,g)$ is the counting of the set from all sets of values obtained from the parents of the jth node variable. The expansion of the denominator factor can be expressed mathematically as below.

$$n(s, j, g) = \sum_{i=1}^{n} I(\underset{i}{\overset{j}{\pi}} = s)..................(1.3)$$

$$n[q, s, j, g] = \sum_{i=1}^{n} I(\underset{i}{z} = q, \underset{i}{\overset{j}{\pi}} = s)..................(1.4)$$

Where $\overset{j}{\pi} = \overset{j}{\Pi}$, the function I (E) = 1 given E is true, and I(E)=0 if E is false. The K2 learning algorithm uses Bayes as its core function in its each iteration enumerating all potential candidate graphical structure. The outcome of this enumeration is an optimal learnt structure which is stored in g^*. This optimal structure posses highest value of

$P_b(g, \overset{n}{z})$. Such that $\forall g \in G - g_0, if\ P_b(g, \overset{n}{z}) > P_b(g^*, \overset{n}{z}),\ then\, g^* \leftarrow g$

The above equations were required to be decomposed simply into a computational model, otherwise this theoretical model requires a very large number of computations involving factorial. It means score value for a network g can be enumerated as the sum of scores for the individual query variables and the score for a variable is calculated based on that variable alone and its parents.

The approach in the scoring function inspired learning performs a search through the space of potential structures. These include Bayes, BD, AIC, MDL and Entropy measure, all of which measures the fitness of each structure. The structure with the highest fitness score is finally chosen at the end of the search. It has been pointed out in (Pelikan and Goldberg, 2006) that Bayes often results in overly simplistic models requiring large populations in order to learn a model which holds the capability to captures all necessary dependencies. On the other hand, BDeu tends to generate an overly complex network due to the existence of noises. Consequently, an additional parameter is added to specify the maximum order of interactions between nodes and to quit structure learning prematurely (Pelikan, 2005). As noted in (Correa and Shapiro, 2006), the choice of the upper bound given the network complexity strongly affects the performance of BOA. However, the proper bound value is not always available for black box optimization.

Jensen and Nielsen (2007) discussed two important characteristics for scoring function used in the belief network. The first characteristic is the ability of any score to put the accuracy of a structure in equilibrium in context of complexity of structure. The second characteristic is its computational tractability. Bayes has been reported to satisfy both of the above mentioned characteristics. Bayes denotes the measurement of how well the data can be fitted in the optimized model. The decomposition of Bayes can be proceeded as below:

$$BIC(S \mid D) = \log_2 P(D \mid \hat{\theta}_S, S) - \frac{size(S)}{2}\log_2(N)..................(1.5)$$

Where $\hat{\theta}$ is an estimation of the maximum likelihood parameters given the underlying structure S. Jensen and Nielsen (2007) discussed that in case of completion of the database, Bayesian Information Criterion (Schwarz, 1978) is reducible into problem of determination of frequency counting as given below:

$$BIC(S \mid D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log_2(\frac{N_{ijk}}{N_{ij}}) - \frac{\log_2 N}{2}\sum_{i=1}^{n} q_i(r_i - 1)..................(1.6)$$

where Nijk indicates the counts of dataset cases with node Xi in its kth configuration and parents Π(Xi) in jth configuration, qi denotes the number of configurations over the parents for node Xi in space S and ri indicates the states of node Xi.

1.7.2 BDeu

Another scoring measure which depends only on equivalent sample size N′ is Bayesian Dirichlet for likelihood-equivalence for uniform joint distribution (BDeu) introduced by Buntine (1991). Carvalho et al. (2011) has provided and discussed its decomposition as below in mathematical form:

$$BDeu(B,T) = \log(P(B)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma\left(\frac{N'}{q_i}\right)}{\Gamma\left(N_{ij} + \frac{N'}{q_i}\right)}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma\left(N_{ijk} + \frac{N'}{r_i q_i}\right)}{\Gamma\left(\frac{N'}{r_i q_i}\right)}\right)\right)..........(1.7)$$

1.7.3 AIC

Akaike Information Criterion (AIC) originally defined by (Akaike, 1974) is defined mathematically:

$$AIC = -2 \times \ln(likelihood) + 2 \times K.........(1.8)$$

Where K denotes the number of parameters in the given model. However, Bozdogan (1987) decompose AIC into a scoring metric which can be used in BBN. AIC is established on the asymptotic behavior of learnt models and quite suitable for large datasets. Its mathematical equation has been transformed into

$AIC(B\,|\,T) = LL(B\,|\,T) - |\,B\,|$.........(1.9)

1.7.4 MDL

Minimum Description Length (MDL) introduced by Lam and Bacchus (1994) initially and then refined by Friedman and Goldszmidt (1996), Suzuki (1999) and Terent'ev and Bidyuk (2006). It is mostly suitable to complex Bayesian network.. We shall formally define it as below. Let a given sequence $\overset{n}{z} = d_1 d_2 d_3 .....d_n$ of n number of instances, the MDL of a network $g \in G$ can be enumerated as below.

$L(g,\overset{n}{z}) = H(g,\overset{n}{z}) + \frac{k(g)}{2} \cdot \log(n)$ Where the function k(g) represents the independent conditional probabilities in the network. $H(g,\overset{n}{z})$ is entropy of structure with respect to the variable $\overset{n}{z}$ which can be expanded into the following notation. $H(g,\overset{n}{z}) = \sum_{j \in J} H(j,g,\overset{n}{z})$ and $K(g) = \sum_{j \in J} K(j,g)$

Given the jth node variable, the value of MDL can be enumerated as below:

$L(j,g,\overset{n}{z}) = H(j,g,\overset{n}{z}) + \frac{k(j,g)}{2} \cdot \log(n)$ where $k(j,g)$ is the count of independent conditional probabilities of jth variable. This value can be expressed in more detail as below.

$K(j,g) = (\overset{j}{a} - 1) \cdot \prod_{k \in \varphi^j} \overset{k}{\alpha}$ while $\varphi(j) \subseteq \{1,....j-1, j+1,..., N\}$ is a set given $\overset{j}{\Pi} = \{\overset{k}{Z} : k \in \varphi^j\}$

Given the jth node variable, the entropy can be expanded into the following expression.

$$H(j,g,\overset{n}{z}) = \sum_{s \in S(j,g)} \sum_{q \in A^j} \left(-n[q,s,j,g] \cdot \log \frac{n[q,s,j,g]}{n[s,j,g]} \right) \text{.........(1.10)}$$

$$n(s,j,g) = \sum_{i=1}^{n} \left[I(\underset{i}{\overset{j}{\pi}} = s) \right] \text{.........(1.11)}$$

$$n(q,s,j,g) = \sum_{i=1}^{n} \left[I(\underset{i}{z} = q, \underset{i}{\overset{j}{\pi}} = s) \right] \text{.........(1.12)}$$

where $\overset{j}{\pi} = \overset{j}{\Pi}$ indicates that $\overset{j}{Z} = \overset{j}{z} \forall k \in \varphi^{j}$; the function I(E) yields a positive identity number when the predicate E is true and the function I(E) becomes false when I(E)=0.

MDL differs from AIC by the log N term which is a penalty term. As the penalty term is smaller than that of the MDL, so MDL favors relatively simple network as compared to AIC. The mathematical formulation is composed of explanation of Log Likelihood (LL) as given below:

$$LL(B\,|\,T) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) ....(1.13)$$

The value of LL is used in obtaining the decomposition of MDL as below:

$$MDL(B\,|\,T) = LL(B\,|\,T) - (1/2)\log(N)\,|\,B\,| ....(1.14)$$

|B| denotes the length of network which is achieved in terms of frequency calculation of a given feature’s possible states and its parent’s state combination with feature as following:

$$|\,B\,| = \sum_{i=1}^{n}(r_i - 1)q_i ....(1.15)$$

1.7.5 fCLL

Carvalho et al. (2007; 2009; 2011) introduced factorized conditional log likelihood (fCLL) and empirically proved it to be reasonable among other established scores. These scores formulate propositions for well motivated model selection criteria in structure learning techniques. The noteworthy issue by employing these well established scores, however, is that they are prone to intractable optimization problems. Chickering, Heckerman and Meek (2004) argued that it is NP-hard to compute the optimal network for the Bayesian scores for all consistent scoring criteria. AIC and BIC are usually applied under the hypothesis that regression orders *k and l* are identical. This assumption brings extra computation and also come up with an erroneous estimation with theoretical information measure in structured learning. Yang and Lee., (2012) shows the linear impact of improvement in model quality within the scope of exercising Bayes function score in K2 (Cooper and Herskovits, 1992). However, it was arguable that there must be an intelligent heuristics to sharply extrapolate the optimized size of the training data. We are of the view that exploiting various intelligent algorithms for tree and graph, an optimized solution can be achieved.

1.8 Searching Algorithm

There are different search algorithms available in Bayesian Belief Network. Among all of these the search algorithm K2 has been shown a better search algorithm (Jung and Choi, 2013; Hesar, Tabatabaee and Jalali, 2012). Numerous comparison among the various flavors of classifiers based on Bayes theorem have been reported. Recently Hesar, Tabatabaee and Jalali (2012) presented a comparison among machine learning techniques including Naïve Bayes Classifier, K2, Hill Climbing, Iterative Hill climbing and Greedy Tick Thinning. Their finding was formulated on three observations. Appropriate representation of underlying dataset, efficiency or construction time and accuracy. They pointed out that K2 searching algorithm exhibited better performances in drawing an appropriate structure of underlying dataset as well as in case of construction time. Although hill climbing gives somewhat better accuracy as compared K2. Moreover, simple Naïve Bayes classifier was shown as poor among all of the system in their experiment. Moreover, they also pointed out that in the domain of meteorology, K2 searching algorithm was recommended.

1.9 Inference

Inference on a structure obtained in the form of a Bayesian network is the process in which conditional probability tables are generated from the variables in the dataset. Inference on a learnt structure can be exact or approximate given the prior knowledge of the network where the prior knowledge is indicated by a set of evidence variables. Pennock (1998) shows that exact inference is NP hard. Lauritzen and Spiegelhalter (1988) introduced an exact inference algorithm which is one of the most notable exact inference algorithm. This algorithm is for multiple connected networks which initially transforms the learnt structure into a junction tree. The junction tree is finally subjected to exact inference. The concept of junction tree is to disintegrate a global calculation on a joint probability distribution table into a linked set of local computations. The major steps in the junction tree algorithm include Moralization, Triangulation, Construction of the junction tree, Potentials transfer and then Propagation. The complexity of this exact inference technique is a function of three parameters. These parameters include the distinct states of the query variables , overall density of the Bayesian network and the width of the cliques of the nodes in the network. Every potential table of a junction tree denotes a joint probability distribution. A potential table delegates a probability to every possible combination of states of query variables of the network., The number of probabilities in a potential table is the product of two factors which include the number of distinct states of query variables and the width of the node clique. The size of the potential table directly influences the time complexity as it is clear that even for moderate size of both of these factors will give a very large value for the size of the potential.

1.10 Initial Ordered Set of Node Variables

It was earlier mentioned that the computation of all possible structures from a dataset with even a relatively moderate size of feature set leads to more than exponential in time complexity. One way to reduce this complexity was to arrange the initial features in a specific fashion so that the final space of the possible DAG can be reduced significantly. The K2 algorithm uses this strategy. Let us assume that there exists an ordered set of nodes $\{\overset{1}{Z},\overset{2}{Z},.....,\overset{n}{Z}\}$ where this order it is a problem of previous knowledge or requires an intelligent heuristic. In this set, the first feature or node variable $\overset{1}{Z}$ is simply a class node which contains no parent at all. We can also term it as principal root node in the structure. The second node $\overset{2}{Z}$ is only connected to class node which is a compulsion for every node in the Bayesian network structure. $\overset{3}{Z}$ is a compulsory child node to class node but a potential child for the node $\overset{2}{Z}$ which just precede it. We can continue this fashion for every node such that the last node can be the child node of each non class node but with a compulsory child node to class node. The strategy is simple that as we move away from the class node, each node can be linked to more and more node in the capacity of child nodes. In K2 algorithm, a further limitation was imposed in the form of maximum number of points for which only a suitable value can control the feature space in the reasonable computational cost. Here the question arises by which mechanism it is determined whether any node can be a child node to any of its predecessor nodes in the ordered list. The answer is provided by scoring function. The scoring function is calculated $P_b(g,\overset{n}{z})$ at the end of each iteration in the searching algorithm. The node $\overset{1}{Z}$ to $\overset{N-1}{Z}$ may be linked to the Node $\overset{N}{Z}$ but with the condition that at the end of each iteration the posterior probability of the structure in question be higher than its previous value. In this way, the set of ordered list saves us significantly from large number of calculations but meanwhile it raises another question how to define this ordering scheme. Some measures have been proposed by Naeem and Asghar (2013a), Hruschka and Ebecken (2007) in this regards.

1.11 Mutual Information between Variables

The initial ordering scheme for greedy search K2 algorithm is reported to be characterized by Mutual Information (MI). The concept of MI was more than 4 decades ago, when Chow and Liu (1968) introduced the concept of mutual information between two nodes. The concept is a metric to judge the degree of dependence between two arbitrary query variables say $\overset{i}{z}$ and $\overset{j}{z}$ with the following formal expression.

$$MI(\overset{i}{z},\overset{j}{z}) = \sum_{\overset{i}{z},\overset{j}{z}} P(\overset{i}{z},\overset{j}{z})\cdot Log\left(\frac{P(\overset{i}{z},\overset{j}{z})}{P(\overset{i}{z})\cdot P(\overset{j}{z})}\right)\text{.........(1.16)}$$

While examining its nature of dependency between two variables, it is analogous to the well renowned statistical correlation measures. Moreover, when we examine it in perspective of a classification system, then it behaves like a window through which we can approximate the amount of information about second variable $\overset{j}{z}$ defined by the first variable $\overset{i}{z}$. During the calculation, it always gives a value with lower bound of zero and upper bound of one. The lower bound value is assumed when two variables are entirely independent to each other and vice versa. It is also stated that if two variables are independent of each other then $P(\overset{i}{z},\overset{j}{z}) = P(\overset{i}{z})\cdot P(\overset{j}{z})$. This equalize the numerator and denominator factor in logarithmic function in the equation number 1.16; thus simply log value gives a value of 0. The important characteristics MI was exploited in the development of Initial ordering for K2 algorithm by Terent'ev and Bidyuk (2006). Bi and Chen, (2011) show that there are situations when experts are not able to distinguish the relationship between some of query variables. They discussed the relationship of certain crop related factors which may lead to corn border attack. Although they did not cross out the importance of expert elicitation over construction of a Bayesian Network, however they have shown that there are certain situations when experts can come up with wrong construction of Belief network. In such situations, BBN proves its effectiveness in establishing crop disease forecast system.

Scalability of structure learning has always been an issue. A notable effort was made by Freund and Mason (1999) in which incomprehensibility of ensembles through combination by means of Boosting was incorporated into a new technique know as Alternating Decision Trees (ADTree). Taking advantage of this feature of ADTree, some individual algorithmic steps were taken with the objective of reducing the time complexity of structure learning. During the enumeration of scoring measures, it is an essential step of counting the number of instances for various structural configurations of every node's parent. This is quite evident that this computation is burdensome. To address this problem in medium or large network, the data structure of Adtree has been used to reduce the time complexity (Lee et al., 1998). However, Esmeir and Markovitch (2006) shows that there are scenarios where ADTree was unable to learn the concept correctly while generating classifiers with low accuracy even after thousand rounds of Boosting.

1.12 Feature Ordering vs. Feature Subset Selection

Feature sub selection or Feature ordering is another important dimension for those classifiers which are sensitive to the variable ordering scheme. The performance of the BBN classification system has been reported sensitive to the underlying characteristics of the data. It is reported that the performance of a classifier system is a function of discriminative variables. Numerous feature subset selection systems have been reported in last two decades; however no universal technique has been introduced to cater each and every kind of data which is applicable to every classification system. It is a preliminary requirement for any classification system to get its input 'prepared'; here the 'prepared' denotes that the input must be presented in the form of binary, nominal, continuous or categorical feature values. Although, feature selection is found useful for every classifier and this leads to the emergence of numerous taxonomies in literature; but there are situations when the user of classifier intends not to surrender any features during pursuit of improving the accuracy of the classifiers. Moreover, it is reported that some classifiers are quite sensitive to the order of the variables supplied as basic input to the classifier. In such scenarios, feature ordering become more important. These include Random Forest, Naïve Bayes Belief network, PART, TAN etc. We in this study have analyzed that two classification systems have been found significantly sensitive to the order of variables / features involved. These are Naïve Bayes Belief Network and Random Forest. Feature Ranking algorithms like feature subset selection algorithm establishes the relevance of an attribute compared to the class, however, there is no question of dropping any feature (Naeem and Asghar (2013a). When we talk in the perspective of the future ordering rather feature selection under the gist of improvement in the classification then we can divide feature selection techniques into two categories. Ranking oriented feature selection techniques and Subset oriented feature selection techniques. Some notable techniques fall in first category including Gain Ratio, Info Gain, Relief Attribute, Symmetrical Uncertainty Attribute, Chi Squared Attribute. While some well known techniques in the second category include Correlation based Feature Subset selection, Consistency Subset and Filtered Subset. We have formulated comparison of our proposed technique to all of these techniques. Although we are not the first to define a concept of feature ranking in the context of classification system yet there are numerous related contributions delivered by us in this study.

1.13 Problem Statement

The literature surrounds numerous feature selection techniques. Most of them have been constructed on an *ad hoc* basis, attempting to reduce large number of features with the underlying characteristics of relevancy and redundancy. However, we argued that there are situations when surrendering features is not viable (chapter 4). This requires to build optimal feature ordering raising the question.

- Can a feature ordering technique be derived with the capability to minimizing the classification error ?
- How can we construct a feature ordering technique which is not only restricted to structure learning and equally useful for other feature order sensitive classifier. ?

The second problem introduced in this thesis is related to scoring metrics. Currently many of the scoring metric are parametric in nature wherein the efficiency of the scoring metric is greatly influenced by external parameters. Unluckily, knowing the best value of parameter in advance is not possible. This formulates a problem

- Can we develop a scoring metric which is non parametric, free of implicit assumptions but delivering better classification accuracy in structure learning. ?

Certainly, the crux of such scoring metric and feature ordering technique relies on bounded metric. This again raises a question.

- Can we analyze this metric in the framework of well renowned entropy and mutual information.?

1.14 Research Contribution

After investigating the literature review, we chose BBN for our domain of interest. The reason lies in BBN's capability of excellent performance in pdf estimation in pragmatic situations. Furthermore, BBN has shown its sound mathematical rigor in probability theory which is not true in case of the multitude of other *ad hoc* approaches. We can briefly summarize the originality of the this study in following two dimensions. The first dimension is contributing towards scoring function with following characteristics.

- Analysis of various structure learning techniques.
- Derivation of the optimal information theoretic scoring function namely Non parametric Factorized Likelihood Metric (NP-FiLM) maximizing a discriminative model likelihood (Chapter 2)
- Empirical study of NP-FiLM, illustrating how they behave in comparison to six peer scoring metrics, numerous tree and function classifier in the paradigm of stability and accuracy across a range of variety of datasets (50 UCI and one HCV nucleotide sequence).

The second contribution is in the field of feature variable sensitive classifiers with the following observations.

- Introduced a novel measure of coherence between two features (herein called as Polarization Measure (PM))
- The introduced technique is not only efficient but also delivers better result as compared to its peer techniques.
- The introduced technique is quite scale able as well as stable to large and small data set.
- We have shown empirical results for the comparison between well known rankers and feature subset selection techniques along with some recommendations.

Another tangential contribution of this thesis is towards the analysis of tree classifiers. We have shown that the PM metric can be used as a meta characteristic to predict the classification accuracy of the Decision Stump tree. This shows its optimal and sound characteristics being an information theoretic measure explaining the mutual dependence relationship.

1.15 Organization of Thesis

This study comprised of six chapters. This chapter was the introduction and background of the prevalent techniques and theoretical background of structure learning. Chapter 2 describes all of those classification system which is related to our proposed system. We have analyzed technical detail of these existing classifiers mostly of which belong to the tree and function classifiers. Chapters 3 provide in detail discussion of the scoring function. We have introduced a new scoring metric and also contributed with a solid theoretical work in this chapter. Experimental work is also described in this chapter for validation of theoretical work. In chapter 4, we have discussed subset oriented feature selection and variable ordering feature selection. A simple yet robust measure Polarization Measure (PM) is introduced. The introduced measure is fast with an asymptotic complexity of O(n) and saleable to large dataset. In chapter 5, practical application of the proposed measure in the shape of a model has been realized. Two practical applications have been discussed in this chapter. Chapter 6 is dealt with a conclusion and some ideas of possible extension of this study in the field of structure learning.

Chapter 2

ANALYSIS OF EXISTING CLASSIFIERS

In the previous chapter, preliminaries of structure learning and feature selection were discussed. In this section, we shall discuss each of the classifier to whom we have made comparison in empirical evaluation of our proposed metric. In the last section, a comparison among these classifiers and analysis is presented.

2.1 Tree Classifiers

Tree Classifier which is also known by some other terms per se Decision Tree, Regression Tree and Prediction Tree are important and robust predictor tools. They have established their place in the community of data mining, machine learning and statistics. They pose an explicit illustration of the dataset in the form of a structure as a predictive model which maps observation around an item and its conclusive decision about item’s target value. In general the decision tree exercises a greedy approach or an intelligent heuristics to direct their search in the course of vast hypothesis space. Due to this nature of decision tree, determining an optimal decision tree has been proved an NP-complete problem. The tree classifier can be assumed as the sister classifier for the graphical models based on the Naïve Bayes theorem. Straightforwardly the tree is scaled down version of a graph. Hence both tree and graph based algorithms have great many features to be competed with each other. That is why it is instructive to give a brief outlines of these approaches.

2.1.1 Random Forest Classifier

Random Forest (RF) classifier (technically a homogeneous ensemble) was initially introduced by Breiman (2001). It takes un-pruned decision trees such that each node of the tree with best features such that a randomly selected subset out of all features is chosen. The data sampling used in this process is bootstrap in which sampling is performed with replacement from the original dataset. The un-pruned trees are built for reducing bias while the randomization is meant for maintaining high diversity between trees in the forest. We shall recall that here Bias denotes the systematic error term which is independent of the underlying learning sample; whereas the variance is defined as the error caused by the variability of the model in the learning instances with randomness. The crux of this classifier revolves around it voting mechanism. Decisions are determined through simple voting. This approach has been recognized a well known, well established successful ensemble methods. The generalization error of a forest is the accumulated value of the strength scores of the individual trees in the forest and upon the dependencies among them. The technique introduced by RF delivers high classification accuracy as compared to numerous well established classification approaches such as AdaBoost (Freund and Schapire, 1997) and SVM (Vapnik, 1999). The reason behind its better performance roots in its ability of being robust to noise, void of over fitting problem and its improved time complexity (Dietterich, 2000). One notable characteristic is its high efficiency over the significantly large dataset. The application of RF classifier has been reported in diversified domains of interest. Some include identifying curvilinear structure of mammograms in the domain of biomedical image processing (Berks et al., 2011), Genomic selection (Ogutu, Piepho and Schulz-Streeck, 2011), machine fault diagnosis (Yang, Di and Han, 2008), Natural Language Processing (Kobyliński and Przepiórkowski, 2008) and many more.

Breiman (2001) indicated that Random Forest is akin to generate error rates almost at the same level as in case of Bayes rate over a wide cross-selection of learning tasks. However Robnik-Šikonja (2004) pointed out that improvement in accuracy in some domain is possible either through application of a combination of various feature selection criteria to decrease correlation in the forests or in other way substitution of majority voting by means of locally weighted voting. This obviously provides a scope that there is a significant margin to apply feature selection technique to improve the accuracy of Random Forest classifier. Ozcift (2012) introduced a wrapper feature subset evaluator which uses Random Forest as its kernel. They exercised the evaluator over four dataset and presented improved results in comparison to fifteen classifiers. However, their technique surrender very large fraction of actual dataset. Such technique may become argue able in situations where the users have the intention to utilize all or large proportion of the original features. Menze et al. (2011) introduced a version of Random Forest namely oblique Random Forest (oRF). It was shown that oRF is built out of multivariate trees which precisely learn optimum splitting directions at internal query nodes employing linear discriminative models instead of applying the random coefficients in RF. Furthermore it was also observed that it is optimized in classification ROC Area for those dataset which have tighten correlation among features; however its overall usefulness is limited to only binary classes. Breiman (2001) pointed out that the un-pruned trees in random forest are drawn for reducing bias. Earlier to it, Buntine (1992) presented Bayesian based classification algorithms for the purpose of tree averaging to shorten the variance in learning procedures. Later on, numerous techniques used Naïve Bayes theorem in their discriminant functions.

2.1.2 Decision Stump

Decision Stump was originally introduced by Iba and Langley (1992). This falls under the breed of classifiers in which one level tree is used to classify instances by sorting them, while the sorting procedure is based on futuristic value. Every node in a decision stump dictates a query variable from an instance which is to be classified. Every branch of the tree holds the value of the corresponding node. Although decision stump is widely used classifier yet it is assumed as a weak classifier. In this threshold oriented classification system, sample instances are classified beginning from the root node variable. The sorting is carried out on their feature values which a node can take on. If the selected feature is specifically informative, this classifier may yield better results, otherwise it may lead generating the most commonsensible baseline in the worst situation. The weak nature of the classifier lies in its inability to tackle the true discriminative information of the node. Although to cope up this limitation, the single node, multi-channel split decision criteria is introduced to accentuate the discriminative capability; nonetheless its results are still not as appealing as compared to its peer classifiers. On the other hand, the decision stump weka implementation posses three advantages. Firstly it has the ability to handle training instances given any distribution. Secondly it requires less computational cost. Thirdly, it is a favorite baseline algorithm for the robust boosting classification system. The later intuition has been introduced and formalized by Tumer and Ghosh (1996). Ensemble learning classifiers are measured in two parameters. These include the bias and the variance of the learned model (Wolpert, 1997). The perception of the bias concept of a model is the measurement of the difference between the true function responsible for generating the data and the ‘‘average’’ function returned by the learning algorithm, where “average” is dictated by the overall possible training instances. Whereas the variance is concerned, it is the statistical variance over the possible training instances of the function returned by the learning algorithm (Oza and Tumer, 2008). Boosting is aimed towards reducing bias in comparison to variance. This is the underlying reason for which a boosting classifier may lead to improve the results by adjusting the bias/variance ratio when the base classifier has high bias and low variance. Apart from decision stump, Naı¨ve Bayes classifiers is a also a good candidate for being such a base classifier.

2.1.3 Random Tree

The Random Tree classifier introduced by Breiman in 2001. This classifier draws a tree in which K randomly chosen attributes are selected at each node. There is no pruning operation involved. Furthermore, it has the capability to accept the estimation of class probability distribution on a backfitting (hold out set) method. Random Tree classifier exhibits significant training efficiency, but consuming limited memory. Second advantageous characteristics lies in its ability to use only single pass over the data to build up the tree. Fan et. al (2004, 2005) pointed out that these classifier are not only efficient but also simple with appreciable reduced classification error. It is a good candidate for its role as base classifier in ensembler techniques. The notion for this classifier’s superiority over its peer classifier in ensemblers relies in its optimally approximates for each instance yielding a true probability of being a member of a given class. Moreover, the random tree ensembles can form effective implementations of Bayes Optimal Classifiers.

2.1.4 REPTree Classifier

Reduced Error Pruning Tree (REPTree) classifier is a variant of decision tree learner classifier, however it is relatively faster than conventional decision tree algorithm with respect to noisy training data. In this classifier, a decision tree which can be a regression tree uses information gain as its split criteria and pruning is performed with backfitting. Moreover, the pruned tree decreases the complexity in the classification process. Usually pruning is employed to determine the best sub-tree of the initially grown tree with minimum error for the test samples. The query variables are sorted only once before the induction of the tree. Some other characteristics of this classifier include the count of tree depth, although it can be set as free of any restriction. The minimum total weight of the instances in a leaf and the minimum proportion of the variance on all the instances which require to be present at a node in order to perform splitting the node with in the regression tree. Missing values are treated by means of dividing the relevant instances into pieces same as that of C4.5. The REPTree classifier is widely used but mostly as a base classifier in the ensembler such as Bagging, first of all this notion was pointed out by Breiman, 1996. The core reason behind is that this tree classifier generates a suboptimal tree under the constraint that a sub-tree can only be pruned if it dose not have a sub-tree with a lower classification error than itself. The high potential candidacy of this classifier lies in the underlying fact that this classifier is prone to exhibit high variance. It is a know fact that ensembler are a good choice to produce a reasonable balance between bias and variance. The Bagging classifier employs the REPTree classifier as the base classifier with the motivation to lessen the variance (Breiman, 1996).

2.1.5 Decision Tree

One notable cluster of classifiers for which the proposed scoring function oriented Bayesian Network is very much related is decision tree algorithms. The decision tree algorithms were originally introduced in 1970, Where the study of Breiman et al., (1984) and his contemporary researchers including Quinlan (1986); Friedman, Geiger and Goldszmidt (1997); Brodley and Utgoff, (1995); You and Fu, (1976); Qing-Yun and Fu (1983); Fayyad and Irani (1992) and to restrict to a few; albeit there are many others who come up with a solid theoretical grounds in this computational area. One point worthy aspects of this breed of algorithms is that many decision tree algorithms introduced by researchers time after time; they all share numerous characteristics in common. These are based on a "divide-and-conquer" approach. According to this approach a classifier is built up in a shape of a decision tree, with non-leaf nodes testing the value of a particular class variable or pattern attribute or even some function of one or more data attributes. Leaf nodes of the tree are marked by identifiers of classes within the scope of problem domain. For every test instance, the outcome of the nonleaf node leads to a branch to build either a sub tree or a leaf node. To classify a new instance, it begins at the root node and directs to all of the branching and their corresponding non leaf node culminating at the specific test node of the tree. This way, the algorithm is able to ascertain the best tests at these nodes. Generally it is stated that computational procedures necessitated by building a decision tree can be rather complicated due to the searching heuristics and measure responsible for splitting the tree. However, on the other hand, the decision tree algorithms have been marked as very successful in various experimentation reported in numerous domain of interest.

2.1.6 CART

Classification And Regression Trees (CART) which was originally proposed by Breiman in 1984, established its popularity since its inception. CART is an appealing choice in case of noisy data in pursuit of developing a sophisticated tool for concept induction. This classifier has proved its influential role in practice of machine learning in last two decades. This classifier is a general framework towards the design of decision trees. Let’s assume there is a set of training data which means it contains labeled class. Every instance of the dataset is composed of query variables where each query variable is either categorical or numerical in nature. The working of building up the decision tree in CART can be described as below:

I. A single or a group of query variable is chosen at every non-leaf node of the tree.

II. Test function is enumerated for the selected attributes.

III. The size of the tree is restricted by means of suitable pruning procedure.

IV. Missing data instance found in various query variables is handled in CART. Either these values are estimated or ignored at all in certain cases.

V. Leaf nodes are identified as well as marked.

VI. Every leaf node is assigned with a class label.

2.1.7 C4.5 / J48

The C4.5 algorithm also known in the name of J48 algorithm in implementation of weka data mining tool (Hall et al., 2009). The C4.5 classifier was proposed by Quinlan in 1993. It also retains its popularity amidst its peer classifier and is known as a well-liked decision tree algorithm. The original C4.5 can be used for regression and classification problems. However, in weka implementation of J48, it can handle the classification problem only. In the process of building a decision tree, this classifier employs two-way splits for numeric variables and multi-way splits for categorical variables at nodes of the tree. At each node, the classifier examines a family of possible tests including various well known measures like Entropy, Gini, Information Gain or Misclassification Error. The mathematical formulation of these essential measure is as below.

$$Information\,Gain\,Ratio(X,Y) = \frac{Gain(X,Y)}{Split\,Info(X,Y)}\ldots\ldots\ldots\ldots\ldots\ldots..(2.1)$$

Where

$$Gain(X,Y) = Entropy(Y) - \sum_{a \in A} \left[\frac{Y_a}{|Y|} \times Entropy(Y_a) \right] \text{...........................}(2.2)$$

$$Split\,Info(X,Y) = -\sum_{a \in A} \left[\frac{|Y_a|}{|Y|} \times \log\left({|Y_a|}\big/{|Y|} \right) \right] \text{...........................}(2.3)$$

Where Y_a is the subset of Y for which the feature X has the state a. These equations are meant only for calculation of information gain for variables of discrete nature. The continuous valued feature variables are usually subjected to process of discretization. This measure is responsible for maximizing the value of splitting criterion dynamically defined during searching mechanism. In specific, training instances are sorted ab initio on the values of the numeric attribute when selected with the subsequent examination of each pair of adjacent values for enumeration of the best threshold value. The splitting criterion, which is required to be maximized in the formation of tree, is an information theoretic measure. This measure is responsible to take into account various numeric and various probabilistic distribution of test outcomes (Quinlan, 1996). However, there is an issue of a dense tree. The increased density of the tree leads to the over fitting of the training data. The adjustment of density of tree is carried out by a pruning procedure. The pruning procedure is characterized by statistical significance of splits in the nodes within the tree.

2.1.8 LADTree

LADTree is Logit boost multi class Alternating Decision Tree, presented by Holmes in 2002. Whereas the Logit boost strategy is boosting algorithm formulated by Friedman, Hastie and Tibshirani (2000). Prior to the introduction of LADTree, Freund and Mason (1999) presented Alternating Decision Tree (ADTree) algorithm. ADTree is a classifier which carries its notion in decision tree and option tree (Kohavi and Kunz, 1997). ADTree was introduced with provision of symbolic representation of classification process which was one important constraint in building up the classifier. A significant characteristics of ADTree was that; it was aimed towards a metric for confidence whereas this property was a nice feature in the domain of medical informatics. Another functional characteristics of ADTree was its capability of generalization of both voted decision stumps and voted decision tree as well. Although ADTree demonstrates itself to be a successful classifier that can combines the nice features of decision trees with the predictive accuracy of boosting algorithm emanating into a system with an interpretable classification rules. However Holmes et al., (2002) pointed out that the original formulation of the ADTree was restricted towards the binary classification problems limiting its general scope. Holmes et al., (2002) extend ADTree classifier (Freund and Mason, 1999) into a multi class problem handler algorithm. They evaluated numerous wrapper methods in order to introduce and extended version of ADTree. The strategy was to split the classification problem into several two-class (binary class) problems by means of adapting the multiclass LogitBoost and AdaBoost. Apart from multi class benefit, the other improvement in the LADTree in comparison to basic ADTree was its comparable accuracy but with significantly smaller size of trees.

2.1.9 J48graft

J48graft is a variant of C4.5 implemented in weka (Hall et al., 2009). J48graft was introduced by Webb in 1997 and later on improved in 1999. Grafted trees are built in two simple steps. The first step creates an initial tree. The second step complete the tree. Grafting is a technique for constructing ensembles of decision trees, where every tree is a grafted tree. It was reported that Grafting is comparably superior to Bagging (Rodrıguez and Maudes, 2006). Another advantage of grafting is that grafted trees are generic tree classifiers which can be easily coupled in ensemble techniques such as Bagging and Boosting. However it was noticed by Rodrıguez and Maudes (2006 that the induction of grafted tree in boosting is a function of variance. In J48graft, the modification in C4.5 is around pruning or un-pruning strategy which is termed as grafting the tree. As C4.5 was suffering from large number of tree which causes overfitting issue in learning of training data. This problem was solved by J48graft algorithm. It adds nodes to an existing decision tree with the underlying objective of reducing the prediction error. It considers one set of training data limited to only leaf nodes belonging to the initial decision tree. The pruning strategy is based on the set of training instances which fail at most one test on the path to the leaf node. Prior to the emergence of J48graft, C4.5X was introduced which was the first decision tree grafting algorithm; albeit it was not given the status of a practical machine learning classifier because it was introduced with the objective of tweaking the utility of Occam’s razor (Webb, 1997). However, it opened some new concepts and ideas for creating a practical learning algorithm namely C4.5+. The algorithm C4.5+ was shown to improve classification accuracy in comparison to conventional C4.5 in the wide range of domain of application. However, there were still room to reduce the bias and error present in C4.5+. Keeping in view of this motivation, J48graft significantly reduce the variance, although bias was not reduced much but time complexity was improved because J48graft is superior in time complexity as compared to earlier grafting algorithm.

2.1.10 BFTree

Best First Tree (BFTree) classifier introduced by Shi (2007). It employs binary split for categorical variables and numerical variables as well. The missing values are handled by fractional instance strategy. C4.5 algorithm introduced by Quinlan (1993) gained its popularity as a standard decision tree algorithm. In standard tree algorithms including C4.5 and its variant, the expansion dimension was depth first order. Shi (2007) noticed that there is margin of improvement if depth first search is replaced by best first search. In best first learner, the best node is taken first to be expanded. Moreover, the definition of the best node lies splitting criteria which eventually results in diminution of impurity measure. The standard metrics for measuring the impurity includes Gini Index, Entropy or Misclassification error. Shi (2007) pointed out that although the resulting tree might be same upon being fully expanded. But the order in which it is constructed is different which plays an important role in controlling the overfitting problem of the tree. Moreover, it was also observed that at the full growth stage of the tree, some arcs are not exhibiting correct orientation towards the underlying information contained in the domain of interest (Shi, 2007). This problem of overfitting was due to the noisy elements present in the dataset. Pruning is a useful solution In the direction of avoidance from this overfitting. Pruning discards those sections of the tree which are not helpful in the predictive nature of the tree. In BFTree new pruning strategy was introduced during node expansion process. This new pruning strategy enumerate the expansions being performed on cross validation. In BFTree, two pruning techniques were introduced, one works in a pre-pruning fashion while the other pruning technique employed in a post-pruning manner. The minor difference between both of these twin pruning methodologies was that in first technique namely best-first-based pre-pruning, the pruning halts splitting when met an increase in the cross-validated error. Whereas in the second technique known as best-first-based post-pruning operates on fully developed tree followed by the process of trimming of unwanted branches based on the cross-validated error.

2.2 Neural Network Classifiers

Neural network classifiers are stemmed out from complex functioning of human brain. A human brain contains more than ten billion neurons which are all interconnected in a peculiar fashion. The human brain with the help of these neurons can perform computationally composite, multifarious but demanding perceptual tasks such as recognizing objects, sound and many more. The human brain is capably of postulating parallel processing. Inspired by the amazing human brain, the researchers devised a breed of classification systems known as neural network classifiers. In this study, we made comparison to two most notable neural networks and one regression model. The detail of which is given in next sub section.

2.2.1 Multilayer Perceptrons

A neural network is a model formulated by an interconnected group of artificial neurons. This neural network employs a computational model for the purpose of information processing. Multilayer Perceptron (MLP) was introduced by Simon and Berger (1998). It belongs to the class of feed forward back propagation neural network classifier. MLP is considered as mostly notable neural classifiers in machine learning. In fact MLP has established its place as a de facto standard in the domain of pattern recognition as well (Vigdor and Lerner, 2006; Martins, Pires, Pires, 2007; Kyperountas, Tefas, Pitas, 2007). Some of its characteristics include that MLP can be drawn manually or by means of algorithm or a combination of both. It has practical application in solving problems stochastically such as complex problem of fitness approximation. Another characteristics of MLP is that it can receive any suitable modifications during the course of training session. In weka implementation, the nodes in MLP are sigmoid; however these nodes becomes un-threshold linear units in case the target class is comprised of numeric data. Distinguishing it from the conventional statistical inspired classifiers, MLP has the capability to learn without the pre requisite of prior knowledge about the probability distribution of the dataset. In MLP, number of network outputs is equal to the number of classes and number of network inputs is equal to the number of attribute variable. Each output neuron denotes one candidate class accompanied by the highest valued output that is considered as the network prediction. This technique in general is known as 1-of-n output encoding scheme (Mitchell, 1997).In MLP system, the input signals are weighted and summation function is applied to them. An extra connection weight is referred as the threshold neuron, all of these are applied to weighted sum result. This gives a linear combine output as shown by the following equation.

$$y = \sum_i w_i u_i \text{..........................}(2.4)$$

$$h(y) = \frac{1}{1 + \exp(-y)} \text{..........................}(2.5)$$

Where u_i is the ith input to the neuron while w_i is the connection weight for the input u_i. Furthermore, u_0 is taken as value of -1 and threshold is represented by w_0. The neuron output $h(y)$ is the output obtained from the activation function. The output signal obtained from every neuron is restricted by means of logistic sigmoid function.

2.2.2 RBF Network

Radial Basis Function (RBF) Network is the implementation of normalized Gaussian network. The basis function is determined by the notable clustering algorithm K mean. The basis function is fitted at the top of the system which eventually proceeds for learning regression model. In case of discrete class variable, the logistic regression is used while linear regression is used in case of numeric class variable. RBF and Multilayer Perceptron network both are most widely used neural network system. RBF has been implemented in function approximation and pattern classification (Mao, 2002). Various particular characteristics of RBF network have been reported (Mao, 2002). Firstly, it holds the globally optimal approximation attributes. Secondly RBF exhibits appealing classification capability, thirdly its learning procedure can converge sharply and last but not least, it is an optimal neural network which can realize the mapping function in the feed-forward neural network system. Furthermore, RBF standardizes all numeric variable to zero mean and unit variance.

2.3 Bayes & Linear Classifier

In machine learning, some classifiers uses feature vectors of a given instance specifying it with a certain weight vector. The relationship between feature vector and wait vector is obtained through simple dot product. The category which is obtained with best score is marked as successful class. Such classifiers basically depend on a score. Logistic regression which is implemented in weka in the name of Logistic package is also included in our comparison report in next chapter.

2.3.1 Logistic

Logistic Regression was originally introduced by Le Cessie and Van Houwelingen (1992). The weka implementation is somewhat different from the original version. We in this study have used the weka implementation; hence it is quite apropos if we discuss the prime features of weka implementation of Logistic Regression. It is a classifier for building and employing a multinomial logistic regression model along with a ridge estimator (Hall et al., 2009). Let there are k number of classes, n number of instances and m number of feature attributes are given, the parameter matrix (here in termed as B) can be enumerated as $m \times (k-1)$ matrix (Hall et al., 2009). The probability of jth class with the last class with exception can be enumerated as:

$$P_j(X_i) = \frac{\exp(X_i B_i)}{\left[(sum[j = 1...(k-1)] \times \exp(X_i B_i)) + 1\right]} \text{..........................} (2.6)$$

In weka implementation of logistic regression, missing records are treated by means of replacement for which a filter “ReplaceMissingValuesFilter” is used. Moreover, nominal variable are transformed into numeric by applying “NominalToBinaryFilter”.

2.3.2 Naïve Bayes (NB)

Naïve Bayes (NB) is one of the earliest version of the robust breed of classifiers based on Naïve Bayes theorem.

For a sample s with m number of features $\{g_1, g_2, g_3, ...g_m\}$, the posterior probability belonging to the lass h_k can be expressed as $p(h_k \mid s) \propto \prod_{i \in s} p(g_i \mid h_k)$ where $p(g_i \mid h_k)$ are conditional density distribution which is estimated from training samples. Although, it holds the independence assumption, it usually exhibit significant classification accuracy. This algorithm was introduced by John and Langley in 1995. They addressed an important issue related to “how to deal with continuous dataset”. Before introduction of this version of classifier, continuous variables were treated either through discretization or treating each variable as Gaussian distribution. John and Langley (1995) argued that every variable can not be treated this way rather kernel density estimation can give sound solution. The assumption of this non parametric density estimation was used in the Naïve Bayes theorem and empirically it was shown that in many particular situations, use of kernel estimation may lead to better results. Furthermore, large reductions in error on several natural and artificial data sets was reported. This ultimately place the NB as one of the impressive classifiers. However, later on many advanced discretizion technique emerged. A good survey in this regard is presented by Yang, Webb and Wu (2010) for a look at a comprehensive introduction to this aspect of data processing with an extensive taxonomy of discretization techniques along with analysis of major discretization methods.

2.4 Analysis

We proceed the analysis of the above state of the art classifiers in two dimension. The first analysis is a typical analysis in which best classifier for yielding a better accuracy is identified. The figure 2.1 and 2.2 are representation of this kind of analysis. We first obtain fifty natural dataset from UCI (Frank and Asuncion, 2010). These datasets were quite diversified in their specifications. Figure 1 is indicating that J48graft deliver best performance by averaging of accuracies (77.93%) of all dataset followed by Random Forest (RF) which gives a close margin of 77.86%. J48 which is an earlier version of J48graft also deliver nearby results which is 77.6%. On the other hand, the worst accuracy average is forwarded by Decision Stump (DS) and its implementation with ensemble classifier Ada Boost Decision Stump (AB DS) with values of 58.93% and 61.4% respectively. Apparently, these figures are quite convincing with a possible verdict in favor of J48graft and J48 as well. However, if we observe the results from a different angle (see figure 2.2) then comparative results are delivering a different view.

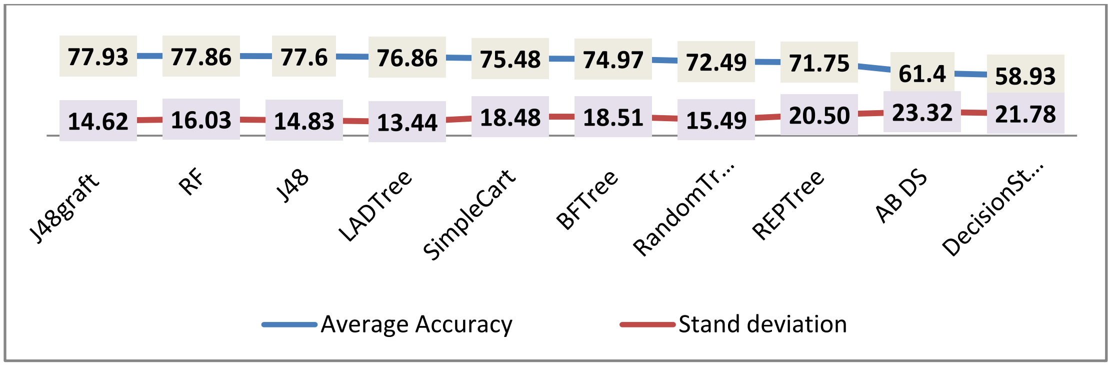


Figure 2.1: Comparison of tree classifiers for average accuracy over 50 natural dataset

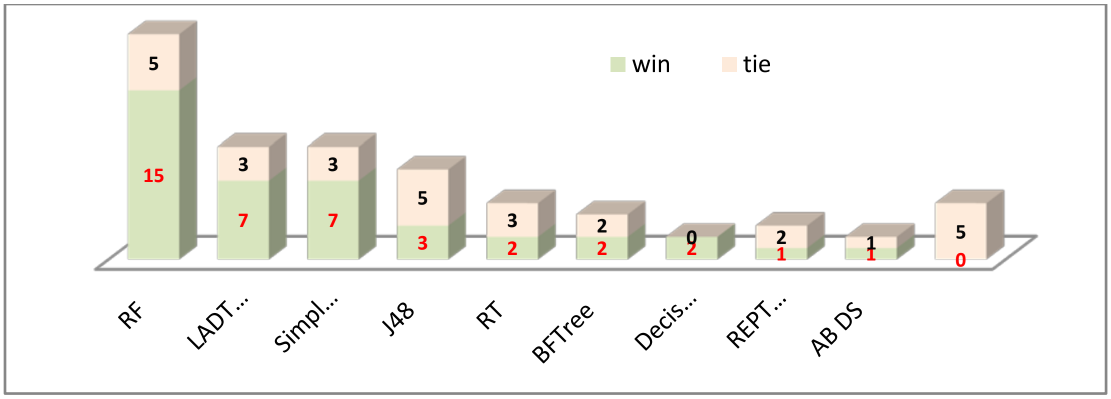


Figure 2.2: absolute win and tie accuracy for tree classifiers among 50 natural dataset

As illustrated by the figure 2.2, RF shows the best overall performance as it gives unmatched highest accuracy for 15 datasets while in 5 datasets its performance was also highest but in tie with some of the other classifiers. When we observe the performance of the J48graft, it only gives highest accuracy for 5 dataset but in tie with some other classifier; nonetheless, it gives not a single absolute (unmatched) highest performance. This delivers a notion with conclusion that one can’t judge the comparison of performance among classifiers based on any one of the criteria either by average of accuracy or else picking the count of highest unmatched win and tie win. However, the level of confidence for any verdict about the ‘best’ classifier can only be endorsed when a classifier outperforms its peer systems in all of these three aspect as shown in the figure 2.1 and 2.2. Keeping in view of the general overview of both of the figures one can suggest that RF is comparatively a better classifier albeit it followed by J48graft in average accuracy but the comparison was too close. It means the standard deviation of the accuracy for all of the fifty dataset for RF was higher as compared to relatively uniform accuracy of J48graft. It gives us another gross evidence that difference of average accuracies between two classifiers at large number of dataset result in approximate proportionate difference of respective standard deviation of both of the classifiers. However it is conditioned with the significant difference of average accuracy.

The figures 2.1 and 2.2 were meant for a notion of simplistic and straight forward rather trivial comparison among tree classifiers. While there is another dimension for analyzing these classifiers which we shall term as meta characteristics of a dataset. These include simple characteristics such as number of attributes, class count and size of cases. Moreover, some advanced characteristics of dataset such as Entropy, Mutual Information and one of our proposed information theoretic measure Polarization Measure. This kind of analysis can be useful for the non expert users for their selection of a classifier to support automating parameter optimization, model selection with a prior but generalized information obtained.

Before we analyzed, it is mandatory to pre process or transform the data, there are many *transformations* applicable to a variable before it is used as a dependent variable in a regression model. These transformations can not only restrict towards changing the variance but may incur alteration the units of variance to be measured. These include deflation, logging, seasonal adjustment, differencing and many more. However, the nature of data in our case motivates us to adopt the normalization transformation of the accuracy measures and the specific characteristics for which analysis is required. Let x_i denotes the accuracy of ith dataset by any classifier then the normalized accuracy y_i can be obtained by the equation as below:

$$y_i^n \leftarrow \frac{x_i^n}{m} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.7)$$

where

$$m = \max \arg(x_i^n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.8)$$

The next step is to get a pair wise list with sorting performed on y_i such that we denote the sorted list as $\overset{\leftarrow}{y}$.With the application of these normalization, a set of normalized characteristics was prepared which was later used to generate a regression model. A linear regression model is quite useful in order to express a robust relationship between two random variables. The linear equation of regression model indicates the relationship between two variables in the model. Y is regressand or simply a response variable whereas X is regressor or simply an explanatory variable. The output regression line is an approximate acceptable estimation of the degree of relationship between variables. One important parameter in linear regression model is co efficient of determination also known as R-squared. The closer this value to 1, the better the fitting of regression line is represented. R-squared dictates the degree of approximation of the line passing through all of the observation.

Wolpert and Macready (1997) stated in their *No Free Lunch Theorem*, that no machine learning algorithm is potent enough to be specified outperforming on the set of all natural problems. It clearly points out that every algorithm possess its own realm of expertise albeit two or more techniques may share their realm in partial.

The significance of R-squared is dictated by the fraction of variance explained by a data model but question arises what is the possible relevant variance requiring a suitable explanation. Unfortunately it is not easy to fix a good value of R-squared as in most of the cases it is far off to get a value of 1.0. In general it is assumed that a value greater than 0.5 indicates the noticeable worthiness of the model. However, still it is a matter of choice as in case of comparison between various models (such as in ours) the comparison of R-squared counts more.

Some related work includes the distribution of classes expressing a robust BBN in the literature. Rajaram et al., (2011) presented an extended version of traditional NB algorithm named NB+ using class distribution as a significant factor. They argued about the issues in improving the accuracy of classifier while handling the cases with same class probabilities. NB+ method relies on a partial matching method in which method of closely matching tuples is employed. Here the tuple indicates a minimal set of attributes (at least one) to be present in order to match the test sample with the training data. The author argued that the introduced technique is quite suitable to the situations where distribution of class probabilities are same. In order to validate their claim, they show eighteen dataset with improved classification accuracy as compared to simple NB. There are some observations which are quite arguable. Firstly among all of the eighteen dataset used in the experimental evaluation, except IRIS, no dataset contains same class distribution. IRIS dataset contains three classes with strictly equal probability of 33% (50/150) per class. However, the rest of the dataset does not contain same class probabilities; rather some datasets like chess contains two classes with counts of 2839 for class of ‘f’ and 357 for class of ‘t’ which is a heavily tailed distribution. In fact, all of the eighteen dataset except IRIS contains variety of class distribution ranging from normal to non uniform distribution. Secondly majority of the dataset which were shown with improved results, contains binary classes restricting the scalability of application of NB+ to other types of data. Thirdly, the application of the NB+ was equivalent to feature selection before the application of Naïve Bayes because the influence of the minimal set of variables on the class label in the training set was achieved prior to achieving the final structure learning.

Table 2.1 and 2.2 shows various linear regression model using specific meta classifiers. The cells in red colors shows the substantial model fitting. These curve fitting were tested with many flavors of regression models ranging from 1st degree to 10th degree order polynomial, 1st order logarithm to 5th order logarithm, polynomial inverse and a lot of special cases data fitting model provided in the commercially available tool DataFit (2013). We noticed that the best curve fitting was found for tenth order degree polynomial regression model. There were only two models (marked by asterisk sign) whose best data fitting was noticed with 9th degree order fitting.

Table 2.1: Polynomial regression of tree classifier accuracy using simple and information theoretic meta characteristics

	Attrib	Classes	Cases	PM	MI(log)	Joint
BFTree	0.398	*0.215	0.217	0.254	0.161	0.69
J48	0.064	0.197	0.268	0.247	0.179	0.22
J48graft	0.059	0.194	0.267	0.24	0.175	0.23
Decision Stump	0.222	0.798	0.626	0.875	0.206	0.75
LADTree	0.034	0.209	0.307	0.345	0.193	0.245
Random Tree	0.164	0.282	0.252	0.291	0.18	0.32
Simple Cart	0.388	0.221	0.19	0.24	0.154	0.47
REPTree	0.216	0.346	0.23	0.308	0.121	0.58
Random Forest	0.099	0.224	0.295	0.287	0.138	0.3
Adaboost Decision Stump	0.212	0.775	0.617	0.836	0.249	0.73

Breadth First (BF) Tree, Random Tree, Simple Cart, REP Tree and Random Forest can be comparatively explained by joint Entropy. We calculated the average joint entropy of each attribute with class attribute, hence the final score is indicative of a score of entropy towards the class variable. The root cause lies in the splitting criteria which is characterized by entropy inspired measure. J48 and J48graft both can be explained by number of cases. In simple meta characteristics, only cases meta features was found relatively better although its R square was below 50.

Apart from this we also devised a measure Polarization Measure (PM) which was used in the experimental work in chapter 3 and 4. The mathematical detail of plugging PM into BBN (developing into NP-FiLM) is detailed out in next forthcoming chapter. Surprisingly, we noticed that this measure incur significant R-squared value in case of Decision Stump (DS) and its implementation with Ada Boost DS (AB DS). The R-squared value was 0.875 and 0.836 respectively. It clearly indicates that the classification accuracy of both of these classifiers can be greatly predicted a prior by using PM. It is noticeable that no other meta feature deliver this level of R-squared confidence of determination. Furthermore, LADTree was also found with highest value of R squared using PM metric.

Table 2.2: Linear regression of function classifier accuracy using simple and information theoretic meta characteristics

	Attributes	Classes	Cases	PM	MI
Logistic	0.233	0.324	0.304	0.27	0.163
Multilayer Perceptron	0.286	0.223	0.264	0.27	0.156
RBF Network	0.131	0.248*	0.316	0.25	0.189

Mutual Information (MI) which is an information theoretic measure. MI is basically an intersection of entropy of two features. MI strictly defines the mixed relationship of two variables by which both of them are bound to each other. However we noticed that it did show up better as compared to other meta characteristics. We see that basic meta characteristics were found comparatively better in cases of function classifier. However this can not be generalized as PM measure also gives close results for function classifiers as shown by the table 2.2.

When we compare between simple and information theoretic meta characteristics, then we can draw conclusion in general that information theoretic meta characteristics are giving relatively better performance in line fitting model to predict the model accuracy. In the last, we can establish from the simple analogy that smaller problems require comparatively lesser time for their solution. Integrating this analogy, it was observed that Random Forest and Breadth First Tree are the slowest algorithms, whereas Bayesian Networks, simple Naïve Bayes and Random Tree are significantly faster. LADTree also exhibit good performance. J48 and J48graft both also show satisfactory execution time, which is a fine indicator taking into account their non trivial error rates.

Chapter 3

NON PARAMETRIC FACTORIZED LIKELIHOOD SCORING FUNCTION

In the previous chapter, some well renowned decision tree classifiers were examined. In this chapter, we shall present a novel scoring function Non Parametric Factorized Likelihood Metric (NP-FiLM) with its mathematical rigor. The NP-FiLM measure introduced in this chapter carries a sound theoretical foundation and formal analysis of its mathematical properties. An experimental comparison with existing measures that are obviously closely related has also been forwarded. The empirical study presented in the chapter is extensive but interesting.

3.1 Towards a Novel Scoring Function

A scoring metric in general can be expressed as the sum of local score that depends only on every variable and its parental nodes. With a given dataset D, parent set Π for n feature fi, Ψi is the score for each node. The cumulative scoring criteria Ψ can be expressed formally:

$$\Psi(B,D)=\sum_{i=1}^{n}\Psi_i(\Pi_{fi},D)....................(3.1)$$

The scoring function in general are based on Log likelihood drawn from the dataset. The Log Likelihood (LL) which can be described as the log probability of dataset D given network structure G as shown by equation 3.2.

$$LL(G\,|\,D)=\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right)...............(3.2)$$

Where N_{ijk} indicates that ith feature is instantiated with kth state along with the jth state of qth parent of ith feature. A very simple and easy to calculate, hence a decomposable format. Moreover, it can be noted that adding an arc to a network always tend to increases the probability of likelihood of the underlying network. We can raise a proposition.

3.1.1 Proposition 1

Let Ɱ denotes a Bayesian network over the query variables X. Moreover, it is also assumed that Bayesian network parameters Φ_M are locally and globally independent. Then the size of the model is a function of number of links Ḻ and distinct states Ṩ such that

Size (Ɱ) = *f*(Ḻ, Ṩ)………………..(3.3)

A simple decomposition will result in

Size (Ɱ) = $\sum_{x \in X} (|\, pa(x) \,| \times |\, x \,|)$..............(3.4)

Which means the complexity of a model Ɱ can be found by the product of count of parents and states of a node variable.

3.1.2 Lemma 1

Let Ɱ be a Bayesian network being represented by the set of query variables X. The optimized and most representative model $Ḿ_X$ of the underlying dataset contains only essential links. It can be shown that no other network M_X can have lesser number of links or say smaller size of the model.

Proof

Let Ɱ be any ordinary model which denotes parameter distribution say P_{UX}. On the other hand, $Ḿ_X$ is an optimized model. It can be observed that whenever two nodes x_i and x_J are linked which increases the accuracy of the model. If these are connected in model Ɱ, they must be present in $Ḿ_X$. However, if there is a situation where the size of Ɱ is smaller than size of $Ḿ_X$ then it is so because some links in Ɱ carries the opposite direction to that of the corresponding optimized model. It justifies the search for a minimal model. If the network is a Bayesian network, and containing only essential links then the model is optimized model.

Obviously, any extra arc which is not causing any increase in the information of the structure must be ignored. The extra arc is prone to give rise two issues. First issue is problem of overfitting during training phase, eventually poor accuracy in testing phase might be observed. Secondly, this enhances the complexity of the network. Computational complexity will be increased during inference (prediction) phase given a dense network. The solution to this problem appears in form of addition of penalty factor. The term penalty factor has its notion in penalizing the complexity of network structure. That is why, a complex network may bear high Log Likelihood value but the degree of penalty factor can adjust the score to be equivalent to a less complex network. The scoring function which carries penalization can be generally expressed in a following non decomposable notation.

$$SF(G\,|\,D) = LL(D\,|\,G) - \sum_{i=1}^{n} peanlityfactor(X_i, G, D) \text{..............}(3.5)$$

Several well known scoring function which we discussed in previous chapter belong to penalized scoring function. The only major difference is the magnitude of the penalty factor while they incur similar overhead for memory consumption (Liu, Malone and Yuan, 2012). However, we have investigated this issue from a different angle. In studying the structure learning, there is a general principle of inductive learning introduced by William Ockham (1285-1349) that select the simplest hypothesis such that the hypothesis is consistent with the underlying observation. It has been reported that this principle has a vivid rationalization in structure learning using BBN (Jensen and Nielsen, 2007). Proceeding with this notion of simple hypothesis, let F and C are two features such that C is a class feature and F is a non class feature. We are to find out a metric of relationship between two features which can deliver the answer of how much class feature is explainable by the non class feature F. Let F is the realization of a distinct states given C contains b number of unique states.

$$F = \{ f_i \mid i = 1,...a\} \text{.................}(3.6)$$

$$C = \{ c_j \mid j = 1,...b\} \text{.................}(3.7)$$

The above is a simple case of point estimation of learning where there is only single input variable with a single target feature (class variable). In fact point estimation is the base case for numerous learning models which gradually developed towards inclusion of other input variables. In this case, a learning model predicts a value for the target feature class for all of the sample instances. The prediction minimizes the error relies on the error which is being minimized. The joint probability state between both of these feature variables can be described as:

$$\delta(C,F)=\sum_{c}\sum_{f}P(C=c,F=f)\text{................}(3.8)$$

The above joint probability is not normalized always; thus it is not resulting into value of 1 to compare with other pair wise values. Such probability distribution are termed as potentials. We can write the potential ξ formally as:

$$\xi_{C,F}=\sum_{c}\sum_{f}P(C=c,F=f)\text{................}(3.9)$$

Our aim here is to maximize the discriminant objective function out of this potential. We incur a change in this potential such that

$$\xi_{C,F}=\sum_{c}\left[\max\arg\left(\sum_{f}P(C=c,F=f)\right)\right]\text{................}(3.10)$$

The above is the discriminant prior over simple point estimation which in fact serves as another measure of coherence between two relations when viewed from the information theory perspective. This basic unit can be integrated into a well behaved measure spanning over relationship of set of features versus class variable.

3.1.3 Lemma 2

The discriminant joint probabilities obtained from the potential in the equation 3.10 may lead to turn into maximum a prior probabilistic inference for a simple point estimation case in structure learning.

Proof

We re write the equation 3.11 such that

$$\xi_{C,F}=\sum_{c}\left[\max\arg\left(\sum_{f}P(C_c,F_f)\right)\right]\text{................}(3.11)$$

Let $\vartheta(F)$ denote the marginal probability of the feature. The potential shown in the above equation can be converted into conditional probability by placing the marginal probability as the denominator factor in the above equation such that

$$\lambda_{C,F} = \sum_{c}\left[\max\arg\left(\sum_{f}\frac{P(c,f)}{\vartheta(F)}\right)\right]\ldots\ldots\ldots\ldots\ldots\ldots(3.12)$$

The simple point estimation potential (see equation 3.11) is decomposed into conditional joint probability factor. However, we are not dealing in ordinary cases of single input features. It must be required to generalizing it to a dataset with more than one non class features.

3.1.4 Lemma 3

NpLFM is a decomposable scoring function.

Proof

While generalizing NpLFM, we have n number of non class feature variables and a single class variable within the dataset D. We can express easily reduce this simple point estimation into a generalized maximum a posterior inference notation as below:

$$NpLFM(D,G) = \sum_{i=1}^{n}\max\arg(X_i, Pa(X_i), C, D)\ldots\ldots\ldots\ldots\ldots\ldots3.13$$

A scoring function is decomposable if its expression is convertible to a sum of local scores, where local score refer to a feature variable in the family of feature variable in pursuit of drawing graph G. the simple calculation between two feature variable is shown in 3.12. An extended version of this equation can be expressed as $\sum_{j=1}^{qi}\sum_{k=1}^{r1}N_{ijk}$ Where i is feature iterator, j is parent iterator, k is feature state iterator and c is class iterator. If we include the factor of class variable, a minor change will be developed into $\sum_{j=1}^{qi}\sum_{k=1}^{r1}N_{ijck}$. Plugging this value into equation 3.13, we can express as

$$NpLFM(D,G) = \sum_{c} \left[\frac{1}{|C_c|} \max\arg \left(\sum_{j=1}^{qi} \sum_{k=1}^{r1} N_{ijck} \right) \right] \text{................(3.14)}$$

If we introduce a link between Xi and Xj pointed towards Xj, then only the local value of NpLFM will be altered for the purpose of evaluating whether this addition gives any significant improvement in the structure being represented by G such that.

$$\Delta(X_p X_q) = \left[\left\langle \sum_{i=1}^{n} \max\arg(X_q, Pa(X_q), C, D) \cup \{X_p\} \right\rangle - \sum_{i=1}^{n} \max\arg(X_q, Pa(X_q), C, D) \right] \text{......3.15}$$

Hence the it can be concluded that NpLFM belongs to the class of decomposable scoring function. The decomposition property is quite useful when searching mechanism has to calculate net score over addition or deletion of an arc in G.

We revive our motivation for the introduction of new scoring metric, according to which the increase in the potential candidate for the addition of the node found in a queue, the number of possible configuration over Xi will also get large. From this large number of factors, only those factors will be selected which has more contribution towards explanation of any class member. However, it also beget some critical observation. We consider feature set and class variable as defined in equation 3.6 and 3.7. We consider the last feature in ordered list. Surely in a non augmented network, this must be linked to class variable with a specific discriminant value of joint probability. An inclusion of the next feature in the set of its parent list will be restricted by a higher value of discriminant value. However, as the new node is linked, such chances are quite narrow unluckily, because the factor joint probability distribution will start thinning with the increase of new parental value. It means, in a randomly ordered set of features, there is very little chance that structure appears to be other than simple Naïve Bayes. We already have illustrated that simple Naïve Bayes is suffering from under fitting. The question arises how to tackle this issue. A clear solution lies in the intelligent ordering of the variables prior to application of search and score bound heuristics.

3.1.5 Proposition 2

If the Feature set is denoted by $F = \{f_1, f_2, f_3, ... f_n\}$ then ordering weight of any feature will be determined by weight factor shown in equation 3.14.

$$\omega_F = \lambda_{C,F} - \lambda_{C,F} \text{.........................(3.14)}$$

The terms $\lambda_{C,F}$ and $\lambda_{C,F}$ plays the role of existence restrictions. We shall consider both of them as existence restrictions such that $(F,C) \in \lambda_{C,F}$: the link F→ C explains the discriminant objective with respect to the class and the and $(F,C) \in \lambda_{F,C}$: the link F← C means the discriminant score with respect to the feature. In our earlier research (Naeem and Asghar, 2013b), we highlighted the correct topological ordering between two features. This was shown by an earlier version of the proposed scoring function in which we highlight that majority of the scoring metrics can't precisely capture the casual relationship between two variables in pursuit of true topology in numerous situations; this ultimately leads to the selection of potential neighbor and parents becoming unreasonable. However Integration to Segregation (I2S) is capable of rightly identify it in majority of the cases as compared to BIC, MDL, BDeu, Entropy and many more. Moreover, Madden (2003) described that a structure in which class node is placed at the top most may lead to higher predictive accuracies. This type of scheme was termed as ''selective BN augmented NBC'' (Madden, 2003). Hence the later score value must be eliminated from the first value which will result into a weighted score vector as shown in the equation 3.14.

$$\lambda_{F,C} = \sum_{f} \left[\max\arg \left(\sum_{c} \frac{P(f,c)}{\vartheta(C)} \right) \right] \text{................(3.15)}$$

See equation 3.12 and 3.15 for detail of equation 3.15. A function for simple descending order is applied to the weights achieved from the equation 3.14 which results into an ordered list of input variables. $\overset{\leftarrow}{F} = \{\overset{i}{\omega}_f \mid i = 1...n\}$................(3.16)

Plugging this ordered set into the equation 3.14 will give result in

$$NpLFM(D,G,\overset{\leftarrow}{X}) = \sum_{c} \left[\frac{1}{|C_c|} \max\arg \left(\sum_{j=1}^{qi} \sum_{k=1}^{r1} N_{ijck} \right) \right] \text{................(3.17)}$$

3.1.6 Lemma 4

The ordered set initialized by an intelligent heuristic may convert NpLFM into a well behaved scoring metric.

Proof

Let us consider a set of n un-sorted features F = {f1,f2,f3..fn}. We start from any of the succeeding feature say jth feature fj such that it lies somewhere in the trivial un-ordered list such as $\{\rho, f_j, \varsigma\} \therefore \rho \cap \varsigma = \phi$ where ρ is the set of predecessor and ς is the set of successor nodes. We know that K2 adds incrementally for a node as its parent from a given ordering whose addition possibly increment the score of the resulting structure. K2 can chose any of the parent set before fj. If a feature fs exist such that it can significantly contribute towards score of structure, then following expression must hold $\forall(f_s, c) \in \rho,\ f_s \to f_j : true$ but if the expression $\forall(f_s, c) \in \varsigma,\ f_s \to f_j : false$. A careful consideration of expression of NpLFM (see equation 3.17) we can frame out the following characteristics possessed by the introduced scoring metric.

1. No penalty factor
2. Non parametric
3. Scalable to large dataset
4. Decomposable
5. Value increases only on adding those nodes which contribute information towards structure being built, otherwise halts.

It is a known fact that parametric approaches are statistically less robust. This lack of robustness has its roots in requiring more training samples for the same level of discriminating power as compared to non-parametric ones. On the other hand, the non parametric approaches holds strong assumptions on the formation of the underlying model, and are thus capable of expressing itself with less number of parameters, enabling an easy estimation in number of training samples. NpLFM holds no prior information factor as well as no penalty factor. Contrary to it, the selected parameter value of alpha which controls the penalty factor in BDeu greatly influence the BDeu's performance. In other words, it can be stated that BDeu is significantly dependent on the specific value of alpha parameter; yet it is quite hard to predict its appropriate value a priori (Liu, Malone and Yuan, 2012). For some datasets, Average Hamming Distance (AHD) metric was found in consistent with value of alpha when sample size was increased in a particular fashion. Usually AHD get decreased as the value of alpha is increased. But unluckily this result was not generalize able as this specific trend was restricted to only a few datasets only. Secondly, the author produced sample of various sizes based on the gold standard. Such dataset may also posses peculiar fashion in support of alpha or against value of alpha. Moreover, Liu, Malone and Yuan (21012) concluded that performance of BDeu is highly dependent on the selected parameter specially value of alpha and in fact there is no specific mechanism found to estimate the most appropriate value of alpha in prior.

3.2 Information Theoretic Interpretation

Before describing empirical results, we illustrate that NP-FiLM has an interesting mutual coherence interpretation with its roots in information theoretic elucidation. We revive some basic concept of entropy and mutual information and their mapping to each other. Let X is a feature and C is a class then joining entropy is a function conditional entropy of X given C and vice versa and Mutual Information (MI) as given by the equation 3.18

$$H(X,C) = H(X \mid C) + MI(X,C) + H(C \mid X) \text{................}(3.18)$$

The conditional entropy of X given C can be split into two components, one component maximizes the discriminant function while the other component is trivial part which can be ignored. In fact the notion of NP-FiLM has its roots in splitting of conditional entropy of X given C as shown by the equation 3.19

$$H(X \mid C) = NPFiLM(X \mid C_m) + NPFiLM(X \mid C_n) \text{................}(3.19)$$

This establish the relationship of NP-FiLM to conditional entropy. On the other hand, the conditional entropy itself is a component of entropy of single variable (X in this case) and the mutual information between two variables (X and C). This is expressed by equation 3.20

$$H(X \mid C) = H(X) - MI(X,C) \text{................}(3.20)$$

From equation 3.19 and 3.20, one can easily conclude and establish the relationship of NP-FiLM and MI indicated in equation 3.21.

$$H(X) - MI(X,C) = NPFiLM(X \mid C_m) + NPFiLM(X \mid C_n) \text{................}(3.21)$$

Notice that the proposed measure can be expressed by entropy and Mutual Information giving a notion that NP-FiLM has its explicit illustration in entropy and mutual information.

$$NPFiLM(X \mid C_m) = H(X) - MI(X,C) - NPFiLM(X \mid C_n)................(3.22)$$

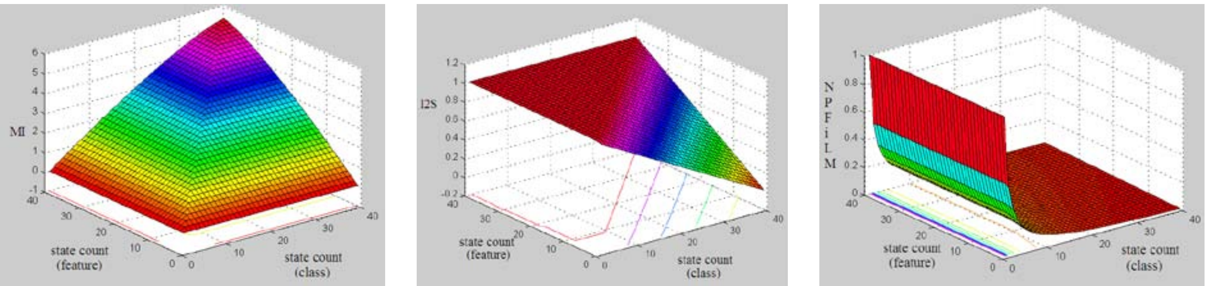


Figure 3.1: surface plot for joint distribution of two variables for MI, I2S & NP-FiLM

However, an interesting argument arises what is the impact of NP-FiLM in scoring metric as compared to Mutual Information and our earlier proposed scoring metric I2S. Figure 3.1 (left) indicates the surface chart of MI when number of states in both of the features are gradually increased. Notice that in MI, the value is always increasing behaving symmetrically while the relationship of two nodes in BBN is never yielding a symmetric accuracy. Secondly for a uniform distribution of two variables it only reaches its maxima in some certain cases when specific number of states in both of the features is achieved. On the contrary NP-FiLM (right figure), number of states plays an important role as the enumeration of class states increases, there is probability that its joint distribution will also get sparse and thinner in each individual distribution. Such well behaved phenomenon is also observed in the BBN classifier where the equal distribution of class node with respect to the other feature results in poor classification accuracy. Moreover, we also demonstrate the behavior of I2S, although the value indicated by the middle figure points out that it is also an asymmetric in behavior like NP-FiLM which is in quite harmony with graphical learning. However, it is not as well behaved as NP-FiLM. It is showing that a change in state count of the feature give a drastic change in the corresponding value of I2S while the state count change in class node is giving a slow change in the I2S value. Keeping in view of this limitation in I2S and MI, this phenomenon is adjusted in NP-FiLM to tailor it into a suitable scoring metric for BBN.

3.3 Benchmark Datasets

Table 3.1: Dataset used in this study

Dataset	Attrib	Classes	Cases	Dataset	Attrib	Classes	Cases
arrhythmia	280	16	452	labor	17	2	57
audiology	70	24	226	letter	17	26	20000
autos	26	7	205	liver-disorders	7	2	345
balance-scale	5	3	625	lung-cancer	57	2	32
breast-cancer	10	2	286	mfeat-fourier	77	10	2000
breast-w	10	2	699	mfeat-karhunen	65	10	2000
bridges_version1	13	6	105	mfeat-morphological	7	10	2000
bridges_version2	13	6	105	mfeat-pixel	241	10	2000
car	7	4	1728	molecular-biology_promoters	59	4	106
colic	23	2	368	mushroom	23	2	8124
colic.ORIG	28	2	368	page-blocks	11	5	5473
credit-a	16	2	690	pendigits	17	10	10992
credit-g	21	2	1000	postoperative-patient-data	9	3	90
cylinder-bands	40	2	540	segment	20	7	2310
dermatology	35	6	366	shuttle-landing-control	7	2	15
diabetes	9	2	768	sonar	61	2	208
flags	30	8	194	spect_test	23	2	187
glass	10	7	214	spect_train	23	2	80
haberman	4	2	306	splice	62	3	3190
hayes-roth_test	5	4	28	sponge	46	3	76
hayes-roth_train	5	4	132	tae	6	3	151
heart-h	14	5	294	tic-tac-toe	10	2	958
heart-statlog	14	2	270	trains	33	2	10
iris	5	3	150	waveform-5000	41	3	5000
kdd_synthetic_control	62	6	600	zoo	17	7	101

A number of benchmark datasets have been used for the evaluation in this study. These include dataset with binary classification problems as well as multivariate classification problems obtained from the UCI data repository (Frank and Asuncion, 2010). These dataset are processed into weka support format (arff) available at sears project (2013). These data sets were randomly selected so as to chose them from various real-world domain with varying characteristics. Table 3.1 is indicating an overview of these dataset in which attributes count, number of rows (cases) and classes are shown. It is preferred if we select dataset with variety of information under these categories to avoid any bias results in favor of a specific technique. None of the dataset was discretized prior to feeding in the weka package. However, weka itself discretize the continuous data using its default setting introduced by Fayyad and Irani (1993). The performance of the proposed measure used in introduced classifiers is measured by accuracy which is a function of True Positive Rate (TRR) and False Positive Rate (FPR). It is formally defined as the ratio of negative and positive instance correctly classisified (TP + TN) and enumeration of all classified instances as shown by the equation 3.18:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \text{....................(3.18)}$$

3.4 Empirical Validation

In table 3.2 to 3.4 and figure 3.1 to 3.7, one can observe the comparisons of NP-FiLM scoring function based BNN classifier with the other existing techniques. The results are tabulated showing ten tree classifier, simple NB, three function based classifiers and six well known scoring function based NB system. The scoring function comparison have been made using various parent values of 4, 3 and 2 (polytree) within the searching algorithm of K2 and hill climbing as well.

Table 3.2: Comparison with peer scoring function using K2 (max. parent : 4)

Dataset	NP-FiLM	Bayes	AIC	BDEU	MDL	Entropy	fCLL
arrhythmia	70.80	70.80	70.80	69.91	71.02	66.15	71.68
audiology	78.76	76.11	76.11	73.01	76.11	75.22	76.55
autos	80.98	80.49	74.63	83.90	74.63	79.02	79.02
balance-scale	72.64	70.88	70.88	71.84	72.00	70.88	70.88
breast-cancer	70.98	70.28	68.53	69.58	70.63	62.94	61.89
breast-w	96.71	96.71	96.85	96.57	97.00	96.42	97.14
bridges_version1	65.71	65.71	65.71	65.71	65.71	41.90	56.19
bridges_version2	63.81	64.76	62.86	64.76	60.95	41.90	54.29
car	91.61	90.80	92.65	90.80	85.71	88.25	88.25
colic	82.34	81.79	82.07	82.07	81.52	72.01	75.27
colic.ORIG	78.53	79.08	78.26	78.80	78.53	65.22	66.30
credit-a	85.94	85.07	85.51	85.80	86.23	81.16	82.17
credit-g	74.60	74.50	74.70	75.00	75.30	69.60	70.70
cylinder-bands	77.41	75.37	76.85	0.00	77.96	0.00	0.00
dermatology	97.54	98.09	97.54	97.54	97.54	89.62	93.17
diabetes	74.74	74.48	74.09	75.13	74.87	72.53	72.79
flags	61.34	57.22	61.34	57.22	62.37	35.57	59.79
glass	71.03	72.43	70.56	69.16	70.56	72.43	75.70

Table 3.2 (continue): Comparison with peer scoring function using K2 (max. parent : 4)

Dataset	NP-FiLM	Bayes	AIC	BDEU	MDL	Entropy	fCLL
haberman	73.86	72.55	72.55	72.55	72.55	73.86	73.86
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	85.03	84.35	85.03	84.01	81.97	84.01
heart-statlog	81.48	81.85	82.59	80.74	80.37	81.85	82.22
iris	92.67	92.67	92.67	92.67	92.67	90.67	92.67
kdd_synthetic_control	98.67	98.83	98.00	97.67	97.17	16.67	93.33
labor	94.74	92.98	91.23	89.47	91.23	87.72	87.72
letter	84.53	86.48	83.97	81.71	76.62	0.00	87.04
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	65.63	71.88	75.00	75.00	81.25	78.13
mfeat-fourier	79.85	80.25	80.15	77.80	78.05	76.75	77.75
mfeat-karhunen	92.75	92.95	93.15	92.10	92.05	85.70	88.15
mfeat-morphological	70.20	69.85	67.95	68.85	68.20	68.85	67.95
mfeat-pixel	94.75	94.55	94.00	93.55	93.40	0.00	92.85
molecular-biology_promoters	33.02	23.58	28.30	30.19	29.25	30.19	30.19
mushroom	99.74	100.00	100.00	100.00	99.99	100.00	99.83
page-blocks	95.47	96.46	95.30	96.33	95.63	96.62	96.35
pendigits	94.78	96.56	95.26	95.14	93.25	95.60	96.55
postoperative-patient-data	64.44	64.44	65.56	64.44	64.44	62.22	63.33
segment	95.32	95.28	94.85	94.63	91.39	94.85	91.69
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	86.67	86.67
sonar	77.88	78.37	77.40	76.92	79.81	75.00	78.85
spect_test	70.05	66.84	67.91	68.98	71.12	63.64	65.78
spect_train	68.75	67.50	67.50	63.75	68.75	58.75	53.75
splice	95.17	95.55	94.73	0.00	95.64	52.57	52.57
sponge	94.74	93.42	93.42	94.74	93.42	94.74	92.11
tae	47.02	47.02	47.02	47.02	47.02	47.02	47.02
tic-tac-toe	75.89	78.29	83.82	73.80	73.80	82.99	83.40
trains	60.00	50.00	60.00	70.00	60.00	60.00	80.00
waveform-5000	82.60	81.72	81.36	81.48	81.54	72.22	80.24
zoo	97.03	95.05	96.04	100.00	94.06	96.04	93.07
Average	**78.49**	**77.77**	**78.02**	**74.63**	**77.58**	**67.24**	**74.31**
NP-Film win/neutral/lose		**24/10/16**	**28/11/11**	**28/10/12**	**25/12/13**	**36/7/7**	**32/6/12**
absolute win/draw	**10/10**	**6/9**	**5/8**	**3/11**	**7/9**	**2/8**	**5/5**

In all of these experiments, ten fold cross validation was exercised. It means the dataset was divided into ten equal subset. There were ten sessions, all were run such that in each session, one subset considered as test data while the union of all other subsets treated as training data. At the completion of these sessions, median value of statistical results is considered as the final result of the classifier. There is some general explanation towards these tables of results. Each numeric value against a specific dataset indicates the accuracy of classifier dictated in the respective head column. At the last three rows of the table 3.2, one can observe the aggregated results. firstly the proposed measure is compared with every other classifier in terms of average. In second last row performance is represented in term of win/neutral/lose that NP-FiLM based BBN classifier wins or loses from the specific algorithm indicated in the corresponding head column. Whereas in the neutral cases, no significant statistical difference found in the results. That is, any other classifier exhibited statistically better than the proposed technique according to corrected t-test with $p < 0.05$ (Nadeau and Bengio, 2003). The simple t-test dictates that the samples are independent. However, because of the procedure of cross validation functionality, the sample instances are not independent. It gives high value of type 1 error if this assumption is generally ignored (that is, the test indicating, there is a difference between the tested technique while in fact there is not). The corrected t-test exercises a fudge factor to enumerate the dependence between sample instances which practically emanates into acceptable type I errors (Nadeau and Bengio, 2003). Moreover, in the result tables the highlighted yellow color points out that this algorithm performed statistically better than all other competitive classifier according to corrected t-test with $p < 0.05$. In all the other cases, it can be noted down that there is no significance statistical difference between the results drawn. In the last row of the result tables one can examine counts of best classifier for any dataset. In the last row, the comparison is made among all of the classifiers in terms of win and lose, for example NP-FiLM give highest score in ten dataset and remain in draw position in 10 other dataset.

Before discussing the comparative results for tree and function classifiers, it is preferable if we draw results while keeping maximum number of external and internal parameters quite same. This includes the same searching algorithm, the same number of potential candidate for parental node, estimation of frequencies, the pre processing steps such as deciding what to do with missing samples and discretization of continuous data. We kept all of these parameters same and plug seven scoring function one by one including our proposed NP-FiLM. In BBN, number of potential parents is a non trivial parameter. Its value greatly influence the shape of the final structure. A higher value is responsible to yield a dense network as compared to keeping a small value. A dense network also pose to increase the size of parameter learning. Moreover, the enumeration for maximum potential parents for a non class node given a certain scoring function which is being exercised in a particular searching algorithm is indeed a bounded value for every dataset. The increase in this value does not imply that the non class nodes will be conditioned with more parents rather it gets exhausted. This study introduced three sets of experiments to validate the effectiveness of our NP-FiLM. In first session of the experiment, the maximum value of parents is set to four which means a dense network as compared to other two sessions in which this value was set to three and two respectively. Other general parameters for classification in weka experimentation were simple estimator and 10 fold cross validation. The setting of markove blanket set to false, initNaiveBayes value set to true and random order set to false.

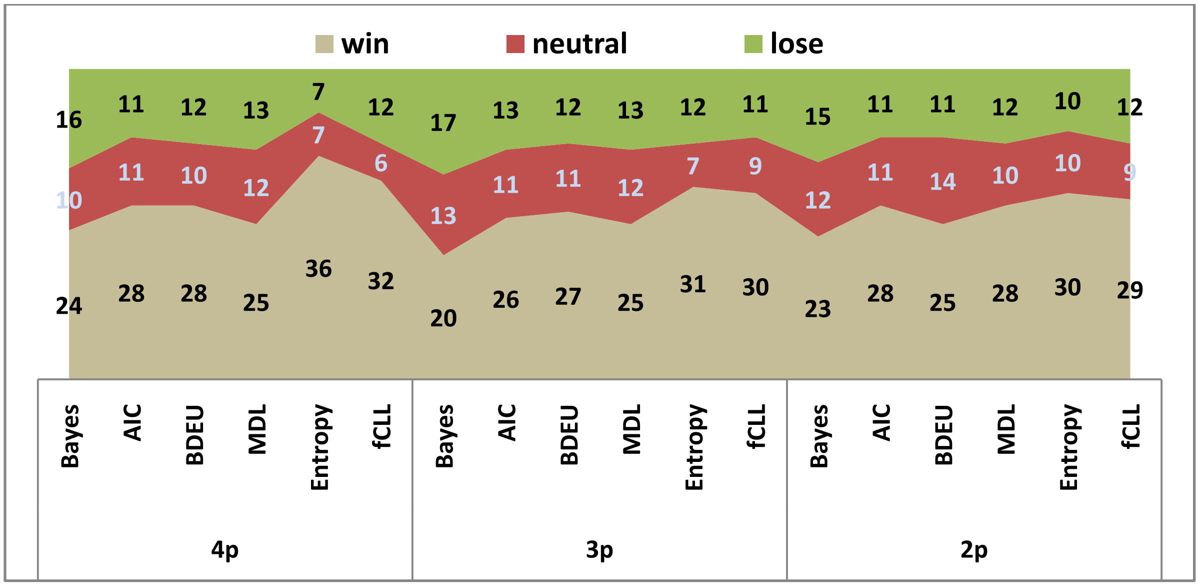


Figure 3.2. win/neutral/lose NP-FiLM BBN vs. peer scoring metrics (50 dataset)

Table 3.2 indicates a significant relative win of NP-FiLM as compared to other scoring function. We have also included a recently introduced scoring function (fCLL) introduced by Carvalho et al., (2011). The authors of fCLL have made available the source code of the program, hence this code was useful in obtaining the result on the dataset in table 3.1. The scoring function fCLL was evolved in the background of improvement in TAN, however, we exploited its functionality in context of general Bayesian network with maximum parent set of four. fCLL achieved highest accuracy in five dataset. When we examine the results from the perspective of average accuracy, again NP-FiLM outperforms the other scoring function.

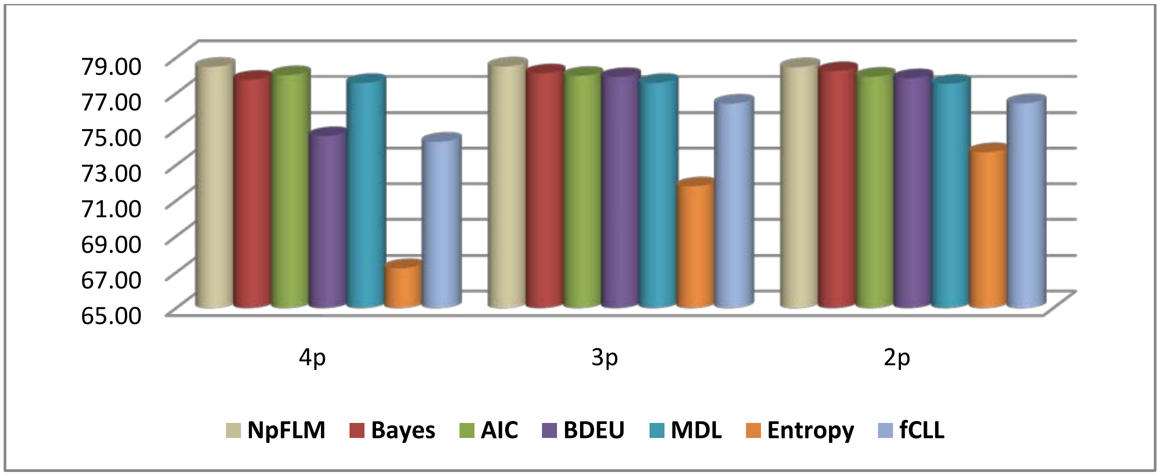


Figure 3.3: Average accuracy: NP-FiLM BBN vs. peer scoring metrics (50 dataset)

This session of experiment was repeated by keeping parent set value of three and two respectively as indicated by the figure 3.2 and figure 3.3. A careful examination of all of the figure 3.2 is indicating that the best performance of NP-FiLM was obtained in the relatively dense graph (see figure 3.3). Secondly an important observation we noticed is that in all of three cases, the proposed scoring function exhibit almost same accuracy. It means the proposed scoring function usually (not always) generate a polytree whatever the value of maximum number of parent is set to above than two. This aspect of the technique reduces its computational cost significantly. In fact, its simple heuristic can enable to select the best non class node as the parent value unless another node with best characteristics is not found, it is not conditioned with that specific node. While drawing the best representative structure of the underlying dataset, there are literally numerous factors which comes into play and dictate whether a best structure can be emanated out of churn of scoring function and searching algorithm. In fact, dataset in general can never fit neatly into general all-encompassing characteristics and their corresponding predictions. The problem of learning the best Bayesian network structure using notable scoring function BDeu, Bayes and AIC was studied by Silander and Myllymäki (2012). They consider the nature of NP-hard problems when it is required to find the best structure among all possible structures. They introduced a distributed algorithm which spans its processing power over multiple processors to find the best structure among all of the available networks using three scoring functions. The algorithm was limited to around thirty variables. The other limitation in the technique is that it was more akin to brute force. There is no doubt that brute force can yield the best result in many scenarios but at the cost of high processing power.

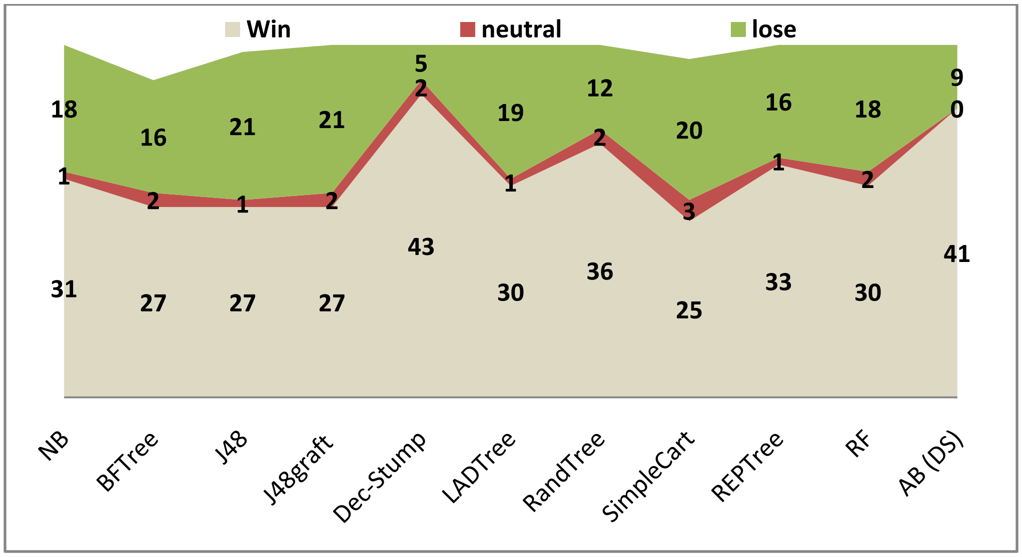


Figure 3.4: win/neutral/lose: NP-FiLM BBN vs. tree classifiers

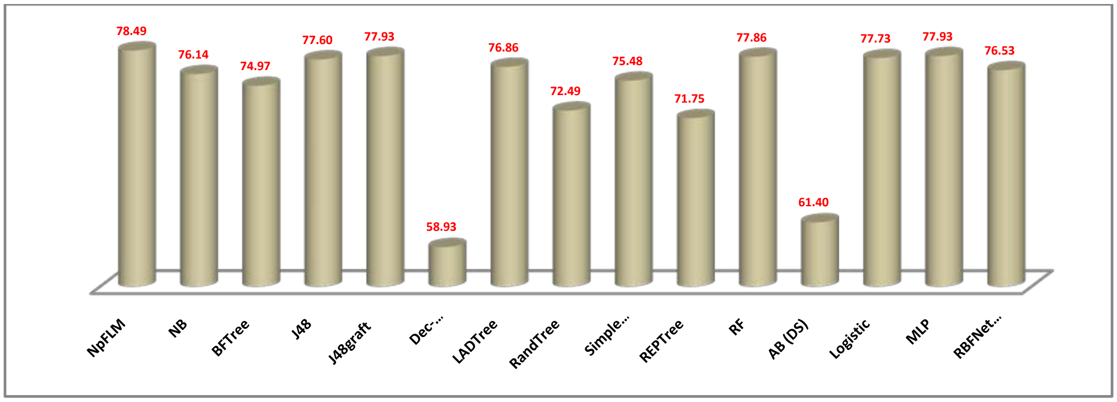


Figure 3.5: Average accuracy: NP-FiLM BBN vs. non BBN classifiers

Figure 3.4 and 3.5 represents the results of tree based classifiers including Simple NB, Breadth First (BF) Tree , J48 which is implementation of C4.5, J48graft, Decision Stump, LAD Tree, Radom Tree, Simple Cart, Random Forest (RF) and Decision Stump with Adaboost (AB(DS)) ensembler.

In comparative result NP-FiLM give best result in 14 out of 50 dataset followed by random forest for which random forest gives best result for 9 result. In some of the dataset, the highest score was shared by more than one classifier such as J48, J48graft and RF where J48 and

J48graft give highest result for dataset ‘trains’ and mushroom. The later dataset were also perceived with highest accuracy by Random Forest classifier. Some dataset were too large in number of features that some of the classifiers did not give result in reasonable time, so we have excluded them from respective cumulative results. On the other hand, when we observe the result from different perspective of win/neutral/lose (see figure 3.4), then it is the comparison of only NP-FiLM with respect to other tree classifier. In these comparison, one can notice that Decision Stump and Adaboost Decision Stump both give poor result in comparison to NP-FiLM while J48 and its modified version J48graft were in close competition; albeit NP-FiLM outperforms all of these tree classifiers. It is noteworthy that while calculating average, missing cells were omitted for comparison on equalitarian basis. It is evident from figure 3.5 that the highest average accuracy was obtained by NP-FiLM which is 78.49% followed by J48graft and RF classifier while the worst classifier in this comparison was Adaboost with a score of 61.4%. A broader comparison of the figure 3.4 and 3.5 and the detailed tabulated result in appendix A indicate that on the overall, the tree classifiers are comparatively well suited for ‘thin network’, whereas the notion thin network points out the degree of size. A small size means less complex network while a big size indicates highly complex network. Exemplifying it, the dataset arrhythmia and audiology contains 280 and 70 features respectively with the class size of 16 and 24 respectively. These datasets can give rise to a complex network. The tree classifiers did not deliver best in both of these cases. in fact the same is true for other datasets where the raw dot product of number of attributes and class size is relatively larger; albeit this product score does not strictly (rather generally) indicate the complexity of the size. (see Proposition 3.1 for detail). The dataset where the performance of tree classifiers is relatively better posses very simple structure (thin network) as in case of balance-scale, hayes-roth_test or some others.

When we discuss the number of parents for a non class node in BBN, three groups can be introduced. The first group contains single parent in which each node is linked to its single parent which is a class node. The second group is Tree Augmented Naïve Bayes (TAN) introduced by Friedman et al. over a decade ago in which each node is linked to a class node and one non class node as its parent. The third group was quite independent of this category, in which any node must be linked to class node but apart from this basic assumption, any node can have other node as its parent where the count of parents is usually restricted by the user of the system. Madden (2009) termed this group as General Bayesian Network (GBN).

Table 3.3: Comparison of NP-FiLM versus published results (Madden, 2009)

Dataset	**Naïve**	**TAN**	**GBN-K2**	**GBN-HC**	**NP-FiLM**
Adult	84.03	86.15	86.16	86.02	85.90
Australian	85.8	85.06	86.22	85.93	85.94
Breast cancer	97.38	96.99	97.32	97.15	97.00
Car	85.15	93.96	89.61	86.36	91.61
Chess	87.85	92.09	94.45	94.95	90.55
Cleve	82.87	81.04	81.07	82.33	83.11
Connect-4	72.11	76.43	79.08	73.88	74.60
Corral	87.05	99.23	99.62	99.38	93.75
DNA-splice	95.26	94.92	95.93	95.81	95.17
Flare	80.12	82.65	82.24	82.56	82.65
German	74.61	72.07	74.2	73.25	74.70
Glass2	81.16	79.37	79	77.29	84.66
Heart	82.74	83.11	82.3	83.04	81.11
Hepatitis	86.38	88	87	86.38	83.87
Letter	74.67	86.28	81.76	75.12	84.54
Lymphography	82.16	81.07	77.46	75.06	87.16
Mofin-3-10	85.34	91.96	86.85	93.04	94.26
Nursery	90.29	93.3	91.18	91.68	91.24
Pima	75.69	76.37	76.33	76.18	78.26
Segment	91.27	95.27	94.64	93.45	95.84
Soybean-large	91.83	92.35	89.22	78.02	93.12
Spect	68.53	70.29	68.98	74.19	68.75
Tic-tac-toe	69.76	76.32	69.26	68.38	75.89
Vehicle	60.62	70.36	67.3	62.5	72.93
Vote	90.27	93.84	93.57	95.11	92.18
Waveform-21	80.9	81.96	81.67	79.73	83.90
Average	**82.46**	**85.40**	**84.32**	**83.34**	**85.49**
Win	**1**	**8**	**5**	**3**	**11**

However one restriction of Markov Blanket was essentially implied, according to which markov blanket is used to ensure that every non class node in the learnt structure must be a part of markov blanket where the markov blanket of any node points out to its parents, children, and other parent of its children within a learnt network structure. Madden (2009) give a comparison between three of these type of network and give its assertions that GBN is relatively a better network structure and is inherently robust enough to be adapted into any specific domain set. This is the reason that a lot of variant of GBN have been proposed preferably suitable in various domain of interest while the other two networks were quite void of this phenomenon. Albeit It was pointed out that GBN may suffer from some limitations, yet this breed of classifiers deserve more attention due to its versatility in nature and insight into classification decisions yielding good accuracy. Some related work was also addressed by Liu, Zhu and Yang, (2013) in which the issue of attribute independence was considered. Liu, Zhu and Yang (2013) argued that independence assumption incur the difficulty of expressing the attribute mutual dependence resulting into poor classification accuracy. Keeping in view of this argument, they introduced a Bayesian classifier based on optimization model (BC-OM). The model introduced a measure of coefficient between two attributes which was based on chi-squared statistic. This measure evolves in form of an objective function in terms of an overall measure of the dependence for a whole of the underlying structure. A higher value of this objective function was an indication of optimal structure. The objective model illustrates that there are some conditions when modeling correlation among variable tend to increase the classification accuracy. The authors also show that BC-OM can be a reasonable tradeoff between the computational complexity in the structure learning phase and the quality of the approximation of correlations among attributes.

Madden (2009) challenged existing paradigm according to which Tree Augmented Naïve Bayes (TAN) is superior in its classification accuracy over General Bayesian Network (GBN). Madden (2009) produce a comparative study of four NB classifiers. Simple Naïve Bayes as show in the column next to dataset in table 3.3. simple Naïve Bayes indicates all of the features have at most single parent which is a class node. Optimal TAN is build by marking the maximum weighted spanning tree within a complete graph connecting the nodes, while nodes are annotated by the conditional mutual information between all pairs of non class variables but conditioned on the class node, as shown by the equation 3.19.

$$I(x_1, x_2 \mid c) = \sum_{x_1, x_2, c} \left[P(X_1, X_2, C) \log \frac{P(X_1, X_2 \mid C)}{P(X_1 \mid C) P(X_2 \mid C)} \right] \text{...............(3.19)}$$

Madden (2009) presented comparison of these two classifiers and two flavors of GBN. The first was termed as GBN-K2 in which BDeu scoring function was used within K2 search algorithm.

The second GBN was GBN-HC, in which MDL scoring function was used with hill-climbing search function. They exercised these experiments over 26 datasets from UCI machine learning repository and concluded that the prevalent axiom that TAN usually outperforms is incorrect. The environment used in their experiment motivated us to give a comparison on the published results because the pre processing steps were quite similar to our study. All of the 26 datasets were discretized using the same mechanism which we employed. Moreover, missing attributes were ignored in both of the studies. Table 3.3 illustrates that NP-FiLM deliver outperforms the others in 11 datasets. Moreover, the average of the classification accuracy was also highest towards NP-FiLM. The experimental analysis reveal out that the poor performance which was earlier reported by Friedman, Geiger and Goldszmidt (1997) about GBN has its roots in simple empirical frequencies in order to estimate General Bayes Network parameters. It can be concluded that parameter smoothing plays important role in improvement of a classifier. Madden (2009) pointed out that GBN has much more potential to be considered for any specific domain because of its diverse nature in drawing structure.

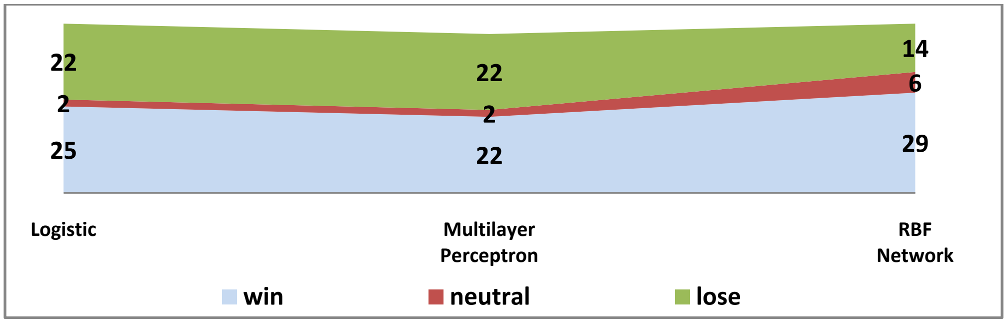


Figure 3.6: win/neutral/lose: NP-FiLM BBN vs. neural network and linear classifiers

Figure 3.6 is another comparison of NP-FiLM towards function classifiers which include Logistic (a regression model), Multilayer Perceptron and Radial Basis Function (RBF) Network. These classifiers in general have high time complexity as compared to their peer classification system. Specially Multilayer perceptron consume exceptionally outstanding time and we have come up with blank value in cells for some larger dataset cylinder-bands, kdd_synthetic_control, mfeat-pixel and splice (see appendix A). It can be observed that although Multilayer Perceptron give some comparable results to NP-FiLM in which NP-FiLM wins over 22 dataset and also lose on other 22 dataset. But the time complexity of NP-FiLM is far lower than dictated by Multilayer perceptron. Moreover, the average accuracy for Multilayer perceptron was also low for 46 dataset which is 77.93% but NP-FiLM gives average of 78.49% (see figure 3.5).

In order to validate the performance accuracy of NP-FiLM, we exercised various sessions of experiments from different angle. Previously we confide ourselves on its comparison on K2 which is a greedy algorithm. Although greedy search algorithm is still popular in BBN (Xu et al., 2013) where they introduced an estimation technique of distribution algorithm L1BOA which is L1-regularized Bayesian optimization algorithm. The technique comprised of two step procedures. In first step, the preliminary parent set for each of the node is assessed. In the second step, each node is evaluated in terms of its corresponding set of possible parents by means of passing it through a greedy search space. The model was shown to be more accurate over simplical models. The comparison result were shown on a synthetic dataset. The distribution of synthetic population is a factor in controlling the quality of learnt structure. This technique is arguable in choice of synthetic population. Moreover, numerous researchers have argued over the performance of K2 greedy algorithm. Lerner and Malka (2011) demonstrated that substituting the K2 search algorithm by Hill Climbing (HCL) search technique has the potential to improves the accuracy in the BN structure. This motivates us to collect the result of these well known scoring function using hill climbing. Again, we set the default setting of BBN using hill climbing search. We made the comparison to our introduced measure NP-FiLM. The results in general validate the findings of Lerner and Malka (2011) that hill climbing is potentially prone to yield better results, however, the accuracy results for NP-FiLM was still better as compared to its peer scoring function. One dataset arrhythmia which is significantly wider in size of features, it contains 280 features and sixteen classes, albeit its sample size is short which is only 452. We noticed that scoring function such as Bayes, AIC, BDeu, MDL and Entropy did not give result even a pass of 48 hours when using hill climbing. This left us with no option but to omit this dataset result at all from the table 3.6 and 3.7.

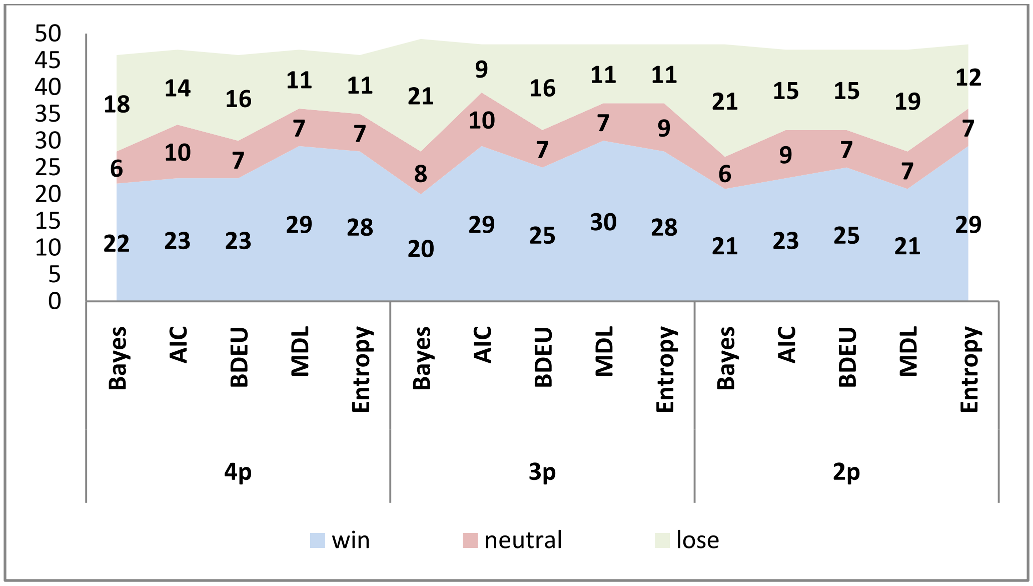


Figure 3.7: win/neutral/lose: NP-FiLM BBN vs. peer scoring function BBN using HCL

During the experiment of comparison with peer scoring function, it was noticed that entropy scoring function always yield a very thick and dense network in which significantly large number of arcs are established. Although adding each arc in general increases the training accuracy but at the same time, it is being done at the cost of reduction in accuracy of test samples. In fact there is a need to select such a scoring function where neither very thick nor very thin network is generated so that overfitting and underfiiting both can be aggregated to their minimum. As far as BDeu is concerned, its performance is mainly characterized by the alpha value which controls the penalty factor. The same is true for Bayes scoring metric. Unluckily there is no such mechanism found which can dictate in advance about the specific value of alpha for any particular data. An optimized value of alpha for each dataset is always a variable and can not be predicted in advance. As far as AIC and MDL are concerned, we term both of them as sister scoring function as both of them differs by the way of controlling penalty factor only. Some techniques were used to improve the performance of MDL (Suzuki, 1999) where the searching algorithm during the construction of network was revised. they replace the greedy algorithm K2 (Cooper and Herskovits, 1992) by branch and bound algorithm and MDL as scoring metric. The superiority of MDL over its peer technique BIC was also elaborated by Zgurovskii, Bidyuk and Terent'ev (2008). They analyzed two important scoring function MDL and Bayesian Information Criterion (BIC). They concluded that MDL is superior in time complexity as compared to BIC as the number of nodes gradually increases. Secondly it is a good practice to use prior knowledge or expert knowledge for developing a per order set of the nodes before subjecting the variables nodes in K2 search. Another finding was about BIC's behavior of over learning; simple to say, BIC can lead to overfitting in numerous cases.

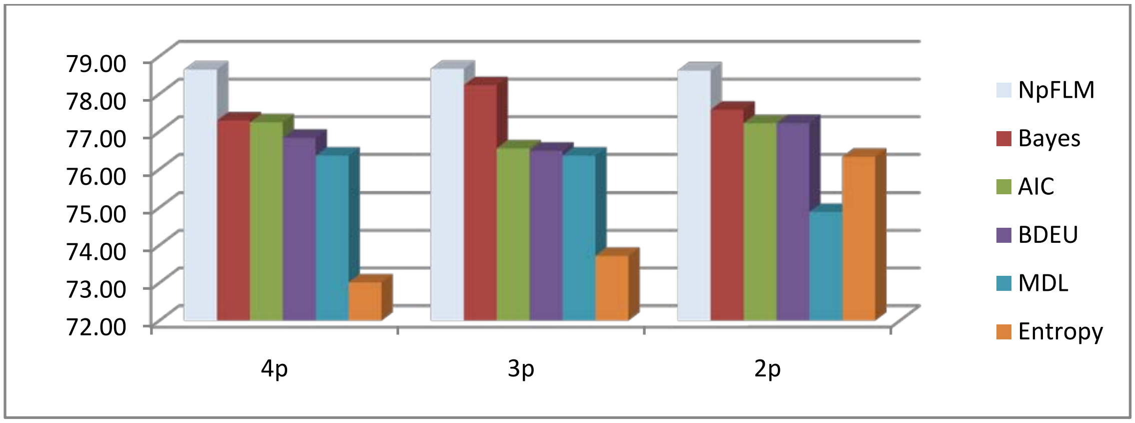


Figure 3.8: Average Accuracy: NP-FiLM BBN vs. peer scoring function BBN using HCL

A general analysis of tables 3.2 and figure 3.2 to 3.8 reveals that the accuracy of the dataset using NP-FiLM is constant although a very important influential parameter 'parent count' is altered from 2 to 4. In fact, this measure keeps a balance between the scarcity and the high density of the network with a small alteration only. Rajaram et al., (2011) stated that ''small alterations" to Naïve Bayes are in an abundance in literature in pursuit of correct adjustments for specific nature of dataset. A true balance prevents it from growing very dense or very thin network. This clearly avoids the test network to suffer from over fitting or under fitting phenomenon. However, in its peer technologies, the density or scarcity is not well controlled. In case of BDeu and Bayes, the value of alpha greatly determine the density level. Entropy measure usually always prone to give dense network. Whereas MDL and AIC are giving relatively better performance, however we have shown that in all of the cases using K2 and Hill climbing with variation in the size of potential parent set, the proposed measure outperformed.

Table 3.4: Comparison with Kabir et al., (2011)

Dataset	ECNBDMMI	ECNBDMMII	NP-FiLM
Thyroid	95.59	96.0035	99.0721
Iris	98.53	100	92.6667
Adult	87.38	89.97	85.9034
Car	89.9	90.65	91.6088

In the last, another comparisons of NP-FiLM to technique forwarded by Kabir et al., (2011) is described as shown in table 3.4. Kabir et al. (2011) presented two models ECNBDMM-I and ECNBDMM-II for improving accuracy of the naïve Bayes classification system. The underlying idea behind these models was to split the training data into clusters where clustering was performed on a simple K mean cluster. Each cluster was considered to learn the model and then test data is evaluated. The authors illustrate that clustering can produce a better training set eventually an improved model learning. Moreover, in these models, the number of clusters is again a question; albeit authors produce a criteria of weighted training error such that.

$$training\_error = \sum_{i=1}^{k}\left(cluster\_error\_of\,(Ci) \times \frac{n_i}{N} \right) \text{..............(3.18)}$$

Where C = {C_1, C_2, C_3,…… C_k} comprise of set of k number of classes.

n_i=Number of data of ith cluster and N denotes the count of all training instances. The authors set the initial value of k to 2 and then increase it gradually till it reaches a specific stop threshold. The stop threshold is marked by continuous increase of weighted training error after few observation. This generate an optimal value of K. Our experimental comparison the published result made by Kabir et al. (2011) shows that ECNBDMM-II are somewhat efficient with good accuracy on their best dataset; albeit the dataset is quite narrow to generalize their techniques. (See table 3.1). One dataset Iris is particularly a short data and the number of states in each of its features are below medium in size. It suggests that their technique might be well suited for thin networks. However, there are many issues arguable in models ECNBDMM-I and ECNBDMM-II. Firstly this technique in each run, training data set is limited enough to build a "correctly represented" model. While clustering whole of actual dataset there are n number of clusters; only a single cluster is used for training model whereas the test data is assumed to be fixed. It means n number of models are developed considering each cluster for its training and each model is evaluated on same "fixed" test data. Such model can be termed as the building block of incomplete data arguing a question of data biases.

3.5 Summary

In this chapter, we have provided some quite relevant techniques in scoring based searching algorithm. A large number of benchmark dataset were used, which clearly give a conclusion that the proposed metric in this study is quite well for thin as well as thick network by keeping a balance between underfitting and overfitting. In the next chapter, we shall discuss one particular aspect of K2 which is initial ordering with some recommendations and proposal of a ranker evaluator.

Chapter 4

POLARIZATION MEASURE: FEATURE RANKER

In many classification problems specially BBN classification system, identification of influential and characterizing features (also known as attributes or variables) within the observed data set plays an essential role in minimizing the classification error. We can term this problem as curse of dimensionality in general. Some pragmatic solutions to the curse of dimensionality can be trifurcated into three dimensions.

4.1 Feature Reduction

The first dimension is feature reduction. In feature reduction, new set of features are emanated from the existing set of features; in fact the prior features are meant to lose their identity at the cost of new features. These techniques cater for capturing maximum volume of information into a reduced number of newly born features. Latent Semantic Analysis and Principal Component Analysis are well known data reduction techniques.

4.2 Feature Subset Selection

The second dimension is feature subset selection. In feature selection, only a sub set of the actual features is considered with the aim of rejecting the redundant and/or irrelevant variables. Note down that the irrelevant or redundancy is measured with respect to class variable. This category further comprises of three standard approaches; embedded approach, filter approach and wrapper approach. Although originally Kohavi and John (1997) introduced the binary category of filter and wrapper approaches; however, researchers argued that this category can be extended to third type known as embedded approach. The embedded approach is coined by the inherent nature of the underlying classification algorithm. The classification algorithm itself brings out the operation of feature selection under its criteria of supervised or unsupervised learning. OneR Attribute Evaluation is a notable example of such embedded approach where the logic of classification technique itself decides the selection of attribute at any specific level. In filter approach, features are selected a prior to the application of classification technique. Filter approach has nothing to do with the target classification technique in use. The filter approach rests on well defined statistically established principles such as pair-wise correlation, standard deviation etc. Majority of the feature subset selection evaluators belong to this category. In table 4.1, except wrapper subset evaluator, all of the techniques belong to this category. The wrapper approach is punched with the target classification technique which acts like a black box. Hall and Holmes (2003) introduced another taxonomy marked by evaluation of individual or subset of features. Table 4.1 presents some of the available feature subset selection technique under this category. These evaluators are available in weka implementation (Hall et al., 2009). This table will be helpful in the result section for analysis and comparison among various approaches.

Table 4.1: Taxonomy of Feature subset selection

Individual Attribute	Subset
Chi Squared Attribute Evaluator	Cfs Subset Evaluator
Filtered Attribute Evaluator	Classifier Subset Evaluator
Gain Ratio Attribute Evaluator	Consistency Subset Evaluator
Info Gain Attribute Evaluator	Cost Sensitive Attribute Evaluator
OneR Attribute Evaluator	Cost Sensitive Subset Evaluator
Relief Attribute Evaluator	Filtered Subset Evaluator
Symmetrical Uncertainty Attribute Evaluator	Wrapper Subset Evaluator
SVM Attribute Evaluator	

4.3 Variable Ordering for BBN Classifier

The third category to tackle the problem caused by the feature dimensionality is variable ordering. We discussed in previous chapters that an important requirement in K2 algorithm is an initial topological ordering. The topological ordering is important as it dictates the set of potential candidates of parent for any node appearing in a specific fashion. In K2 algorithm, the parent variables always precedes its possible children node. This is a question of interest whether this ordering imply any requisite accommodation in pursuit of the optimal structure learning. There are different ways to determine this topological ordering, either based on prior knowledge if exists or otherwise this ordering is set arbitrarily or randomly. Many researchers stated that during the structure learning phase, the position of the class node placing it at the top most result into an improved predictive accuracy (Cheng and Greiner, 1999; Friedman, Geiger and Goldszmidt, 1997; Madden 2003; Singh and Valtorta, 1995). This causes in Bayesian network that the class node becomes parent node for every other node. This is a conventional pre requisite in modern BN classifiers in practice.

An important finding around the impact of initial ordering in K2 algorithm was introduced by Silander and Myllymäki (2012). The authors used the Flags (UCI) dataset which contains 29 query variables and shows that the classification accuracy is a function of initial ordering of variables. In fact the variable ordering mechanism (Silander and Myllymäki; 2012) can eliminate the effect of brute force to some extent as once the ordering of variables is set, the computational space is significantly reduced. On the other hand, Lerner and Malka (2012) argued that the initial ordering, random ordering or ordering made by any intelligent ordering has no impact on classification accuracy. However, they presented their findings on a single dataset Fluorescence in situ hybridization (FISH) where such finding can not be generalized for other datasets. Moreover, They used J_3 scatter criterion for ordering purposes (for detail, see Lerner et al., 2012), although J_3 scatter criteria was introduced more than thirty years ago, more sophisticated mutual relationship measures have also been defined.

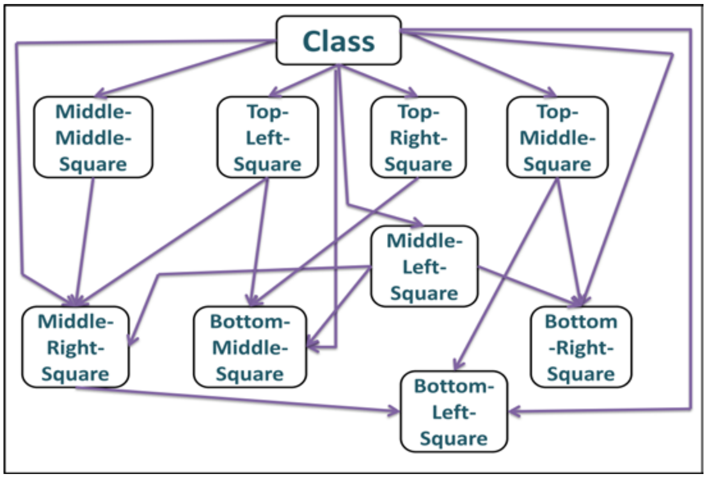


Figure 4.1: BBN (BIC) drawn from Tic-Tac-Toe game

4.4 An Illustrative Paradigm

In order to facilitate the argument in favor of precedence of variable ordering over subset selection, we shall draw a real world example from a famous game Tic-tac-toe or nougats and crosses. This example illustrates why popular belief of feature subset selection is not always suitable in every situation. This dataset contains 958 cases, 9 features and two classes. A BBN drawn by using BIC scoring function with maximum four parent is shown in figure 4.1. This is a type of the dataset where we can not afford improvement in classification at the cost of surrendering any features because all of the features have a direct and confirmed impact on final class variable. We applied CFS evaluator (Hall, 2000) using best first searching method, it eliminates four features from this dataset including top middle square, middle left square, middle right and bottom middle square. Furthermore, a reduction in accuracy of the classifier was also observed. This is a typical scenario where the end user is not interested in surrendering any feature at all; albeit the motivation for an increase in accuracy is still persisting. We in experimental section have shown that we can still improve the accuracy of this dataset using BBN and Random Forest (RF) classifier by means of applying feature ranking technique.

It is useful to give some precise insight into the general methodology of the evaluators which are in discussion in this study. We shall discuss each of them as following:

4.5 Subset Evaluators

Correlation-based Feature Selection (CFS) introduced by Hall (2000) is based on the evaluation of attributes subset; the success of this algorithm initiate a series of introduction of subset evaluators subsequently. The central crux of this technique relies on the idea of introducing such subsets which minimizes the inter-correlation and maximizing the intra-correlation. Note down that inter-correlation relates to the correlation among members of the subset and intra-correlation refers to the correlation to class variable. The rationale behind this technique is that the subset with attributes highly related to each other is prone to be poor predictor of the class. The merit of any potential subset is defined by the equation as below.

$$Merit_s = \frac{k\,\overline{r_{cf}}}{\sqrt{k + k(k-1)\,\overline{r_{ff}}}} \qquad (9)$$

Where r_{cf} is an indication of how predictive a group of features are; and r_{ff} shows the degree of redundancy among features of a subset. The basic measure between any two features is a function of entropy of feature indicated in the name of symmetrical uncertainty as below.

$$SU = 2.0 \text{ X} \left[\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)} \right] \quad (10)$$

Consistency Subset asses the worth of a subset of features by the degree of consistency found in the class values when the training instances are projected onto the features subset. The important point in consideration is that consistency of any subset can never be smaller than that of the complete set of features; that is why this technique can be exercised in conjunction with exhaustive search looking for the smallest subset with consistency approaching the consistency value of the complete set of features. Classifier subset evaluator delivers an estimated merit of ‘set of features’. It works on training data or a separate hold out testing set while employing a particular classifier.

4.6 Feature Rankers

Gain Ratio Attribute Evaluator and Information Gain Attribute Ranking both are simple individual attribute ranking mechanism. In this technique, each attribute is assigned a score. The score is delineated by means of the difference of an attribute’s entropy and its class conditional entropy. The difference between both of these entropies formulates the information gain for each of the attribute. Dumais et al, (1998) and Yang and Pedersen (1997) reported that this uncomplicated technique is much suitable in case of text classification.

Relief Attribute Evaluator which is an individual attribute evaluation technique is more versatile as compared to its peer FSS because it can operate discrete as well as continuous data. Moreover, this technique is quite capable of handling noisy data. Originally it was introduced by Kira and Rendell (1992) for two classes only; however, it was improved for multiclass (Kononenko, 1994). The central idea in this technique is identification of nearest neighbor from same as well as opposite class. Using Relief single attribute, evaluation is carried out by means of comparing training examples with their corresponding neighbors. Relief single attribute evaluator repeatedly samples an instance while examining the value of the given query variable for the same and different classes. This procedure leads to evaluation of the worth of every attribute (Hall et al., 2009). Ranking is performed by means of relevant scores. A valuable attribute is defined by the same values for instances from the same class and different values for instances from different classes. Relief posses certain characteristics in feature selection techniques. Firstly it is quite robust to noise present in the data. Secondly it is significantly efficient feature selection algorithm. Compared to many other peer techniques, it chooses small set of statistically significant features only while rejecting many features as irrelevant or redundant, albeit this small size set is not prevalent in all of the situations but it can be observed in many scenarios. An improved version of Relief was also introduced namely ReliefF, by Kononenko, Simec and

RobnikSikonja (1997). It enhances the capability by extending its use to the top-down induction of decision trees in each of the possible selection step.

Symmetrical Uncertainty Attribute Evaluator is restricted to discrete features only. This technique approximates the association score between discrete variables with respect to the class. Classifier Subset Evaluator and OneR Attribute Evaluator both are member of embedded class of FSS. The underlying logic behind OneR Evaluator is based on OneR classifier (Holte, 1993). Chi Squared Attribute Evaluator is based on well established statistical measure for test of hypothesis where scoring value between each attribute and class is calculated for marking it as suitable or unsuitable feature for classification technique. Filtered Attribute Evaluator and Filtered Subset Evaluator both are filter based techniques. In both of these techniques, the attribute or set of attributes are evaluated by passing them through an arbitrary filter defined on the training dataset.

The general principle for Consistency-Based Subset Evaluation can be describes: the data is divided in such a way that the attributes with strong single majority class are separated from the other attributes (Almuallim and Dietterich, 1991; Liu and Setiono, 1996). This approach lay out the foundation for several other FSS techniques. Kohavi and John (1997) introduced this breed of techniques which never operate independently. They always works keeping in view of the target data mining technique. This usually gives them an added advantage over their peer FSS techniques due to an enhanced interaction between the classifier's inductive bias and the searching mechanism. The estimated accuracy of the classifier is usually calculated by means of cross validation during the working of wrapper technique. The modified forward selection search is used to generate a ranked list of attributes. The only notable bottleneck of such techniques is increased computational cost specifically in case of large volume of attributes.

4.7 Searching Algorithm for Evaluator & Rankers

Any evaluator always works in the array of a specific search algorithm. Some notable search algorithm includes exhaustive search, greedy stepwise and best first searching algorithms. Exhaustive search carries out an exhaustive search through the whole space of features such that initially empty subset is considered and then gradually each and every possible subset is considered. It terminates the process marking the best subset. Best first adopts the greedy hill climbing technique intensified by incorporating backtracking facility where the level of backtracking depth is a function of the number of consecutive non-improving nodes. Best first is more versatile as compare to other techniques and much optimized in its peer techniques. It has the provision to get started from the empty set of attributes moving the search in forward direction, or start with the full set of feature searching in backward direction, or even it has the capability to begin at any point with options of search in either direction by means of addition and deletion of every possible single feature at a given point. Greedy stepwise performs its searching with either nil or all of features or even an arbitrary number of features in a greedy forward or backward search. It terminates the addition/deletion of any remaining attributes which cause less in size of evaluation score. This searching technique has also the capability to generate a ranked list of features by means of passing over the search space from one side to the other and then recording the order of selected features. Drugan and Wiering (2010) introduced a FSS using Minimum Description Length (MDL). However, it is limited to only binary classes.

4.8 Towards Polarization Measure: Novel Feature Ranker

The central issue in supervised data mining techniques is associated with induction of discriminant model identifying a given instance of an object mapping into a specific class. The induction of the classifier requires that each object is to be enumerated by means of an array of variables. With the advent of advanced computational technologies, data with numerous features is quite a norm. In this scenario, a fundamental axiom is built: the degree of usefulness of all of the available variables for inducing the optimized model. Here the data mining community usually comes up with the idea of selecting the best subset; however we have already specified that there might be the situation when not even a single feature can be given up while still keeping the target of optimized induction model. It is useful if we develop mathematical expression by characterizing feature ranking problem in the context of machine learning. We begins with T= D(F,C) as a sampling space or also known as training dataset in which there are g features and h instances; the set of features can be expressed as F = $\{f_1, ..., f_g\}$ and the dataset instances can be represented as D=$\{d_1, ..., d_h\}$. Moreover C = $\{c_1, ..., c_n\}$ refers to the set of tagged labels so here in this case these are classes. For each instance $d_x \in$ D, it can be denoted as a vector of features, i.e., o_x=(v_{x1}, ..., v_{xg}), where v_{xk} is the value of o_x related to the feature f_l. Given a training dataset T= D (F, C), the task of learning algorithms for classification is to induce a hypothesis $h_0 : F_l \rightarrow$ C from T, where Fl is the value domain of $f_l \in$F.

After this brief introduction we shall move towards inscribing polarity measure between a feature and class attribute. Let the distinct state of the feature are expressed as $F_1 = \{1, \ldots. m\}$ while the distinct states of the class attribute are $C = \{1, \ldots. n\}$. The value of h is already defined as the count of instances in the dataset. Now we denote a_{ij} as the joint probability among feature F_l and Class C, then Polarity Measure (PM) can be mathematically denoted as below:

$$PM(f_l, C) = \sum_{i=1}^{m} \left[\left(\max_{i=1}^{m} \arg [a_{ij}]_{j=1}^{n} \right) \Big/ h \right] \qquad (1)$$

The above equation is in fact a mathematical expression to find the polarity measure between any of two attributes; the only difference is that the class is placed at the second position. For the purpose of comparison it is compulsory to calibrate the value of PM between 0 and 1. The equation above gives a value from 0 to 1 in the following particular situations.

$$PM_{\max} \leftarrow 0 \therefore \left[\underset{i=1,j=1}{\overset{m,n}{[}} \overset{\Uparrow}{a_{ij}}] - \underset{i=1,j=1}{\overset{m,n}{[}} \overset{\Downarrow}{a_{ij}}] \right] \rightarrow 0 \quad (2)$$

$$PM_{\max} \leftarrow 1 \therefore \left[\underset{i=1,j=1}{\overset{m,n}{[}} \overset{\Uparrow}{a_{ij}}] - \underset{i=1,j=1}{\overset{m,n}{[}} \overset{\Downarrow}{a_{ij}}] \right] \rightarrow h \quad (3)$$

a_{ij} : Joint probability with feature 1 in ith state and feature 2 in jth state.

$\overset{\Downarrow}{a_{ij}}$: Minimum joint probability among all of the possible space

$\overset{\Uparrow}{a_{ij}}$: Maximum joint probability among all of the possible space

In the next two step, we shall calculate the difference of the PM value such that the first value is PM (f_i, C_l) from feature to class where as the other value of PM(C_l, f_i) is obtained after swapping the position of the class node and the feature node. The net value is also divided by the later so as to find the net discirminant effect. All of these discriminant values are sorted in ascending order. This will give us a list of feature which is ranked. We demonstrated that the polarization measure can explain the state of the class by means of expressing joint probability in a specific manner.

4.9 Asymptotic Analysis

The equation for the PM given above indicates that it needs a single scan of all of the database transactions. It means the time complexity for this measure is O(n). However inside three iterations, we need to update the information into hash table such as:

```
$for\ h = 1 ......sizeof(D)$
    $for\ i = 1........m$
        $for\ j = 1.........n$
            $hashtable\ a_{ij} \leftarrow D(v_i, c_j)$
        $end\ for$
    $end\ for$
$end\ for$
```

We know that Hash table time complexity is measured in terms of amortized analysis (Cormen, 2009). The amortized analysis is different from average-case performance analysis because of using probability in its analysis (Cormen, 2009). The best time complexity of open hashing in which a single array element can store any number of elements is O(1). Whereas the worst time complexity is O(h) although we can improve it to O(Log(h)) by using a balanced binary tree for each bucket. Hence, we can conclude that the total time complexity for the PM = O(h) x O(log(h)) = O(hxLog(h)) in worst case while it is ϴ(h) X ϴ(1)= ϴ(h) in best case.

4.10 Empirical Validation

A number of benchmark datasets have been used for the evaluation in this part of study. These include dataset with binary classification problems as well as multivariate classification problems obtained from the UCI data repository (Frank and Asuncion, 2010). Table 4.2 is indicating an overview of these dataset in which attributes count, number of rows (cases) and classes are shown. It is preferred if we select dataset with variety of information under these categories to avoid any bias results in favor of a specific technique. To measure the ability of various feature ranking techniques along with their respective searching algorithm the preferable choice of accuracy was adopted as a simple standard classification measure in this experiment. The setup of experiment related to classification includes selection of searching algorithm. The search algorithm during the structure learning is fixed to K2 wherein the parameters for K2 were default parameters in which initialize as Naïve Bayes is an important parameter. If it is set to false then it implies an empty network is to be used as initial network structure but if it is set to true then it means the initial network being used for structure learning will be a Naive Bayes Network, in which there will be an arrow from the node to each other node.

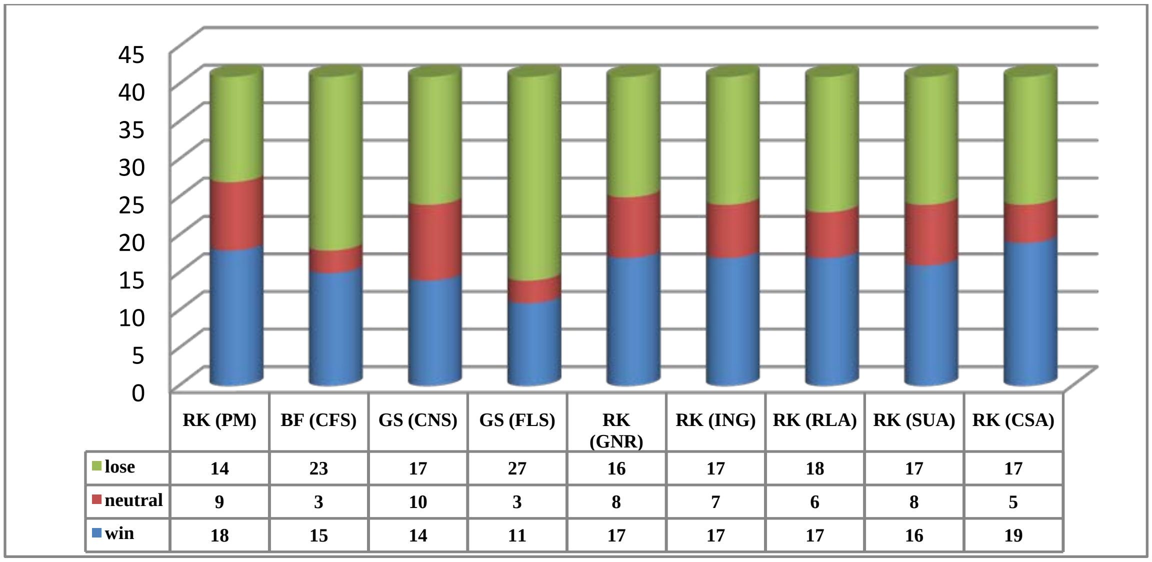
Table 4.2: Dataset used in comparison of various feature evaluators

Dataset	Attrib	Cases	Classes	Dataset	Attrib	Cases	Classes
Anneal	39	898	5	Letter	17	20000	26
Audiology	70	226	24	lymphography	19	148	8
Australian	15	690	2	mammographic	6	830	2
balance-scale	5	625	3	mofn	11	1324	2
Bupa	7	345	2	monk-2	7	432	2
Car	7	1728	4	mutagenesis-	12	1618	2
#chess	37	3196	2	nursery	9	12960	5
Colic	23	368	2	pima	9	768	2
contact_lenses	5	24	3	#primary-tumor	18	339	21
#crx	16	690	2	satimage	37	6435	6
Diabetes	9	768	2	segment	20	2310	7
#eastWest	26	213	2	shuttle	10	5800	6
Flare	13	1066	3	sick	30	3772	2
Glass	10	214	6	soybean-large	36	266	15
glass2	10	163	2	tae	6	151	3
Haberman	4	306	2	tic-tac-toe	10	958	2
hayes-roth	5	160	3	titanic	4	2201	2
#hepatitis	20	155	2	vehicle	19	846	4
kr-vs-kp	37	3196	2	vote	17	435	2
#labor	17	57	2	waveform	22	5000	3
led7digit	8	500	10	zoo	17	101	7

We set the default value which is true. The second setting is related to implementation of markove blanket classifier. This setting indicates that when complete structure learning is achieved then it is tested whether every node is a part of markove blanket for the nominated class and if any node is out of this setting then a correction is made. The default setting was set to false in this case. Moreover, if it is set to true then a true effect of the feature ranking cannot be observed; this left with the option of proceeding with default setting of 'false' in the underlying experiment. The other setting is maximum number of parents. In fact this setting is directly related to the computational efficiency of the structure learning in BBN. It is already an established fact that increasing this value may lead to exponential rise in time complexity of the algorithm. We chose a value such that we can get result for all the dataset with small or large number of attributes; the value we chose in our experiment is four. Another setting in K2 was related to random ordering of the input feature which was also set to false. The status of random Order is nontrivial because in all of the evaluators, the ranking of attribute was about to be tested and if this value is set to a function of randomness then effect of the feature ranking and feature selection algorithm cannot be judged correctly. In fact, random ordering of the features and markove Blanket Classifier values has no effect if number of parents is restricted to only one. The default scoring function used in BBN is Bayes Information Criterion and the same was considered in the experiment.

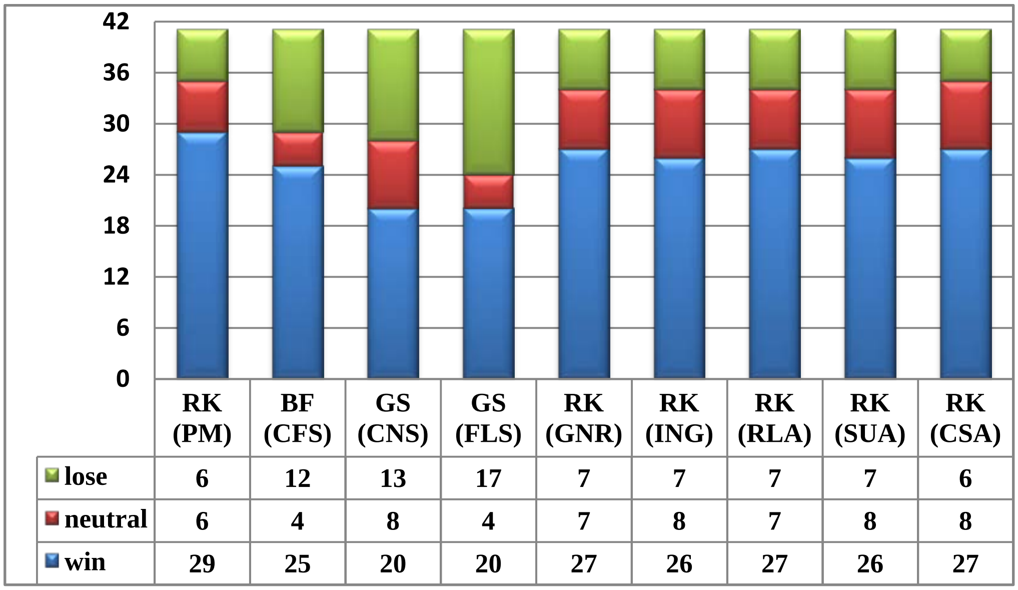
The theoretical detail, significance and evolution of Bayes Information Criterions (BIC) as a scoring function has already been enumerated in previous chapters. While keeping in view the same trend of default setting, we also disable use AD Tree option and restrict the experiment to simple Estimator with alpha value of 0.5 which is a default value for simple Estimator of parameter learning. Other fixed parameters for BBN classification include ten fold cross validation. Some standard setting used in this experiment related to Random Forest include the maximum depth of the trees set to unlimited, the number of features to be used in random selection was set to use all features as we are interested in ranking of features, the number of trees to be generated was set to 10 which is a default setting. The random number seed to be used is set to its default value of 1. The rationale behind using these default values is that the technique suggests these optimized parameters for best results so there was no argue in altering these default values.

The design and comparison strategy used in the experiment is such that: we chose five feature ranking algorithm (evaluators). These include Gain Ratio evaluator (GNR), Info Gain evaluator (IGN), Relief Attribute Evaluator (RLA), Symmetrical Uncertainty Attribute evaluator (SUA) and Chi Squared Attribute evaluator (CSA). All of these evaluators used ranker as their searching heuristics. Apart from these feature ranking evaluators, we also used three well known feature subset selection techniques with their optimized searching algorithm. These include Correlation based Features Selection (CFS) evaluators using Best First (BF) searching algorithm (Hall, 2000), Consistency Subset (CNS) evaluator and Filtered Subset evaluator (FLS) both using Greedy Stepwise (GS) searching algorithm.



	RK (PM)	BF (CFS)	GS (CNS)	GS (FLS)	RK (GNR)	RK (ING)	RK (RLA)	RK (SUA)	RK (CSA)
🟩 lose	14	23	17	27	16	17	18	17	17
🟥 neutral	9	3	10	3	8	7	6	8	5
🟦 win	18	15	14	11	17	17	17	16	19

Fig. 4.2: Comparison of PM to peer techniques using RF



	RK (PM)	BF (CFS)	GS (CNS)	GS (FLS)	RK (GNR)	RK (ING)	RK (RLA)	RK (SUA)	RK (CSA)
🟩 lose	6	12	13	17	7	7	7	7	6
🟥 neutral	6	4	8	4	7	8	7	8	8
🟦 win	29	25	20	20	27	26	27	26	27

Fig. 4.3: Comparison of PM to peer techniques using BBN

Empirically, it is seen from figure 4.2 and 4.3 that PM feature ranking strategy improves RF classification performance in eighteen dataset while it does not affect the performance in 9 cases albeit it gives poor results in fourteen datasets. This result is comparatively better in comparison to its peer techniques. Moreover, it also indicates that there are situations when choosing a part of whole space of features may lead to poor performance of the classifier as such in the case of CFS using its optimized BF searching algorithm. More or less, we can also argue in the same way about Classifier subset which acts in the capacity of wrapper techniques in which it tests the intended classifier during the selection of most influential features. In particular our proposed technique produced significantly better performance on BBN classifiers to majority of its peer techniques and better but not significant improvement performance on RF classifiers for six out of eight feature selection techniques.

The figure 4.3 is indication of amelioration or deterioration of BBN classification accuracy. The figure 4.3 also supports our proposed technique in general where in 28 dataset the improvement was observed while in 6 dataset, the accuracy was reduced and in 6 dataset the accuracy was neither improved nor reduced. These dataset have been marked by # sign in table 4.2. A careful examination of table 4.2 reveals that the empirical reason for reduction in accuracy for PM in 6 dataset seems to be the small size of dataset albeit we cannot generalize it. The figure 4.2 and 4.3 also points out that BBN classifiers is more responsive to change in feature ordering scheme whereas RF classifier has not shown as significant response as compared to BBN.

Chapter 5

A BBN STRUCTURE LEARNING MODEL USING PM & NP-FILM

In the previous chapters, we discussed our core contribution towards structure learning. In this chapter, we shall present a framework demonstrating how these measures can be plugged into the structure learning framework. We have exercised two applications as an example, which include settlement in labor negotiations and knowledge discovery in HCV sequences. The framework consists of three broader sections, an overview is given below.

5.1 Step 1: Data Collection

In this step, we collect raw information from the original source. These information are to be arranged in a way that cause-effect relationship can be realized. For the labor negotiation process, the original data is retrieved from interviewing various stakeholders of the organization usually an industry manager is mostly well versed with these information. The process of collecting raw information differs from on scenario to another. The DNA sequences of Hepatitis C Virus (HCV) consists of entirely different steps with no iteration involved at all. In the first step, which is sample preparation, double helix of DNA ladder is broken shorter fragments. The size of these shorter fragments is around four hundred to six hundred bases pairs. The double stranded DNA fragments are processed to retrieve single strands. In the next stage, the fragments of DNA are put onto very small beads which are micron sized. This stage result into the identification of the signals produced in the previous stage. This step is time consuming and may take approximately eight hours. In the third stage, single stranded fragments are copied by means of an externally supplied enzyme. The out of this stage is double stranded fragment. In the last stage, the genome are analyzed by commercially and non commercially available software. In fact these four stages are quite generalized, many variants of sequencing are available in the research. We in this step just want to discuss that collection of the raw information is usually a multifaceted and exigent task because of its subjectivity, non-linearity in underlying information, and dynamic nature of raw data.

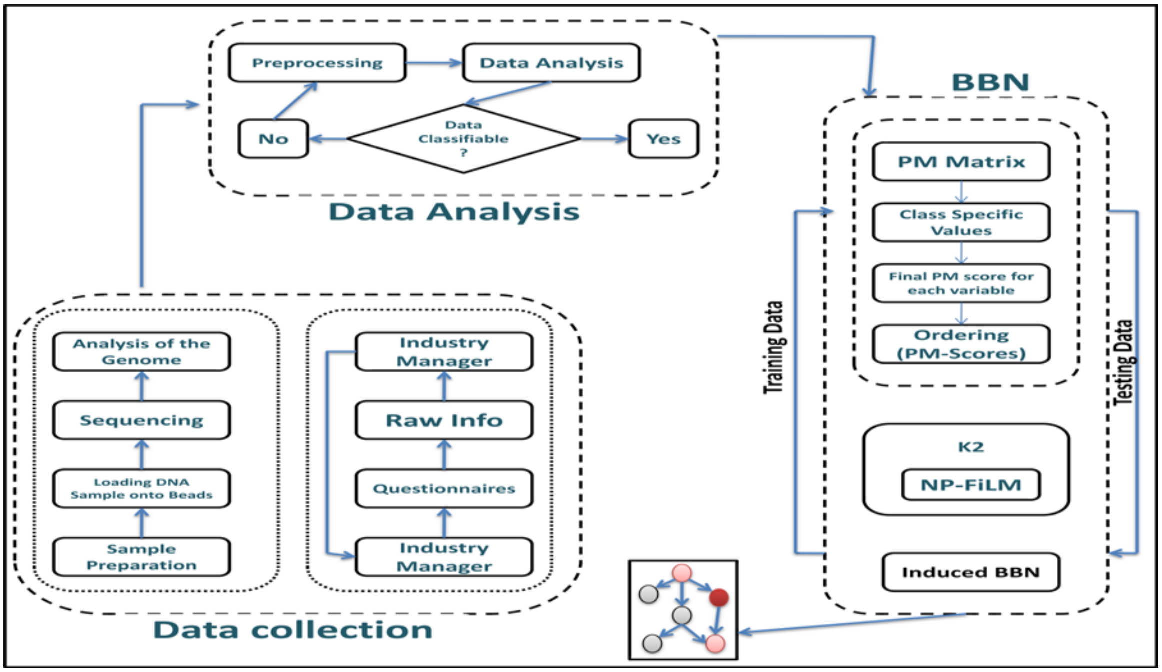


Figure 5.1: Framework for Structure Learning using PM and NP-FiLM

5.2 Step 2: Data Analysis

The modules in this step are responsible for designing the matrix of information so that the information can be tabulated. In fact this step is quite challenging because the performance of the further components are directly influenced by this step. This step deals in identification followed by removal of noisy data. The handling of outliers is also another issue. The outlier greatly influence the machine learning technique but on the same, outlier is a legit data. Another issue is under sampling in which we are to examine those classes which are very low in probability distribution. The outcome of this step is a dataset which is classifiable.

5.3 Step 3: BBN Learning

The third component is compose of sub component of structure learning, in the first stage, the feature variable are re order such that they can be helpful in maximizing discriminant function of classes. in fact, NP-FiLM is enhanced version of PM where in probability distribution is achieved among more than two variables whereas in PM, only two variables are used as input. In general the probability distribution is estimated from training data. There are two approaches to this estimation; probability density and probability distribution using simple histogram estimation. The later enumerate the occurrences of state in a dataset, normalize this count and return it as the probability distribution. It was shown by Fayyad and Irani (1993) that this approach delivers the correct distribution within the limit of infinite data, hence one can express confidence for its reasonable approximation out of finite amount of dataset.

Keeping in view of this general framework, we shall elaborate both of scenarios and will show how structure learning can be performed in using these introduced measures.

5.4 Settlements in Labor Negotiations

The objective of this chapter is to look into tweaking the proposed ranking measure Polarization Measure (PM) and Non Parametric Factorized Likelihood Metric (NP-FiLM) in the data mining task of labor negotiations. We argue for the overfitting problem caused by peer scoring metrics and also highlights the importance of feature ranking instead of surrendering features. In the model, we have shown publicly available data sets of labor negotiation, that the BBN using NP-FiLM with help of feature ranker PM is clearly able to learn more accurate and comprehensible structure. furthermore, we discuss how the learnt model performance could be measured to capture accuracy and comprehensibility as well. Comprehensibility is usually translated into size of the model. Which means a smaller model is deemed more comprehensible.

The dataset used in this model is settlements in labor negotiations which was exercised in Canadian industry. In this dataset, there are not very large number of cases, that is only 57 cases with 16 attributes. The detail of these attributes is shown in the table 5.1. the nature of these variables are versatile including boolean, discrete and numeric dataset. There are no missing values. The class variable is a boolean attribute. The settlements in labor negotiations are to be classified as good worker or bad worker based on the content of the other settlement attributes.

Class is independent and influenced only by the prior information. Table 5.1 is indicating the ordering of the query variable which are determined by scoring metric Polarization Measure (PM). Suppose we want to develop a labor negotiation assistant to diagnose; what are the reason marking any worker a good or bad labor. Moreover, in General Bayesian Network (GBN), it is also more important to validate the causation relationship and conflicting relationship among the variables leading to term any worker a good or bad worker. The system is responsible to report that a worker must be classified a good worker if the required variables denote it favorably (true positive and true negative). However on the other hand, there might be false positive or false negative instances. The vacation is conditionally independent of other variables given class and contribution-to-dental-plan variable. Contribution-to-dental-plan is dependent on contribution-to-health-plan in addition to class variable and it is conditionally independent of all other variable given these two parent variable. These two links are appropriate as reflecting the real world situation where it is a known fact that contribution to health plan is a necessity while dental treatment in general are considered under cosmetic surgery except a few cases of dental issues. That is why, every worker will eventually first prioritize towards contribution to health plan and if s/he can afford then s/he will lead to second priority. The importance of variable vacation can also be explained in the same way. Wage-increase-first-year is causing wage-increse-second-year which eventually giving causation effect for pension variable. Contribution-to-health-plan is dependent on class and pension variables because it receives the causation from pension. Those worker who wish to contribute to their pension fund indicates their long commitment towards the company. This can otherwise cause them think for contribution-to-health-plan.

The variable long term disability term assistance is getting its causation from standby pay. It was observed that bad settlements is attached by a low value for the wage increase in first year of contract attribute. Good settlements are influenced by a high values for first-year wage increase and stand-by pay. Moreover, the conditional distribution of the variables during parameter learning indicate positive values for contribution towards pension and long-term-disability-assistance. When we observe other variables, the same cause-effect is explainable which is more close to the elicitation made by the domain experts.

Some previous work on labor negotiation dataset includes Nayak and Cook (2001) in which essential pattern from small dataset were extracted by using an improved version of association rule mining known as ~AR. The best rules introduced by Nayak and Cook (2001) were interesting and elaborative. We matched these association rules by decomposing them into the conditional relationship brought forward by BBN. The first two best rule founded shows that shift-differential (antecedent), wage-increase-first-year (antecedent) and wage-increase-second-year (consequent) all are associated to each other in the association rule of highest degree of interestingness. If we examine the figure 5.2, the same is true as all of these three variables are attached to each other in the same fashion with a strong correlation under assumption of conditional independence to other variables given class variable. The statutory-holidays was shown with strong association towards class. It corroborates the link between these two variables as illustrated by the figure 5.2. The working hours and contract duration clause are fixed simultaneously in the process of negotiating a labor contract, a model that takes such endogenetic features into account is appropriate as shown in the figure 5.2 where working hours are receiving the dependence from duration. In addition, the proposed model was found that it posses the capability to revise the conditional probability in parameter learning as a result of learning more accurate structure.

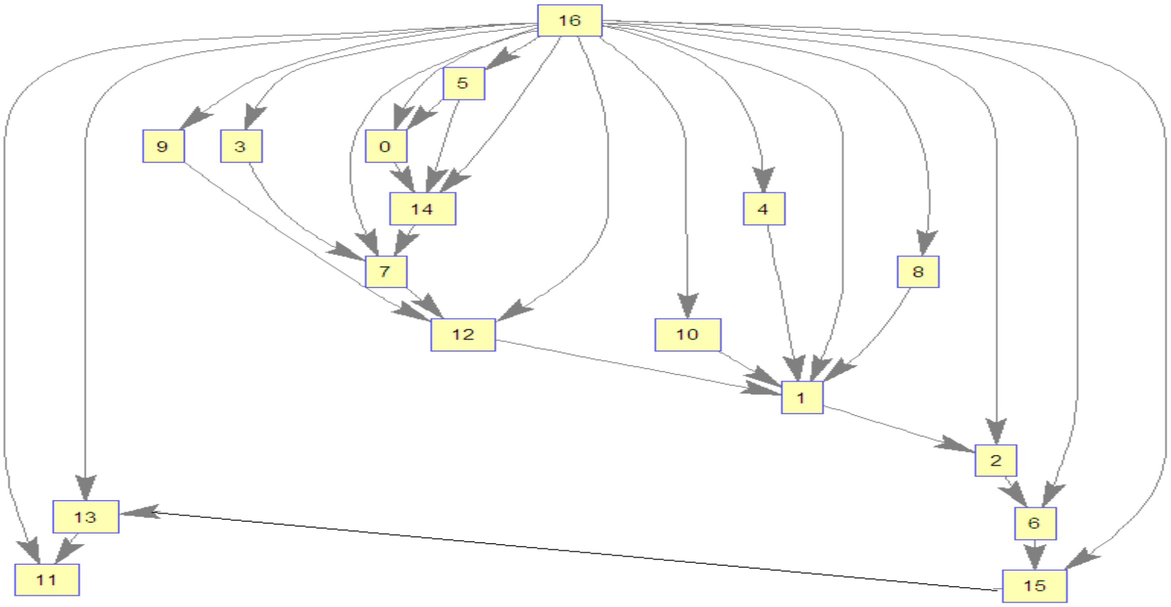


Figure 5.2: Learnt structure of settlements in labor negotiations by PM and NP-FiLM (see column 1 of table 5.1 for node detail)

Regarding the validation of ranker measure PM, we first applied cfsSubset evaluator (Hall, 2000) which is correlation based feature selection evaluator for discrete and numeric class machine learning technique. This technique is optimized with bestfirst searching method. The technique pruned "unnecessary" features. These include working-hours, duration, bereavement-assistance, wage-increase-third-year, standby-pay, education-allowance, shift-differential, pension, contribution-to-health-plan. It is generally accepted that the subsistence of long term contract duration of labor has potentially significant implications towards the macro economy expansion, contraction and behavior. Particularly, duration of contract is specially a non trivial subject relating to the effectiveness of stabilization policies and the dynamic behavior of aggregated variations and fluctuations in the labor market. This was already justified by Fischer (1997) according to which labor contracts during the period of contract can furnish the monetary authorities with a vantage in mitigating the economic shocks in labor market by means of stabilization output. A BBN drawn on these features increases the classification error such that only 82.4% instances were classified. It clearly points out that striking out some essential features is neither appropriate for the model in discussion nor it contributes towards the improvement in classification accuracy. We also applied other feature sub selection techniques on this model but come up with the conclusion that the class and its features are oriented in such a way that every feature is important in a better learning procedures. However, the default ordering of the features produce accuracy of maximum 92.98% (see table 3.2). The ranker evaluator has potential to improve the accuracy if the features are ordered in a position which can maximize the posterior probability of the model. We applied the chi square ranker available in weka (Hall et al., 2009) but it also gives an accuracy of 91.22% which is again not appreciable. The metric PM was applied to this model and it gives the ranking as given in the table 5.1, this ranking followed by application of BBN with NP-FiLM scoring function give better result of accuracy 94.74%. this indicates that our proposed model yields an ostensibly more efficient and generalized least misclassification estimation in the labor negotiations.

Table 5.1: Attributes reordered by Polarization Measure (PM)

Default Ranking	Ranking by PM	Attributes	Default Ranking	Ranking by PM	Attributes
16	0	class	12	7	longterm-disability-assistance
5	1	working-hours	1	11	wage-increase-first-year
0	2	duration	2	12	wage-increase-second-year
14	3	bereavement-assistance	6	12	pension
3	4	wage-increase-third-year	15	13	contribution-to-health-plan
7	5	standby-pay	13	14	contribution-to-dental-plan
9	6	education-allowance	11	15	vacation

5.5 Knowledge Discovery in HCV Sequences

Identification of patterns in nucleotide sequence of HCV is useful in devising a strategy to foster fatal disease. Numerous mathematical models have been presented in literature. For the purpose of most coherent nucleotide sequence in Hepatitis C Virus (HCV), we have developed a model to gene expression data mining which employs a structure learning techniques to figure out the identification of a given genotype within data. The proposed methodology involves data pre-processing, followed by structure learning with outcome of pattern classification. We have evaluated our technique using data from Hepatitis C Virus nucleotide sequence of SAARC countries including Bangladesh, India, Sri Lanka, Nepal, Pakistan and Myanmar. The dataset of other SAARC members was not yet available (HCV Sequence Database, 2005). The proposed modeling approach delivers potential for diagnostic as well as virology applications. Methodology discussed in this study is an approach to converge huge knowledge related to DNA into a very small number of useful information. The described methodology is an approach for knowledge diversification to integration to deliver an insight for analysis in research networks related to virology and medicine. It is required that the techniques need to be designed with certain assumptions to favor some kind of biases in perspective of underlying data (Xu and Wunsch, 2005).

We know that nitrogen base sequences are represented by letters of four nucleotide bases {A, C, G, T}. The research on the phylogenetic taxonomy of HCV nucleotide sequences has pointed out that there are six HCV genotypes numbered 1–6 (Robertson et al., 1998). Every genotype is classified into many subtypes. It was shown that every genotype vary in its geographical distribution as well as in its mode of transmission (Robertson et al., 1998). Among all of these genotypes, the first and second type has been observed with the broadest distribution in the USA, Far East, Europe and partially in African territories. Recently it was reported that Genotype 1b is widespread in Rondônia State of Brazil (Aman et al., 2012). Genotypes 3 and 4 both are mostly rich in considerable number of subtypes as reported by Robertson et al., (1998). Genotype 3 was also found in a broad distribution observed in Thailand, India, Europe, USA and to some extent in Japan. Genotype 4 has been distinguished as the dominant genotype amongst infected individuals from the Middle East, North Africa particularly Egypt where it was observed with a high population prevalence (Akhtar and Moatter, 2004). In fact it is reported that HCV Genotypes are mostly prevalent in Asian and African underdeveloped nations (Méndez-Sánchez et al., 2008). On the other side, genotypes 5 and 6 indicate a limited geographical spread. The genotype 5 has been reported in South Africa while the genotype 6 was found in Macau, Vietnam and Hong Kong (Robertson et al., 1998). Furthermore, severity level is also reported in some of the countries ((Méndez-Sánchez et al., 2008; Méndez-Sánchez et al., 2010; Chávez-Tapia et al., 2012). Investigation of the epidemiology of HCV infections play a significant role in the schemes of its prevention (Strader et al., 2004; Sherman et al., 2007). Study of genotypes is clinically important because different genotypes are relevant to vaccine development, epidemiological questions as well as for the clinical management of HCV infection (Liew et al., 2004; Zein, 2000; Zein and Persing, 1996).

Table 5.2: NP-FiLM BBN vs. peer scoring function for classification of Genotype from nucleotide sequence in SAARC

Region	NP-FiLM	Bayes	BDEU	MDL	Entropy	AIC	fCLL
Bangladesh	55.56	55.56	55.56	55.56	55.56	55.56	55.56
India	48.84	47.94	48.66	47.59	46.15	48.30	46.15
Myanmar	56.45	58.06	55.91	55.38	34.95	55.91	48.39
Nepal	37.50	37.50	37.50	37.50	37.50	37.50	37.50
Pakistan	74.20	69.04	68.30	68.06	74.20	67.32	74.20
SriLanka	57.89	57.89	57.89	57.89	21.05	57.89	57.89

The broader genetic variability of the virus genome has motivated to raise the research question. The problem of identification of structurally similar pattern can be reduced to the structure learning of hepatitis nucleotide sequence dataset using PM and NP-FiLM. The subtype (Genotype) of HCV in this study encompass six regional countries of SAARC as mentioned previously. The genotype which were available in significantly large numbers include DNA strain 1, 1a, 1b, 1c, 2, 2a, 3, 3a, 3b, 3c, 3d, 3e, 3f, 3g, 3i, 3k, 4a, 4d, 6, 6f, 6m and 6n; all of them were retrieved from US official web site for strain data sets (HCV Sequence Database, 2005). These sequences were stored in a dataset where four new basic statistical features were also created. These include the probability distribution of four nucleic acid bases which are elementary constituent of DNA. The A, C, G, T stands for Adenine, Cytosine, Guanin and Thymine. Adenine always pairs with Thymine while Cytosine is paired with Guanine in the opposite strands of the helical pair of strands in DNA. This formulates a dataset in which there were five features including genome sequence and probability distribution of four features of nucleic bases each. The class comprised of twenty genotype observed in various sequence of HCV from SAARC countries. The classification problem was to classify the genotype of any new sequence. As the genome project has been completed, a lot of genomic sequences have been made available. This motivates the research community to analyze these sequences, and find out useful patterns with respect to their sequence structure. Under this motivation, this study validates the performance of PM and NP-FiLM whether they can classify new instances. Table 5.2 indicates the comparison of accuracy of the proposed measure vs. other peer scoring metrics. It can be observed that in two cases, the NP-FiLM delivers significant results. In some cases, there was a tie observed. Here the role of PM was also very important. As we already described in the previous chapters that GBN usually gives better performance as it can cope up the overfitting and underfitting in better way as compared to simple BN in which each node is attached to a class node only. Application of PM ensures the ordering in a way that GBN was realized in all cases of NP-FiLM, however we noticed that in some cases of MDL and AIC, a simple BN was realized which can explain its comparatively lower performance.

Chapter 6

CONCLUSION & FUTURE WORK

Classification is an important technique in expert systems to support the domain experts for identifying knowledge out of large volume of data. The performance of such expert systems is greatly influenced by the accuracy of the core classifier used in the underlying design of the system. In classification, structure prediction from Bayesian inference model is a common practice for the purpose of retrieving hidden rules from masses of data. This process consists of two steps broadly. First step deals in the construction of best suitable structure from the data. The second part deals in the inference from this structure. We in this study have focused on the first part which comprised of construction of most suitable and learning relevant network structure. The core part in the design of a BBN classifier is to introduce a discriminant functions within vector space of attributes through utilization of a priori knowledge. The effectiveness of the Bayesian belief network using greedy heuristics like K2 searching mechanism has enabled its excellent place in the domain of classification systems. In this thesis we centered on approaches for solving two distinctive tasks, both of which necessitated structure learning of BBN from data set.

- First: We argued over various scoring functions including BDeu, AIC, Entropy, BIC, MDL and a recently introduced fCLL on the ground of over fitting while introducing a new parameter free decomposable measure in the domain of structure learning. Theoretically, application of mutual information in structure learning is not a novel idea as it was introduced some six decades ago (Chow and Liu, 1968; Pearl, 1988). We in this study, describes a novel decomposable scoring function for task of structure learning. The introduced measure, known as Non Parametric Factorized Likelihood Metric (NP-FiLM) is characterized by the mutual dependence approximated by marginal and joint probability. The novel measure is particularly designed for discriminative learning because it is decomposable with the capability to permit efficient estimation of structure learning. The accuracy merit of NP-FiLM is evaluated and compared to the common state of-the-art scoring measures given a reasonable size of benchmark data sets obtained from the UCI repository. NP-FiLM performed better than generatively-trained Bayesian network induction algorithm using K2 searching algorithm and numerous scoring function. The proposed measure is expected to build the realistic network which is likely to tally with the practical thinking of field experts in the domain of knowledge engineering.

- Second: In pursuit of improvement in the accuracy of classification system, one significant and initiating step is to build the efficient feature reduction strategy. However, there might be situations where the domain expert is interested in retaining all of the features. At this point, idea of feature ranking becomes more useful and interesting. The concept of feature ranking is limited to those classifiers which are quite sensitive to the initial ordering of the input features (BBN and Random Forest). Although some well known feature ranking techniques are already available; however, we have shown that still more improvement in feature ranking is possible while addressing the enhanced accuracy of the classifier. In feature ranking, our proposed technique based on introduced measure Polarization Measure (PM) proves itself efficient in Random Forest and notable Bayesian Belief Network classifiers with the experimental results presented. Previously, chi square and information gain feature ranking algorithm were shown to be effective in producing better results. However previous results (Hruschka and Ebecken, 2007) were having asymptotic complexity of $O(n^2)$, but we have improved it up to $\Theta(nxLog(n))$ in worst case. We in this study have shown our result to comparatively better to not only chi square and info gain but also some other well known ranking techniques. Moreover, previously the technique was restricted to only BBN classifier but we applied our technique to both BBN and Random Forest. Thus, achieving better results in circumstances with inclusion of all of the features by a machine learning technique seems to justify the proposed technique.
- An avenue of further research in the improvement of BN induction algorithm is to extend its capability for a two time slice dynamic BN model in which node variables are related to each other given certain adjacent time steps. The application of dynamic BN has already proved its stable popularity in various domain like robotics, speech recognition and many more.

- Another extension of this work is possibility of developing a specific class oriented scoring function (metric). Such a scoring metric will be quite useful in the medical diagnostic system where optimal treatment of the patients with identification and classification of a specific class is required only. Clinical data of cancer holds a strong candidature for such a system. Regarding ranker measure, the extension is also possible in a way that current ranker is centralized towards improvement of all of the classes. In situations if it is inadequate for certain class-specific ranker be introduced such that it permits ranking possibly a different order of features for every class.
- A practical application oriented extension of this study is in the domain of data cubes. Data cubes are used to solve the problem of computing the queries to facilitate the business intelligence and decision support system. It is a known fact that BBN is far more compact than a data cube of huge volume. Its amazing level of compactness as compared to data cubes has opened its place to be used instead of conventional data cubes. However, use of BBN as data cubes require the training model be built on complete data set and secondly it must be high in accuracy as we are not concerned with test data in using BBN in capacity of data cubes. We can extrapolate on the basis of its mathematical rigor that the scoring metric proposed in this thesis has the potential to be tweaked into tailoring into a training model oriented scoring metric which can serve an ideal replacement of data cubes.
- The third contribution is towards the analysis of existing state of art classifiers for which some useful insights were delivered. In machine learning , selecting the optimized classification system for a particular system is a non trivial task. However, meta characteristics of a dataset are somewhat useful to give an insight towards the selection of right algorithm for the right problem. While comparing the result of our technique, we also discussed other classifiers for which a meta characteristic based comparison was introduced. We investigated that the pair wise level of our proposed measure PM is quite useful in helping towards the approximate accuracy of the Decision Stump algorithm. This investigation was carried out by regression analysis of nonlinear curve fitting of tenth degree polynomials.
- This analysis can be improved by incorporating a more refined breed of meta characteristics agents. This refined breed of agents are those simple and very fast

classifiers for a very large number of versatile datasets. A collaborative framework can be established by means of these agents to yield a better solution towards the challenging task analogous to prevention from “a square peg in a round hole” in the realm of machine learning.

REFERENCES

[1] Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on, 19*(6), 716-723.

[2] Akhtar, S., & Moatter, T. (2004). Intra-household clustering of hepatitis C virus infection in Karachi, Pakistan. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, *98*(9), 535-539.

[3] Aldrich, J., (1995). Correlations Genuine and Spurious in Pearson and Yule, *Statistical Science,* Vol. 10 (4): pp.364–376.

[4] Almuallim, H., & Dietterich, T. G. (1991, July). Learning with many irrelevant features. In *Proceedings of the ninth National conference on Artificial intelligence* (Vol. 2, pp. 547-552).

[5] Aman, W., Mousa, S., Shiha, G., & Mousa, S. A. (2012). Current status and future directions in the management of chronic hepatitis C. *Virol J*, *9*, 57.

[6] Bagdonavicius, V., Kruopis J. & Nikulin, M. S., (2011). Non-parametric tests for complete data, *ISTE & WILEY*: London & Hoboken.

[7] Berks, M., Chen, Z., Astley, S., & Taylor, C. (2011, January). Detecting and classifying linear structures in mammograms using random forests. In *Information Processing in Medical Imaging* (pp. 510-524). Springer Berlin Heidelberg.

[8] Bi, C., & Chen, G. (2011). Bayesian Networks Modeling for Crop Diseases. In *Computer and Computing Technologies in Agriculture IV* (pp. 312-320). Springer Berlin Heidelberg.

[9] Bozdogan, H. (1987). Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, *52*(3), 345-370.

[10] Breiman L., (2001). Random forests, *Machine learning*, *45*(1), 5-32.

[11] Breiman L., (1996). Bagging predictors, Machine Learn. 24 (2), 123–140.

[12] Breiman L., Friedman J. H., Olshen R.A. & Stone C. J., (1984). Classification and Regression Trees, Wadsworth, Belmont, CA.

[13] Brodley, C. E., & Utgoff, P. E. (1995). Multivariate decision trees. *Machine learning*, *19*(1), 45-77.

[14] Buntine, W. (1991, July). Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence* (pp. 52-60). Morgan Kaufmann Publishers Inc.

[15] Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *Knowledge and Data Engineering, IEEE Transactions on*, *8*(2), 195-210.

[16] Buntine, W., (1992). Learning classification trees, *Statistics and computing*, *2*(2), 63-73

[17] Carvalho, A. M., Roos, T. T., Oliveira, A. L., & Myllymäki, P. (2011). Discriminative learning of bayesian networks via factorized conditional log-likelihood. *Journal of machine learning research*.

[18] Carvalho, A. M., (2009). Scoring functions for learning bayesian networks, *Inesc-id , Technical report, INESC-ID Tec, INESC-ID Tec*. Rep.2009: 54.

[19] Carvalho A. M., Oliveira A. L. & Sagot M. F., (2007). Efficient learning of Bayesian network classifiers: an extension to the tan classifier, *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*, pp. 16-25.

[20] Chávez-Tapia, N. C., Barrientos-Gutiérrez, T., Guerrero-López, C. M., Santiago-Hernández, J. J., Méndez-Sánchez, N., & Uribe, M. (2012). Increased mortality from acute liver failure in Mexico. *Annals of hepatology*, *11*(2), 257.

[21] Cheng, J., Bell, D. A., & Liu, W. (1997). An algorithm for Bayesian belief network construction from data. In *proceedings of AI & STAT'97* (pp. 83-90).

[22] Cheng, J., & Greiner, R. (1999, July). Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 101-108). Morgan Kaufmann Publishers Inc.

[23] Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, *14*(3), 462-467.

[24] Chickering, D. M., Heckerman, D., & Meek, C. (2004). Large-sample learning of Bayesian networks is NP-hard. *The Journal of Machine Learning Research*, *5*, 1287-1330.

[25] Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, *9*(4), 309-347.

[26] Corder, G. W., & Foreman, D. I. (2009). *Nonparametric statistics for non-statisticians: a step-by-step approach*. John Wiley & Sons. ISBN-13: 978-0470454619.

[27] Cormen, T. H., (2009). Introduction to Algorithms (3rd ed.). Massachusetts Institute of Technology, pp. 253–280. ISBN 978-0-262-03384-8.

[28] Correa, E. S., & Shapiro, J. L. (2006). Model complexity vs. performance in the bayesian optimization algorithm. In *Parallel Problem Solving from Nature-PPSN IX* (pp. 998-1007). Springer Berlin Heidelberg.

[29] DataFit, (1995-2012) Oakdale Engineering, www.curvefitting.com (accessed on April 2013).

[30] de Campos, L. M. & Castellano, J. G. (2007). Bayesian network learning algorithms using structural restrictions, *International Journal of Approximate Reasoning*, *45*(2), 233-254.

[31] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine learning*, *40*(2), 139-157.

[32] Drugan, M. M. & Wiering, M. A. (2010). Feature selection for Bayesian network classifiers using the MDL-FS score, *International journal of approximate reasoning*, *51*(6), 695-717.

[33] Dumais, S., Platt J., Heckerman, D. & Sahami M., (1998, November). Inductive learning algorithms and representations for text categorization, *Proceedings of the seventh international conference on Information and knowledge management* (pp. 148-155). ACM.

[34] Esmeir, S., & Markovitch, S., (2006, July). Anytime induction of decision trees: An iterative improvement approach, *Proceedings Of The National Conference On Artificial Intelligence* (Vol. 21, No. 1, p. 348). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[35] Fan, W., (2004). On the optimality of probability estimation by random decision trees, *AAAI*, 2004, pp. 336–341.

[36] Fan, W., Greengrass E., McCloskey J., Yu P. and Drummey K., (2005). Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches, pp. 154–161.

[37] Fayyad U. M., Irani K. B., (1992). On the handling of continuous-valued attributes in decision tree generation, Mach. Learn. 8: 87–102.

[38] Fayyad U., and Irani K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. Thirteenth International Joint Conference on Articial Intelligence, 1022-1027, 1993.

[39] Fischer, S. (1977). Long-term contracts, rational expectations, and the optimal money supply rule, *The Journal of Political Economy*, 191-205.

[40] Frank, A., & Asuncion, A. (2010). *UCI Repository of machine learning databases*. Tech. Rep., Univ. California, Sch. Inform. Comp. Sci., Irvine, CA, 2010. Available from. http://www.ics.uci.edu/~mlearn/{MLR}epository.html. Accessed May, 2013.

[41] Freund, Y. & Mason, L., (1999). The alternating decision tree learning algorithm, In *ICML'99*, 124–133. Morgan Kaufmann, San Francisco, CA.

[42] Freund, Y. & Schapire, R. E., (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of computer and system sciences*, *55*(1), 119-139.

[43] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337-407.

[44] Friedman, N., Geiger, D. & Goldszmidt M., (1997). Bayesian network classifiers, Machine Learning 29:131–163.

[45] Friedman, N., Goldszmidt, M., (1996). Learning Bayesian networks with local structure, Proc. 12th Intern. Conf. on Uncertainty in Artificial Intelligence (UAI'96), Portland, Oregon, USA, Morgan Kaufmann, SF, pp. 252–262.

[46] Gibbons, J. D. & Chakraborti, S. (2003). *Nonparametric statistical inference* (Vol. 168). 4th Ed, CRC press, ISBN 0824740521.

[47] Grimmett, G. and Stirzaker, D., (2001). Probability and Random Processes, Oxford University Press.

[48] Guo, Y. and Schuurmans, D., (2012). Convex structure learning for Bayesian networks: Polynomial feature selection and approximate ordering, arXiv preprint arXiv: 1206.6832.

[49] Hall, M., Frank, E., Holmes, G., Pfahringer B., Reutemann, P., Witten, I. H., (2009). The Weka data mining software: an update, ACM SIGKDD Explorations, Vol. 11, pp. 10-18

[50] Hall, M. A. & Holmes, G., Benchmarking attribute selection techniques for discrete class data mining, *IEEE Transactions on Knowledge and Data Engineering*, 15(6), pp. 1437-1447, 2003.

[51] Hall, M. A., (2000). Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, Proc. 17th Int'l Conf. Machine Learning (ICML2000).

[52] HCV Sequence Database, wesite 2005 [cited 2013 January]; Available from: URL:http://hcv.lanl.gov/content/index.

[53] Heckerman, D., Geiger D. & Chickering D. M., (1995). Learning Bayesian networks: The combination of knowledge and statistical data, Machine learning, 20(3), pp. 197-243.

[54] Heckerman, D., (1995). A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95–06, Microsoft Research.

[55] Heckerman, D., (2008). A tutorial on learning with Bayesian networks, Innovations in Bayesian Networks, pp. 33-82.

[56] Herskovits, E. & Cooper, G., (1990). *Kutató: An entropy-driven system for construction of probabilistic expert systems from databases,* Knowledge Systems Laboratory, Medical Computer Science, Stanford University.

[57] Hesar, A. S., Tabatabaee, H., & Jalali M., (2012). Structure Learning of Bayesian Networks Using Heuristic Methods, *2012 International Conference on Information and Knowledge Management (ICIKM 2012) IPCSIT vol.45 (2012) © (2012) IACSIT Press, Singapore.*

[58] Holmes, G., Pfahringer, B., Kirkby R., Frank E., & Hall M., (2002). Multiclass alternating decision trees, *Machine Learning: ECML 2002* (pp. 161-172). Springer Berlin Heidelberg.

[59] Holte, R. C., (1993). Very simple classification rules perform well on most commonly used datasets, Machine Learning, Issue (11) pp. 63-91.

[60] Hruschka, Jr, E. R., & Ebecken, N. F., (2007). Towards efficient variables ordering for Bayesian networks classifier, *Data & Knowledge Engineering*, *63*(2), 258-269.

[61] Iba, W. & Langley, P., (1992). Induction of one-level decision trees, Proc. of the Ninth International Machine Learning Conference (1992). Aberdeen, Scotland: Morgan Kaufmann.

[62] Jensen, F. V. & Nielsen, T. D., (2007). Bayesian networks and decision graphs, *Information Science and Statistics,* Volume. ISBN 978-0-387-68281-5, Springer New York.

[63] John, G. H. & Langley, P., (1995, August). Estimating continuous distributions in Bayesian classifiers, In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338-345). Morgan Kaufmann Publishers Inc.

[64] Jung, Y. G. & Choi, Y. J., (2013). Using Augmented Bayesian Networks to Compare Preference of Performance, International Journal of Bio-Science and Bio-Technology Vol. 5, No. 1, February.

[65] Kabir, M. F., Rahman, C. M., Hossain, A. & Dahal, K., (2011). Enhanced Classification Accuracy on Naive Bayes Data Mining Models, *International Journal of Computer Applications, Foundation of Computer Science, New York, USA*, *28*(3), 9-16.

[66] Keogh, E. J. & Pazzani, M. J., (2002). Learning the structure of augmented Bayesian classifiers, International Journal on Artificial Intelligence Tools 11:587–601.

[67] Kira, K., & Rendell, L., (1992). A Practical Approach to Feature Selection, Proc. Ninth Int’l Conf. Machine Learning, pp. 249-256.

[68] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P., (1983). Optimization by simulated annealing, Science 220 (4598): 671–680.

[69] Kobyliński, Ł., & Przepiórkowski, A., (2008). Definition extraction with balanced random forests, In *Advances in Natural Language Processing* (pp. 237-247). Springer Berlin Heidelberg.

[70] Kohavi, R. & Kunz, C., (1997, July). Option decision trees with majority votes, In *Machine Learning-International Workshop Then Conference-* (pp. 161-169). Morgan Kaufmann Publishers, Inc.

[71] Kohavi, R. & John, G., (1997). Wrapper for Feature Subset Selection, *Artificial Intelligence*, vol. 97, nos. 1-2, pp. 273-324, 1997.

[72] Kononenko, I., (1994). Estimating attributes: analysis and extensions of RELIEF, Machine Learning: ECML-94 pp. 171-182. Springer Berlin/Heidelberg.

[73] Kononenko, I., Simec, E., RobnikSikonja, M., (1997). Overcoming the myopia of inductive learning algorithms with RELIEFF, Appl. Intell. 7, 39–55

[74] Kyperountas, M., Tefas, A., Pitas, I., (2007). Weighted piecewise LDA for solving the small sample size problem in face verification, IEEE Trans. Neural Netw. 18 (2), 506–519.

[75] Lam, W. & Bacchus, F., (1994). Learning Bayesian belief networks: an approach based on the MDL principle, Computational Intelligence, 10, No. 4, 269–293.

[76] Lamma, E., Riguzzi, F. & Storari, S., (2004). Exploiting association and correlation rules-parameters for improving the k2 algorithm, In *ECAI*, vol. 16, pp. 500.

[77] Lamma, E., Riguzzi, F. & Storari, S., (2005). Improving the K2 Algorithm Using Association Rule Parameters, *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU04)*, 1667-1674.

[78] Lauritzen, S. L. & Spiegelhalter, D. J., (1988). Local computation with probabilities and graphical structures and their application to expert systems, J. Royal Statistical Society B, 50:157–224, 1988.

[79] Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied statistics*, 191-201.

[80] Lee, S. E., Moore, J. K., Holmes, A., Umezu, K., Kolodner, R. D., & Haber, J. E. (1998). *Saccharomyces* Ku70, Mre11/Rad50, and RPA Proteins Regulate Adaptation to G2/M Arrest after DNA Damage. *Cell*, *94*(3), 399-409.

[81] Lerner, B. & Malka, R., (2011). Investigation of the K2 algorithm in learning Bayesian network classifiers, *Applied Artificial Intelligence*, *25*(1), 74-96.

[82] Liew, M., Erali, M., Page, S., Hillyard, D., & Wittwer, C. (2004). Hepatitis C genotyping by denaturing high-performance liquid chromatography. *Journal of clinical microbiology*, *42*(1), 158-163.

[83] Liu, H., & Setiono, R., (1996). A probabilistic approach to feature selection-a filter solution, machine learning-international workshop conference, Morgan kaufmann publishers, inc. July, pp. 319-327 .

[84] Liu, S., Zhu, M. & Yang, Y., (2013). A Bayesian Classifier Learning Algorithm Based on Optimization Model, *Mathematical Problems in Engineering.*

[85] Liu, Y. & Han, J. D. J., (2010). Application of Bayesian networks on large-scale biological data, *Frontiers in Biology*, *5*(2), 98-104.

[86] Liu, Z., Malone B. & Yuan, C., (2012). Empirical evaluation of scoring functions for Bayesian network model selection, *BMC bioinformatics*, *13*(Suppl 15), S14.

[87] Madden, M. G. (2003). The performance of Bayesian network classifiers constructed using different techniques, In Working Notes of the ECML=PKDD-03 Workshop on Probuhilistic Graphical Models for Classification, 59–70.

[88] Madden, M. G. (2009). On the classification performance of TAN and general Bayesian networks, *Knowledge-Based Systems*, *22*(7), 489-495.

[89] Mao, K.Z., (2002). RBF neural network center selection based on Fisher ratio class separability measure, IEEE Trans. Neural Networks 13 (5) (2002) 1211–1217.

[90] Martins, J. F., Pires, V. F., Pires, A. J., (2007). Unsupervised neural-network-based algorithm for an online diagnosis of three-phase induction motor stator fault, IEEE Trans. Ind. Electron. 54 (1), 259-264.

[91] Méndez-Sánchez, N., Villa, A. R., Vázquez-Elizondo, G., Ponciano-Rodríguez, G., & Uribe, M. (2008). Mortality trends for liver cancer in Mexico from 2000 to 2006. *Ann Hepatol*, *7*(3), 226-9.

[92] Méndez-Sánchez, N., García-Villegas, E., Merino-Zeferino, B., Ochoa-Cruz, S., Villa, A. R., Madrigal, H., & Uribe, M. (2010). Liver diseases in Mexico and their associated mortality trends from 2000 to 2007: A retrospective study of the nation and the federal states. Ann of Hepatol 2010. 9(4): 428–438.

[93] Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U. & Hamprecht, F. A. (2011). On oblique random forests, *Machine Learning and Knowledge Discovery in Databases* (pp. 453-469). Springer Berlin Heidelberg.

[94] Mitchell, T. M., (1997). Machine Learning, McGraw-Hill, Singapore.

[95] Nadeau, C. & Bengio, Y., (2003). Inference for the Generalization Error. Machine Learning Vol. 52, Issue 3, pp. 239-281.

[96] Naeem, M. & Asghar, S., (2013a). A Novel Feature Selection Technique For Feature Order Sensitive Classifiers, Anale. Seria Informatica. Annals. Computer Science Series, Tome 11, Fasc. 1 (To be published in June 2013 Issue) http://anale-informatica.tibiscus.ro/?page=11_numarcurent&lang=en

[97] Naeem, M. & Asghar, S., (2013b). An Information Theoretic Scoring Function in Belief Network, *The International Arab Journal of Information Technology*, (Accepted, In press for vol.11 (5)).

[98] Nayak, J. R., & Cook, D. J. (2001, May). Approximate association rule mining, *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference* (pp. 259-263). AAAI Press.

[99] Ogutu, J. O., Piepho, H. P. & Schulz-Streeck, T., (2011, May). A comparison of random forests, boosting and support vector machines for genomic selection, *BMC proceedings* (Vol. 5, No. Suppl 3, p. S11). BioMed Central Ltd.

[100] Oza, N. C. & Tumer, K., (2008). Classifier ensembles: Select real-world applications, *Information Fusion*, *9*(1), 4-20.

[101] Ozcift, A., (2012). Enhanced Cancer Recognition System Based on Random Forests Feature Elimination Algorithm, *Journal of medical systems*, *36*(4), 2577-2585.

[102] Qing-Yun, S., Fu, K. S., (1983). A method for the design of binary tree classifiers, Pattern Recog. 16 (6). 593–603.

[103] Quinlan, J. R., (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA.

[104] Quinlan, J. R., (1996). Improved use of continuous attributes in C4.5, J. Artif. Intell. Res. 4. 77–90.

[105] Quinlan, J. R., (1986). Induction in decision trees, Mach. Learn. 1 (1). 81–106.

[106] Pearl, J., (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Fransisco, 1988.

[107] Pearl, J., & Verma, T. S., (1991). A statistical semantics for causation. Statistics and Computing, 2:91–95.

[108] Pelikan, M., (2005). Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms, Springer, Berlin, 2005.

[109] Pelikan, M. & Goldberg, D., (2006). Hierarchical Bayesian optimization algorithm, Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications, Springer, pp. 63-90.

[110] Pennock, D., (1998). Logarithmic time parallel Bayesian inference, Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, pages 431–438.

[111] Rajaram, R., Pramala S., Rajalakshmi, S., Jeyendran, C. & Prakash, D.S., J. (2011). NB+: An improved Naïve Bayesian algorithm, *Knowledge-Based Systems*, *24*(5), 563-569.

[112] Rebane, G., & Pearl, J., (1988). The recovery of causal poly-trees from statistical data, Intern. J. Approx. Res., 2, No. 3,175–182.

[113] Robertson, B., Myers, G., Howard, C., Brettin, T., Bukh, J., Gaschen, B., & Weiner, A. (1998). Classification, nomenclature, and database development for hepatitis C virus (HCV) and related viruses: proposals for standardization. *Archives of virology*, *143*(12), 2493-2503.

[114] Robnik-Šikonja, M., (2004). Improving random forests, *Machine Learning: ECML 2004* (pp. 359-370). Springer Berlin Heidelberg.

[115] Rodrıguez, J. J., & Maudes, J. M., (2006). Ensembles of grafted trees, *ECAI-06, Proceedings of the 17th European Conference on Artificial Intelligence* (Vol. 141, pp. 803-804).

[116] Schwarz, G. E., (1978). Estimating the dimension of a model, *Annals of Statistics*, 6(2): 461–464.

[117] Sears Project, http://repository.seasr.org/Datasets/UCI/arff/, accessed May 2013.

[118] Sherman, M., Shafran, S., Burak, K., Doucette, K., Wong, W., Girgrah, N., & Deschênes, M. (2007). Management of chronic hepatitis C: consensus guidelines. *Canadian Journal of gastroenterology*, *21*(Suppl C), 25C-34C.

[119] Shi, H., (2007). *Best-first decision tree learning* (Doctoral dissertation, The University of Waikato).

[120] Silander, T, Myllymäki P. (2012). A simple approach for finding the globally optimal Bayesian network structure. *arXiv preprint arXiv:1206.6875.*

[121] Silander, T, Myllymäki P. (2006). A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. Proc 22nd Conf Uncertainty Artif Intell.

[122] Simon, G., Berger M. O., (1998). A two-stage robust statistical method for temporal registration from features of various types. IEEE Sixth International Conference on Computer Vision, pp. 261–266.

[123] Singh, M. & Valtorta M., (1995). Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. International Journal of Approximate Reasoning 12:111–131.

[124] Spirtes, P., Glymour, C., & Scheines, R., (2000). Causality, prediction and search, 2nd ed. Cambridge, MA: MIT Press.

[125] Strader, D. B., Wright T, Thomas DL, Seeff LB. Diagnosis, management, and treatment of hepatitis C. Hepatology. 2004;39(4):1147-71.

[126] Suzuki, J., (1999). Learning Bayesian belief networks based on the MDL principle: An efficient algorithm using the branch and bound technique, IEICE Transactions on Information and Systems, 82(2), pp. 356-367.

[127] Terent'ev, A. N., Bidyuk, P. I., (2006). A heuristic method to construct Bayesian networks, Mat. Mash. Sist., 3, 12–23 (2006).

[128] Tumer, K. & Ghosh, J., (1996). Error correlation and error reduction in ensemble classifiers, Connection Science (Special Issue on Combining Artificial Neural Networks: Ensemble Approaches) 8 (3–4). 385–404.

[129] Vapnik, V., (1999). *The nature of statistical learning theory*, springer.

[130] Vigdor, B., Lerner, B., (2006). Accurate and fast off and online fuzzy ARTMAP-based image classification with application to genetic abnormality diagnosis, IEEE Trans. Neural Netw. 17 (5), 1288–1300.

[131] Wasserman, L., (2007). All of nonparametric statistics, Springer, (2007). Wiley ISBN 0387251456.

[132] Webb, G. I. (1997, August). Decision tree grafting, *International Joint Conference On Artificial Intelligence* (Vol. 15, pp. 846-851). Lawrence Erlbaum Associates Ltd.

[133] Webb, G. I. (1999, July). Decision tree grafting from the all-tests-but-one partition, *International Joint Conference on Artificial Intelligence* (Vol. 16, pp. 702-707). Lawrence Erlbaum Associates.

[134] Wolpert, D. H., (1997). On bias plus variance, Neural Computation 9 (6): 1211–1243.

[135] Wolpert, D.H., Macready, W.G. (1997), No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation* 1, 67.

[136] Xu, H., Yang J., Jia, P. & Ding Y., (2013). Effective Structure Learning for Estimation of Distribution Algorithms via L1-Regularized Bayesian Networks, *Int J Adv Robotic Sy*, *10*(17).

[137] Xu, R, Wunsch, D. Survey of Clustering Algorithms, IEEE Transactions on Neural Networks, May 2005; 16(3).

[138] Yang, B. S., Di, X., & Han, T. (2008). Random forests classifier for machine fault diagnosis. *Journal of mechanical science and technology*, *22*(9), 1716-1725.

[139] Yang, L., & Lee, J., (2012). Bayesian Belief Network-based approach for diagnostics and prognostics of semiconductor manufacturing systems, *Robotics and Computer-Integrated Manufacturing*, *28*(1), 66-74.

[140] Yang, Y., Webb, G. I. & Wu X., (2010). Discretization methods, *Data Mining and Knowledge Discovery Handbook* (pp. 101-116). Springer US.

[141] Yang, Y., & Pedersen, J. O., (1997). A comparative study on feature selection in text categorization, Machine learning-international workshop then conference, pp. 412-420, morgan kaufmann publishers, inc.

[142] Yehezkel, R., & Lerner, B., (2009). Bayesian network structure learning by recursive autonomy identification, Journal of Machine Learning Research 10:1527–1570.

[143] You, K. C. & Fu, K. S., (1976). An approach to the design of a linear binary tree classifier, Proceedings of the Third Symposium on Machine Processing of Remotely Sensed Data, IEEE Press, New York, pp. 3a1–3a10.

[144] Zein, N. N. (2000). Clinical significance of hepatitis C virus genotypes. *Clinical microbiology reviews*, *13*(2), 223-235.

[145] Zein, N. N., & Persing, D. H. (1996, May). Hepatitis C genotypes: current trends and future implications. In *Mayo Clinic Proceedings* (Vol. 71, No. 5, pp. 458-462). Elsevier.

[146] Zgurovskii, M. Z., Bidyuk, P. I., & Terent'ev, A. N. (2008). Methods of constructing Bayesian networks based on scoring functions. *Cybernetics and Systems Analysis*, *44*(2), 219-224.

Appendix A

Table A.1: NP-FiLM BBN vs. peer scoring function using K2 (max. parent : 3)

Dataset	NpFLM	Bayes	AIC	BDEU	MDL	Entropy	fCLL
arrhythmia	70.80	69.69	70.80	70.13	71.02	68.14	72.12
audiology	78.76	76.55	76.11	73.01	76.11	75.22	73.89
autos	80.98	80.49	74.63	83.41	74.63	79.02	80.49
balance-scale	72.64	72.80	73.44	71.84	72.00	73.76	73.76
breast-cancer	70.98	70.98	68.53	69.58	70.63	64.34	63.29
breast-w	96.71	96.71	96.85	96.57	97.00	96.28	97.14
bridges_version1	65.71	65.71	65.71	65.71	65.71	41.90	59.05
bridges_version2	63.81	64.76	62.86	64.76	60.95	41.90	54.29
car	91.61	90.80	92.65	90.80	85.71	91.49	91.49
colic	82.34	80.98	82.07	82.07	81.52	74.73	79.62
colic.ORIG	78.53	79.89	78.26	79.08	78.53	65.22	66.58
credit-a	85.94	85.07	85.65	85.80	86.23	83.33	84.64
credit-g	74.60	74.90	74.70	75.00	75.30	71.20	71.90
cylinder-bands	77.41	75.93	76.85	77.22	77.96	0.00	
dermatology	97.54	98.09	97.54	97.54	97.54	93.72	94.54
diabetes	74.74	74.48	74.09	75.13	74.87	73.96	74.48
flags	61.34	58.25	61.34	57.73	62.37	35.57	59.79
glass	71.03	72.43	70.56	69.16	70.56	73.36	75.70
haberman	73.86	72.55	72.55	72.55	72.55	73.86	73.86
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	85.03	84.35	85.03	84.01	81.97	83.33
heart-statlog	81.48	81.85	82.22	80.74	80.37	81.85	80.37
iris	92.67	92.67	92.67	92.67	92.67	90.00	92.67
kdd_synthetic_control	98.67	98.83	98.00	97.67	97.17	16.67	93.00
labor	94.74	92.98	91.23	89.47	91.23	91.23	85.96
letter	84.53	86.46	83.97	81.71	76.62	88.46	88.65
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	68.75	71.88	75.00	75.00	78.13	68.75
mfeat-fourier	79.85	80.75	80.30	77.80	78.05	79.80	79.85
mfeat-karhunen	92.75	93.05	93.15	92.10	92.05	90.35	91.25
mfeat-morphological	70.20	69.85	67.95	68.85	68.20	69.70	67.95
mfeat-pixel	94.75	94.55	94.00	93.55	93.40	95.10	95.80
molecular-biology_promoters	33.02	29.25	28.30	30.19	29.25	30.19	30.19
mushroom	99.74	99.96	100.00	99.98	100.00	99.96	99.67
page-blocks	95.47	96.40	95.30	96.18	95.63	96.46	95.89

Table A.1: NP-FiLM BBN vs. peer scoring function using K2 (max. parent : 3) (continue…)

Dataset	**NpFLM**	**Bayes**	**AIC**	**BDEU**	**MDL**	**Entropy**	**fCLL**
pendigits	94.78	96.56	95.26	95.14	93.25	97.22	97.17
postoperative-patient-data	64.44	61.11	65.56	64.44	64.44	61.11	63.33
segment	95.32	95.28	94.85	94.63	91.39	96.06	92.77
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	93.33	93.33
sonar	77.88	77.88	78.37	76.92	79.81	78.85	79.81
spect_test	70.05	69.52	66.84	68.98	71.12	65.24	67.38
spect_train	68.75	63.75	66.25	65.00	68.75	57.50	55.00
splice	95.17	95.55	94.73	95.52	95.64	52.57	52.57
sponge	94.74	94.74	93.42	94.74	93.42	90.79	92.11
tae	47.02	47.02	47.02	47.02	47.02	47.02	47.02
tic-tac-toe	75.89	76.62	80.69	73.80	73.80	80.90	81.94
trains	60.00	60.00	60.00	60.00	60.00	60.00	80.00
waveform-5000	82.60	81.72	81.36	81.48	81.54	80.02	82.24
zoo	98.02	95.05	96.04	100.00	94.06	96.04	93.07
	win	**20**	**26**	**27**	**25**	**31**	**30**
	neutral	**13**	**11**	**11**	**12**	**7**	**9**
	lose	**17**	**13**	**12**	**13**	**12**	**11**
Average	**78.51**	**78.13**	**77.98**	**77.92**	**77.58**	**71.81**	**76.42**

Table A.2: NP-FiLM BBN vs. peer scoring function using K2 (max. parent : 2)

Dataset	NpFLM	Bayes	AIC	BDeu	MDL	Entropy	fCLL
arrhythmia	70.80	69.91	71.02	69.47	71.02	70.80	72.79
audiology	78.76	76.11	76.11	73.45	76.11	74.34	74.78
autos	80.98	80.98	74.63	82.44	74.63	80.98	79.51
balance-scale	72.64	73.76	74.08	71.84	72.00	74.24	74.24
breast-cancer	70.98	70.63	69.58	69.58	70.63	68.88	68.53
breast-w	96.71	96.71	96.57	96.57	97.00	96.85	97.14
bridges_version1	65.71	64.76	65.71	65.71	65.71	41.90	58.10
bridges_version2	63.81	65.71	62.86	64.76	60.95	41.90	57.14
car	91.61	90.80	90.80	90.80	85.71	91.61	91.44
colic	82.34	80.16	79.62	82.34	81.52	79.08	79.89
colic.ORIG	78.53	80.16	77.17	80.16	77.99	65.76	67.12
credit-a	85.94	85.65	85.94	85.80	86.23	85.22	84.64
credit-g	74.60	74.30	73.80	75.10	75.30	73.50	73.70
cylinder-bands	77.41	75.37	75.74	77.41	77.96	63.52	67.04
dermatology	97.54	98.09	97.54	97.54	97.54	96.99	97.27
diabetes	74.74	74.09	74.09	75.13	74.87	74.48	74.09
flags	61.34	61.34	61.34	58.25	62.37	35.57	58.76
glass	71.03	73.36	70.56	68.69	70.56	76.17	75.23
haberman	73.86	72.55	72.55	72.55	72.55	73.86	73.86
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	84.69	84.35	85.03	84.01	82.65	84.69
heart-statlog	81.48	82.96	82.22	80.74	80.74	81.11	81.85
iris	92.67	92.67	92.67	92.67	92.67	93.33	92.67
kdd_synthetic_control	98.67	98.83	98.00	97.67	97.17	16.67	92.33
labor	94.74	91.23	92.98	91.23	91.23	89.47	87.72
letter	84.53	83.65	83.97	81.68	76.62	84.44	85.74
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	75.00	71.88	68.75	75.00	75.00	71.88
mfeat-fourier	79.85	80.20	80.30	77.80	78.05	80.15	80.60
mfeat-karhunen	92.75	93.05	93.55	92.10	92.05	92.90	92.65
mfeat-morphological	70.20	69.90	67.95	68.90	68.20	70.05	67.95
mfeat-pixel	94.75	94.40	94.00	93.60	93.40	94.15	96.40
molecular-biology_promoters	33.02	22.64	28.30	30.19	29.25	30.19	30.19
mushroom	99.74	99.54	99.53	99.50	99.47	99.54	99.22
page-blocks	95.47	95.32	95.34	95.49	95.63	95.36	95.16
pendigits	94.78	95.26	95.26	95.14	93.25	95.26	96.12
postoperative-patient-data	64.44	63.33	65.56	64.44	64.44	63.33	64.44
segment	95.32	95.28	94.85	94.76	91.39	94.76	92.47

Table A.2: NP-FiLM BBN vs. peer scoring function using K2 (max. parent : 2) (continue…)

Dataset	NpFLM	Bayes	AIC	BDeu	MDL	Entropy	fCLL
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	93.33	93.33
sonar	77.88	80.29	79.81	79.33	79.81	79.81	81.25
spect_test	70.05	72.19	72.19	67.91	70.59	72.19	67.38
spect_train	67.50	71.25	66.25	66.25	67.50	67.50	65.00
splice	95.17	95.55	95.55	95.52	95.64	52.57	52.57
sponge	94.74	94.74	93.42	94.74	93.42	90.79	93.42
tae	47.02	47.02	47.02	47.02	47.02	47.02	47.02
tic-tac-toe	75.89	74.01	73.70	73.80	73.80	73.80	73.38
trains	60.00	60.00	60.00	60.00	60.00	60.00	70.00
waveform-5000	82.60	81.78	82.08	81.48	81.54	82.38	81.42
zoo	97.03	93.07	96.04	99.01	94.06	96.04	93.07
	win	**23**	**28**	**25**	**28**	**30**	**29**
	neutral	**12**	**11**	**14**	**10**	**10**	**9**
	lose	**15**	**11**	**11**	**12**	**10**	**12**
Average	**78.46**	**78.25**	**77.93**	**77.85**	**77.54**	**73.73**	**76.44**

Table A.3: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing (max. parent : 4)

Dataset	NP-FiLM	Bayes	AIC	BDEU	MDL	Entropy
audiology	78.76	70.80	66.81	69.03	61.95	66.37
autos	80.98	76.59	71.71	78.54	66.83	81.46
balance-scale	72.64	70.88	70.88	71.84	71.84	70.88
breast-cancer	70.98	72.03	70.63	71.68	68.88	67.48
breast-w	96.71	96.71	96.71	96.85	97.00	94.85
bridges_version1	65.71	69.52	68.57	68.57	60.00	41.90
bridges_version2	63.81	70.48	63.81	64.76	61.90	41.90
car	91.61	93.87	93.81	93.00	85.47	94.44
colic	82.34	81.79	83.15	83.42	84.78	77.99
colic.ORIG	78.53	80.43	79.08	81.25	81.52	66.03
credit-a	85.94	85.22	86.09	85.36	86.52	82.17
credit-g	74.60	72.10	72.70	73.60	73.60	68.40
cylinder-bands	77.41		71.30		64.63	
dermatology	97.54	97.27	97.54	97.54	98.09	91.53
diabetes	74.74	75.26	74.74	75.52	75.91	73.70
flags	61.34	58.25	59.79	62.89	57.22	32.47
glass	71.03	72.43	70.56	70.09	70.56	71.50
haberman	73.86	72.88	72.88	72.88	72.88	73.86
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	83.33	79.25	84.01	80.61	79.93
heart-statlog	81.48	82.59	79.63	81.11	82.22	81.48
iris	92.67	93.33	92.67	92.00	92.67	89.33
kdd_synthetic_control	98.67	98.83	97.67	97.33	97.17	42.83
labor	94.74	85.96	91.23	89.47	87.72	92.98
letter	84.53	89.49	85.50	85.04	75.87	
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	65.63	71.88	68.75	81.25	78.13
mfeat-fourier	79.85	79.80	79.75	78.20	77.30	75.70
mfeat-karhunen	92.75	93.70	92.90	92.10	92.10	86.40
mfeat-morphological	70.20	68.95	68.10	68.80	67.95	70.00
mfeat-pixel	94.75					
molecular-biology_promoters	33.02	22.64	27.36	32.08	32.08	33.96
mushroom	99.74	100.00	100.00	100.00	100.00	100.00
page-blocks	95.47	96.40	95.58	96.22	95.43	96.11
pendigits	94.78	96.97	95.82	95.94	92.79	95.07
postoperative-patient-data	64.44	70.00	71.11	71.11	71.11	66.67
segment	95.32	95.37	94.85	94.16	91.56	93.20

Table A.3: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing (max. parent : 4) (continue…)

Dataset	NP-FiLM	Bayes	AIC	BDEU	MDL	Entropy
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	93.33
sonar	77.88	76.92	78.37	79.81	77.88	75.00
spect_test	70.05	65.24	63.64	66.31	69.52	59.89
spect_train	68.75	61.25	63.75	70.00	73.75	60.00
splice	95.17	96.58	94.80		96.24	52.57
sponge	94.74	92.11	93.42	94.74	92.11	92.11
tae	47.02	47.02	47.02	35.10	41.72	47.02
tic-tac-toe	75.89	74.32	86.01	68.89	68.58	85.39
trains	60.00	40.00	50.00	40.00	50.00	80.00
waveform-5000	82.60	81.72	81.62	80.22	80.98	72.64
zoo	97.03	98.02	95.05	96.04	97.03	95.05
	win	22	23	23	29	28
	neutral	6	10	7	7	7
	lose	18	14	16	11	11
Average	78.65	77.29	77.25	76.84	76.36	73.01

Table A.4: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing

(max. parent : 3)

Dataset	NpFLM	Bayes	AIC	BDEU	MDL	Entropy
audiology	78.76	69.47	66.81	69.03	61.95	65.49
autos	80.98	79.02	71.71	80.00	66.83	80.98
balance-scale	72.64		73.44	71.84	71.84	73.44
breast-cancer	70.98	72.03	70.63	71.68	68.88	66.08
breast-w	96.71	96.71	96.71	96.85	97.00	95.71
bridges_version1	65.71	69.52	68.57	68.57	60.00	41.90
bridges_version2	63.81	69.52	63.81	64.76	61.90	41.90
car	91.61	93.87	93.81	93.00	85.47	95.14
colic	82.34	83.15	83.15	83.42	84.78	77.17
colic.ORIG	78.53	80.16	79.08	81.25	81.52	66.03
credit-a	85.94	85.22	86.23	85.36	86.52	83.77
credit-g	74.60	72.20	72.70	73.60	73.60	68.80
cylinder-bands	77.41		71.30		64.63	
dermatology	97.54	97.27	97.54	97.54	98.09	95.63
diabetes	74.74	75.26	74.74	75.52	75.91	73.18
flags	61.34	59.79	60.31	61.86	57.22	32.47
glass	71.03	72.43	70.56	70.09	70.56	73.36
haberman	73.86	72.88	72.88	72.88	72.88	73.86
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	83.33	79.25	84.01	80.61	81.29
heart-statlog	81.48	82.96	80.37	81.11	82.22	78.89
iris	92.67	93.33	92.67	92.00	92.67	90.67
kdd_synthetic_control	98.67	98.83	97.67	97.33	97.17	42.83
labor	94.74	85.96	91.23	89.47	87.72	87.72
letter	84.53	89.47		85.04	75.87	89.67
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	65.63		71.88	81.25	68.75
mfeat-fourier	79.85	79.90		78.20	77.30	79.35
mfeat-karhunen	92.75	93.70		92.10	92.10	90.65
mfeat-morphological	70.20	68.95	68.10	68.80	67.95	70.00
mfeat-pixel	94.75					
molecular-biology_promoters	33.02	29.25		32.08	32.08	33.96
mushroom	99.74	100.00		100.00	100.00	100.00
page-blocks	95.47	95.94		96.44	95.43	96.33
pendigits	94.78	96.97	95.82		92.79	96.91
postoperative-patient-data	64.44	71.11	71.11	71.11	71.11	65.56
segment	95.32	95.37		94.16	91.56	94.59

Table A.4: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing

(max. parent : 3)

Dataset	**NpFLM**	**Bayes**	**AIC**	**BDEU**	**MDL**	**Entropy**
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	93.33
sonar	77.88	77.88		79.33	77.88	75.00
spect_test	70.05	67.38	63.64	66.31	69.52	64.71
spect_train	68.75	62.50	63.75	70.00	73.75	60.00
splice	95.17	96.58	94.80	95.89	96.24	52.57
sponge	94.74	93.42	93.42	94.74	92.11	92.11
tae	47.02	47.02	47.02	35.10	41.72	47.02
tic-tac-toe	75.89	74.74	86.01	68.89	68.58	84.34
trains	60.00	60.00	50.00	20.00	50.00	80.00
waveform-5000	82.60	81.72	81.62	80.22	80.98	79.68
zoo	98.02	98.02	95.05	97.03	97.03	97.03
	win	**20**	**29**	**25**	**30**	**28**
	neutral	**8**	**10**	**7**	**7**	**9**
	lose	**21**	**9**	**16**	**11**	**11**
Average	**78.67**	**78.23**	**76.56**	**76.49**	**76.36**	**73.72**

Table A.5: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing (max. parent : 2)

Dataset	NpFLM	Bayes	AIC	BDeu	MDL	Entropy
audiology	78.76	67.26	66.81	70.35	70.35	61.95
autos	80.98	79.02	71.71	69.76	80.49	66.83
balance-scale	72.64	71.84	72.32	71.84	72.80	71.84
breast-cancer	70.98	73.78	71.68	71.68	67.48	68.88
breast-w	96.71	96.71	96.85	96.85	96.85	97.00
bridges_version1	65.71	70.48	68.57	68.57	41.90	60.00
bridges_version2	63.81	69.52	63.81	64.76	41.90	61.90
car	91.61	93.87	93.81	93.00	93.92	85.47
colic	82.34	84.51	82.88	83.42	79.89	84.78
colic.ORIG	78.53	82.07	79.89	81.79	66.30	81.52
credit-a	85.94	86.09	85.07	85.36	85.22	86.52
credit-g	74.60	72.40	73.20	73.60	73.30	73.70
cylinder-bands	77.41	73.89	68.89	72.22	64.26	64.63
dermatology	97.54	97.81	97.54	97.54	97.81	98.09
diabetes	74.74	76.04	73.83	75.52	74.48	75.91
flags	61.34	57.22	59.79	60.82	32.47	57.22
glass	71.03	71.96	70.56	70.09	70.56	70.56
haberman	73.86	72.88	72.88	72.88	73.86	72.88
hayes-roth_test	50.00	50.00	50.00	50.00	50.00	50.00
hayes-roth_train	60.61	60.61	60.61	60.61	60.61	60.61
heart-h	84.69	82.65	79.25	84.01	84.01	80.61
heart-statlog	81.48	80.74	82.22	81.11	82.22	82.22
iris	92.67	93.33	92.67	92.00	93.33	92.67
kdd_synthetic_control	98.67	99.00	97.67	97.33	42.83	97.17
labor	94.74	89.47	91.23	91.23	87.72	87.72
letter	84.53	85.81	85.45	85.12	86.14	75.87
liver-disorders	56.23	56.23	56.23	56.23	56.23	56.23
lung-cancer	68.75	65.63	71.88	71.88	75.00	81.25
mfeat-fourier	79.85	79.75	79.15	78.30	80.70	77.40
mfeat-karhunen	92.75	93.70	93.25	92.10	92.80	92.10
mfeat-morphological	70.20	68.80	68.10	68.80	70.65	67.95
mfeat-pixel	94.75					93.15
molecular-biology_promoters	33.02	20.75	27.36	32.08	33.96	32.08
mushroom	99.74	99.95	99.99	99.99	99.99	100.00
page-blocks	95.47	95.72	95.56	95.76	95.51	95.43
pendigits	94.78	95.80	95.82	95.94	95.83	92.79
postoperative-patient-data	64.44	71.11	71.11	71.11	65.56	71.11

Table A.5: NP-FiLM BBN (K2) vs. peer scoring function using Hill Climbing (max. parent : 2) (continue…)

Dataset	NpFLM	Bayes	AIC	BDeu	MDL	Entropy
segment	95.32	95.54	94.85	94.76	95.58	91.56
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	93.33
sonar	77.88	79.81	75.96	78.37	75.48	78.85
spect_test	70.05	66.84	67.38	67.38	65.24	70.05
spect_train	67.50	67.50	68.75	68.75	67.50	73.75
splice	95.17	96.52	95.55	95.89	96.24	
sponge	94.74	92.11	93.42	93.42	92.11	92.11
tae	47.02	47.02	47.02	35.10	47.02	41.72
tic-tac-toe	75.89	72.03	76.41	68.89	76.41	68.58
trains	60.00	50.00	50.00	50.00	70.00	50.00
waveform-5000	82.60	81.70	81.40	80.22	81.52	80.98
zoo	97.03	95.05	95.05	97.03	97.03	97.03
	win	**21**	**23**	**25**	**21**	**29**
	neutral	**6**	**9**	**7**	**7**	**7**
	lose	**21**	**15**	**15**	**19**	**12**
Average	**78.62**	**77.58**	**77.22**	**77.23**	**74.88**	**76.33**

Table A.6: NP-FiLM BBN (K2) with max 4 parents vs. tree classifiers & NB

Dataset	NP-FiLM	NB	BFTree	J48	J48 graft	Decision Stump
arrhythmia	70.80	62.39	67.04	64.38	69.91	55.97
audiology	78.76	73.45	73.01	77.88	77.88	46.46
autos	80.98	56.10	72.20	81.95	80.98	44.88
balance-scale	72.64	90.40	78.72	76.64	76.64	55.04
breast-cancer	70.98	71.68	67.83	75.52	75.52	68.53
breast-w	96.71	95.99	94.28	94.56	94.71	92.42
bridges_version1	65.71	67.62		57.14	60.00	57.14
bridges_version2	63.81	65.71		56.19	58.10	57.14
car	91.61	85.53	97.05	92.36	92.36	70.02
colic	82.34	77.99	81.52	85.33	85.05	81.52
colic.ORIG	78.53	67.39	66.85	66.30	66.30	76.90
credit-a	85.94	77.68	85.51	86.09	86.09	85.51
credit-g	74.60	75.40	73.30	70.50	70.70	70.00
cylinder-bands	77.41	72.22	60.00	57.78	57.78	69.26
dermatology	97.54	97.27	93.99	93.99	94.54	50.27
diabetes	74.74	76.30	73.57	73.83	73.70	71.88
flags	61.34	55.15	35.57	59.28	59.28	51.03
glass	71.03	48.60	70.09	66.82	68.22	44.86
haberman	73.86	76.14	71.90	72.88	73.20	72.88
hayes-roth_test	50.00	57.14	67.86	57.14	57.14	50.00
hayes-roth_train	60.61	74.24	82.58	80.30	80.30	43.18
heart-h	84.69	83.67	78.91	80.95	80.95	79.93
heart-statlog	81.48	83.70	75.93	76.67	77.04	72.59
iris	92.67	96.00	94.67	96.00	94.67	66.67
kdd_synthetic_control	98.67	94.67	16.67	91.67	90.83	33.33
labor	94.74	89.47	78.95	73.68	73.68	80.70
letter	84.53	64.12		87.98	88.25	7.09
liver-disorders	56.23	55.36	64.93	68.70	68.70	57.68
lung-cancer	68.75	78.13	84.38	78.13	78.13	75.00
mfeat-fourier	79.85	75.75	75.15	75.25	76.55	19.65
mfeat-karhunen	92.75	93.60	81.95	82.85	84.15	19.50
mfeat-morphological	70.20	69.45	71.75	72.05	72.10	19.95
mfeat-pixel	94.75	93.30		78.65	80.10	19.50
molecular-biology_promoters	33.02	28.30	32.08	21.70	20.75	41.51
mushroom	99.74	95.83	99.94	100.00	100.00	88.68
page-blocks	95.47	90.85	96.55	96.88	96.88	93.13
pendigits	94.78	85.75	96.16	96.56	96.61	20.38
postoperative-patient-data	64.44	66.67	68.89	70.00	70.00	70.00
segment	95.32	80.22	95.93	96.93	96.97	28.57
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	86.67

Table A.6: NP-FiLM BBN (K2) with max 4 parents vs. tree classifiers & NB (continue…)

Dataset	NP-FiLM	NB	BFTree	J48	J48 graft	Decision Stump
sonar	77.88	67.79	71.63	71.15	72.60	73.08
spect_test	70.05	68.98	73.26	65.24	65.24	66.31
spect_train	68.75	71.25	68.75	71.25	71.25	72.50
splice	95.17	95.30		94.08	93.92	62.38
sponge	94.74	92.11	92.11	92.11	92.11	92.11
tae	47.02	54.30	57.62	59.60	60.26	37.75
tic-tac-toe	75.89	69.62	93.74	84.55	85.28	69.94
trains	60.00	70.00	30.00	90.00	90.00	60.00
waveform-5000	82.60	80.00	75.60	75.08	75.92	56.76
zoo	97.03	95.05	92.08	92.08	92.08	60.40
	win	**31**	**27**	**27**	**27**	**43**
	neutral	**1**	**2**	**1**	**2**	**2**
	lose	**18**	**16**	**22**	**21**	**5**
Average	**78.49**	**76.14**	**74.97**	**77.60**	**77.93**	**58.93**

Table A.7 (continue): NP-FiLM BBN (K2) with max 4 parents vs. tree classifiers & NB

Dataset	LAD Tree	Random Tree	Simple Cart	REP Tree	Random Forest	AdaBoost Decision Stump
arrhythmia	70.58	51.11	70.80	69.47	67.04	55.53
audiology	74.78	65.49	73.01	72.57	76.99	46.46
autos	64.88	76.59	74.63	63.41	83.41	44.88
balance-scale	84.48	77.28	79.04	77.28	80.48	72.32
breast-cancer	70.63	66.78	69.23	70.63	69.23	70.28
breast-w	95.57	94.56	94.85	93.85	96.14	94.85
bridges_version1	67.62	52.38		41.90	55.24	57.14
bridges_version2	65.71	43.81		41.90	48.57	57.14
car	90.68	83.16	97.11	87.67	92.65	70.02
colic	82.61	79.08	86.14	84.78	86.14	81.25
colic.ORIG	83.70	69.29	66.58	67.66	68.48	82.88
credit-a	86.23	79.13	85.22	85.65	85.22	84.64
credit-g	70.80	67.10	73.90	71.80	72.50	69.50
cylinder-bands	68.15	63.52	59.81	59.07	73.15	72.59
dermatology	95.63	87.43	93.99	91.53	94.81	50.27
diabetes	74.09	68.10	75.13	75.26	73.83	74.35
flags	58.76	42.78	35.57	35.57	58.25	51.03
glass	65.89	70.09	70.56	66.36	72.90	44.86
haberman	69.61	65.36	74.51	73.53	66.67	73.53
hayes-roth_test	64.29	57.14	64.29	57.14	42.86	60.71
hayes-roth_train	84.85	80.30	82.58	84.85	81.06	43.18
heart-h	78.57	76.87	78.57	77.55	77.89	77.89
heart-statlog	78.15	76.30	78.52	76.67	78.15	80.00
iris	94.00	92.00	95.33	94.00	95.33	95.33
kdd_synthetic_control	86.00	74.00	16.67	16.67	94.00	33.33
labor	84.21	78.95	78.95	78.95	87.72	87.72
letter	51.00	86.16	87.11	84.14	94.71	7.09
liver-disorders	65.51	67.83	67.54	64.06	68.99	66.09
lung-cancer	71.88	75.00	87.50	78.13	71.88	78.13
mfeat-fourier	74.20	65.30	75.50	75.40	80.95	19.65
mfeat-karhunen	77.10	72.05	81.85	78.70	92.00	19.50
mfeat-morphological	71.30	64.15	72.85	71.65	69.20	19.95
mfeat-pixel	83.10	65.35	85.65	74.25	90.50	19.50
molecular-biology_promoters	28.30	29.25	32.08	32.08	33.02	41.51
mushroom	99.90	100.00	99.94	99.96	100.00	96.20
page-blocks	96.05	96.18	96.78	96.88	97.22	93.13
pendigits	81.35	95.72	96.32	95.55	98.82	20.38
postoperative-patient-data	68.89	61.11	71.11	70.00	63.33	70.00

Table A.7 (continue): NP-FiLM BBN (K2) with max 4 parents vs. tree classifiers & NB

Dataset	LAD Tree	Random Tree	Simple Cart	REP Tree	Random Forest	AdaBoost Decision Stump
segment	92.21	95.80	96.15	95.06	97.66	28.57
shuttle-landing-control	93.33	93.33	93.33	93.33	93.33	80.00
sonar	78.85	73.56	71.15	75.48	80.77	71.63
spect_test	67.91	58.29	73.26	69.52	59.89	70.59
spect_train	70.00	58.75	68.75	70.00	65.00	70.00
splice	94.95	68.37	52.57	52.51		86.74
sponge	88.16	94.74	92.11	92.11	93.42	92.11
tae	59.60	68.21	54.97	53.64	66.23	37.75
tic-tac-toe	74.63	79.23	92.90	82.15	92.90	72.55
trains	70.00	50.00	30.00	0.00	50.00	50.00
waveform-5000	79.40	72.44	76.68	76.90	81.80	66.64
zoo	95.05	95.05	92.08	90.10	95.05	60.40
	30	**36**	**25**	**33**	**30**	**41**
	1	**2**	**3**	**1**	**2**	**0**
	19	**12**	**20**	**16**	**18**	**9**
Average	**76.86**	**72.49**	**75.48**	**71.75**	**77.86**	**61.40**

Table A.8: NP-FiLM BBN (K2) vs. function classifiers & NB

Dataset	NP-FiLM	Logistic	Multilayer Perceptron	RBF Network
arrhythmia	70.80	54.87	67.70	59.96
audiology	78.76	79.20	83.19	70.35
autos	80.98	71.22	80.00	61.95
balance-scale	72.64	89.60	90.72	87.20
breast-cancer	70.98	68.88	64.69	70.98
breast-w	96.71	96.57	95.28	95.85
bridges_version1	65.71	60.00	69.52	51.43
bridges_version2	63.81	59.05	71.43	59.05
car	91.61	93.11	99.54	88.25
colic	82.34	80.98	80.43	80.43
colic.ORIG	78.53	69.02	64.40	76.90
credit-a	85.94	85.22	84.20	79.71
credit-g	74.60	75.20	71.60	74.00
cylinder-bands	77.41	78.70		71.30
dermatology	97.54	96.99	96.17	96.17
diabetes	74.74	77.21	75.39	75.39
flags	61.34	43.30	64.43	54.64
glass	71.03	64.02	67.76	63.55
haberman	73.86	74.18	69.28	73.86
hayes-roth_test	50.00	53.57	39.29	50.00
hayes-roth_train	60.61	54.55	69.70	67.42
heart-h	84.69	84.69	85.03	85.03
heart-statlog	81.48	83.70	78.15	84.07
iris	92.67	96.00	97.33	95.33
kdd_synthetic_control	98.67	85.00		99.33
labor	94.74	92.98	85.96	94.74
letter	84.53	77.30	82.08	
liver-disorders	56.23	68.12	71.59	64.35
lung-cancer	68.75	81.25	65.63	81.25
mfeat-fourier	79.85	73.30	83.65	79.65
mfeat-karhunen	92.75	89.40	95.85	95.30
mfeat-morphological	70.20	73.90	74.95	70.55
mfeat-pixel	94.75			94.30
molecular-biology_promoters	33.02	28.30	26.42	33.02
mushroom	99.74	100.00	100.00	98.51
page-blocks	95.47	96.46	96.22	94.83
pendigits	94.78	95.55	94.69	95.20
postoperative-patient-data	64.44	60.00	55.56	56.67

Table A.8: NP-FiLM BBN (K2) vs. function classifiers & NB (continue…)

Dataset	**NP-FiLM**	**Logistic**	**Multilayer Perceptron**	**RBF Network**
segment	95.32	95.80	96.06	87.19
shuttle-landing-control	93.33	93.33	93.33	73.33
sonar	77.88	73.08	82.21	72.12
spect_test	70.05	65.24	56.68	68.98
spect_train	68.75	66.25	63.75	66.25
splice	95.17	91.03		94.36
sponge	94.74	97.37	94.74	93.42
tae	47.02	54.30	54.30	52.98
tic-tac-toe	75.89	98.33	97.39	69.62
trains	60.00	80.00	70.00	60.00
waveform-5000	82.60	86.60	83.56	85.14
zoo	97.03	96.04	95.05	96.04
	win	**25**	**22**	**29**
	neutral	**2**	**2**	**6**
	lose	**22**	**22**	**14**
Average	**78.49**	**77.73**	**77.93**	**76.53**