# Parameter Free and Non Penalized Scoring Metric for Bayesian Belief Network

**Muhammad Naeem*, Sohail Asghar****

**Department of Computer Science, Mohammad Ali Jinnah University Islamabad, 44000 Pakistan (e-mail: naeems.naeem@gmail.com)*
***University Institute of Information Technology, PMAS-Arid Agriculture University, Rawalpindi Pakistan, (e-mail: sohail.asg@gmail.com)*

**Abstract:** This study has introduced parameter free, decomposable with no penalty factor and an efficient saleable scoring metric, the Non Parametric Factorized Likelihood Metric (NP-FiLM) useful for structure learning. The proposed score 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. An empirical evaluation of the proposed metric has been carried out over an abundant number of natural datasets obtained from UCI machine learning repository. The comparison is made with respect to eleven tree classifiers, one regression model and two neural network system. Furthermore, the scoring metric has been examined to six peer scoring metrics within the greedy search mechanism. NP-FiLM oriented Bayesian Belief Network has been satisfactory found with significant results in a paradigm of accuracy and classification error. The introduced scoring function is capable of illustrating the best possible data fitting in the context of hyper-parameters described above.

*Keywords:* Bayesian Network, Classifier, Data Mining, Structure Learning, Scoring Functions, K2

## 1. INTRODUCTION

The evolution of ensemble or single classifier which involves Bayesian Belief Network (BBN) or its variant is a rising research interest in numerous domain of real world application. In classification, structure prediction from Bayesian inference model is a highly symbolic formalism for the purpose of retrieving hidden rules in pragmatic situations. This process consists of two 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 first part. In general a thick network deems to represent an optimized fitted model. This study has introduced parameter free, decomposable with no penalty factor, an efficient scale-able scoring metric, the Non Parametric Factorized Likelihood Metric (NP-FiLM) useful for structure learning. The proposed scoring metric can deliver equally or better results using thin network as a result of which the complexity of the model is significantly extenuated without reducing the classification accuracy.

## 2. BAYESIAN NETWORK

This article is forwarded with the introduction of the theory and definition around structure learning explained briefly to the shrewd readers of this study. This section carries out some myriad terms of BBN in the context of structure learning. A BBN which is also known in alternate names of Belief Network is a graphical model representing a process of an arbitrary nature.

- It can be described by a triplet <*D, G, R*>.
- The first component of this triplet denotes the underlying dataset.
- The second component indicates a graph
- The last component is set of parameters representing the underlying network.
- The second component *G* belongs to the family of Directed Acyclic Graphs (DAG).
- Each node in the 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 node 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.

The number of graphs in structure learning is not limited to a single graph during the searching process; therefore it is useful considering 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 formula given in equation 1.

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

The set of data which is to be learnt can be formally described as $O = \{o_1, ....., o_n\}$ where $o_i = \{z_i^1, z_i^2 ..... z_i^N\}$ Note down that subscript points out the number of observations and the superscript is the indication of the 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. It has been 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 expressed by $Z_j^i$ indicating the counts of ith variables with jth states. Each structure $g \in G$ of the Bayesian Network can be denoted by N sets of parents $\prod^1, ... \prod^N$. In simple words, it can be stated that for each node $j = 1, ...N$ the set $\prod^j$ is a set of parent nodes in which a node has no self loop or close loop. Formally it can be represented such that $\prod^j \subseteq \{Z^1, .... Z^N\} \setminus \{Z^j\}$.

## 3. SCORING FUNCTION

A BBN classifier is technically composed of two components which include a scoring function or scoring metric and a searching heuristic; the way through which a scoring metric is evaluated. (Jensen et al., 2007) pointed out two essential characteristics for any scoring metric. The first characteristic is the capability of any scoring metric to balance the accuracy of a structure versus structural complexity. The second characteristic is its computational tractability. We shall revive currently existing scoring metrics as below:

- Minimum Description Length (MDL) fulfils these characteristics (Lam et al., 1994). MDL is usually suited to complex Bayesian network. Mathematical formulation of MDL is comprised of explanation of Log Likelihood (LL) as described below:

$$LL(B \mid T) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left( \frac{N_{ijk}}{N_{ij}} \right) \quad (2)$$

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

$$MDL(B \mid T) = LL(B \mid T) - (1/2) \log(N) \mid B \mid \quad (3)$$

|B| denotes the length of network. It is frequentistic enumeration of distinct states of a given query feature and its corresponding parent's state combination as described below:

$$\mid B \mid = \sum_{i=1}^{n} (r_i - 1) q_i \quad (4)$$

- Akaike Information Criterion (AIC) (Akaike, 1974) originally is described mathematically:

$$AIC = -2 \times \ln(likelihood) + 2 \times K \quad (5)$$

The value of K indicates the count of parameters in the given model. However in BBN, its mathematical formulation has been transformed into

$$AIC(B \mid T) = LL(B \mid T) - \mid B \mid \quad (6)$$

- BDeu (Buntine, 1999) is another scoring measure relying only on equivalent sample size of (N´. Carvalho et al., 2011) has provided and discussed its decomposition as below in mathematical form:

$$BDeu\ (B,T) = \log(\ P(B)) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \Delta$$

$$\Delta = \left( \log \left( \frac{\Gamma\left(\frac{N'}{q_i}\right)}{\Gamma\left(N_{ij} + \frac{N'}{q_i}\right)} \right) + \sum_{k=1}^{r_i} \log \left( \frac{\Gamma\left(N_{ijk} + \frac{N'}{r_i q_i}\right)}{\Gamma\left(\frac{N'}{r_i q_i}\right)} \right) \right) \quad (7)$$

- (Cooper et al., 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 us consider that there is a sequence of *n* instances such that $z^n = d_1 d_2 d_3 ..... d_n$ the Bayes scoring function of structure $g \in G$ can be formulated in form of the equation.

$$P_b(g, z^n) = P_b(g). \coprod_{j \in J} (\Delta) \quad (8)$$

$$\Delta = \prod_{s \in S(j,g)} \frac{(\alpha^j - 1)! \cdot \prod_{q \in A}^{j} n[a, s, j, g]!)}{(n[s.j.g] + \alpha^j - 1)!} \quad (9)$$

Where $P_b = (g)$ is 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 enumeration of the variable of the network g, and $s \in S(j,g)$ is the counting of the set of 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.

- (Carvalho et al., 2011) introduced factorized conditional log likelihood (fCLL) and empirically proved it to be reasonable among other established scores. Its

decomposability over the network structure is defined as below.

$$\hat{f}\,CLL(G \mid D) = (\alpha+\beta)\hat{LL}(B \mid D) - \beta\lambda \sum_{i=1}^{n}\sum_{j=1}^{q_i^*}\sum_{k=1}^{r_i}\sum_{c=0}^{1} N_{ijkc}\left(\log\left(\frac{N_{ijkc}}{N_{ij^*c}}\right) - \log\left(\frac{N_{ijc}}{N_{ij^*}}\right)\right)$$

$$-\beta\lambda\sum_{c=0}^{1} N_c \log\left(\frac{N_c}{N}\right) - \beta N \rho \tag{10}$$

## 4. TOWARDS NP-FILM

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, the score for each node is Ψi. The cumulative scoring criteria Ψ can be expressed formally:

$$\Psi(B, D) = \sum_{i=1}^{n} \Psi_i(\Pi_{fi}, D) \tag{11}$$

The scoring function in general is 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 12.

$$LL(G \mid D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) \tag{12}$$

Where $N_{ijk}$ indicates that i$^{th}$ feature is instantiated with k$^{th}$ state along with the j$^{th}$ state of q$^{th}$ parent of i$^{th}$ feature. See the equation 12 that LL is a very simple and easy to calculate, hence a decompose-able 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. A proposition can be formulated here as below.

**Proposition 3.1.** Let Μ denotes a Bayesian network over the query variables X. Moreover, it is also assumed that Bayesian network parameters $\Phi_M$ are locally and globally independent. Then the size of the model is a function of number of links L and distinct states Ṣ such that

$$\text{Size}(\text{Μ}) = f(\text{L}, \dot{\text{S}}) \tag{13}$$

A simple decomposition will result in

$$\text{Size}(\text{Μ}) = \sum_{x \in X}(|\, pa(x)\,| \times |\, x\,|) \tag{14}$$

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

**Lemma 3.1.** Let Μ be a Bayesian network being represented by the set of query variables X. The optimized and most representative model $\acute{M}_X$ 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 Μ be any ordinary model which denotes parameter distribution say $P_{UX}$. On the other hand, $\acute{M}_X$ 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 Μ, they must be present in $\acute{M}_X$. However, if there is a situation where the size of Μ is smaller than size of $\acute{M}_X$ then it is so because some links in Μ 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 over-fitting 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 carrying penalization can be generally expressed in a following non decomposable notation.

$$SF(G \mid D) = LL(D \mid G) - \sum_{i=1}^{n} PF(X_i, G, D) \tag{15}$$

Several well known scoring functions which have been discussed in previous sections 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 et al., 2012). However, this issue has been investigated 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 et al., 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. It is 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 distinct states given C contains b number of unique states.

$$F = \{f_i \mid i = 1, \ldots a\} \tag{16}$$

$$C = \{c_j \mid j = 1, \ldots b\} \tag{17}$$

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 joint probability state between both of these feature variables can be described as:

$$\delta(C,F)=\sum_{c}\sum_{f}P(C=c,F=f) \tag{18}$$

The above joint probability is not normalized; which means it is not calibrated between the value of 0 and 1 in order to compare with other pair wise values. Such probability distribution is termed as *potential*. The *potential* $\xi$ can be described formally as:

$$\xi_{C,F}=\sum_{c}\sum_{f}P(C=c,F=f) \tag{19}$$

Our aim here is to maximize the discriminant objective function out of this potential. A change in this *potential* can be incurred such that

$$\xi_{C,F}=\sum_{c}\left[\max\arg\left(\sum_{f}P(C=c,F=f)\right)\right] \tag{20}$$

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.

**Lemma 3.2.** The discriminant joint probabilities obtained from the *potential* in the equation 20 may lead to turn into maximum a prior probabilistic inference for a simple point estimation case in structure learning.

**Proof.** It begins with re writing the equation 20 such that

$$\xi_{C,F}=\sum_{c}\left[\max\arg\left(\sum_{f}P(C_c,F_f)\right)\right] \tag{21}$$

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

$$\lambda_{C,F}=\sum_{c}\left[\max\arg\left(\sum_{f}\frac{P(c,f)}{\vartheta(F)}\right)\right] \tag{22}$$

The simple point estimation potential (see equation 21) is decomposed into conditional joint probability factor. However, this study is not dealing in ordinary cases of single input features. It must be required to generalize it to a dataset with more than one non class features.

**Lemma 3.3.** NP-FiLM is a decomposable scoring function.

**Proof.** While generalizing NP-FiLM, there are n number of non class feature variables and a single class variable within the dataset D. It can be expressed easily to reduce this simple point estimation into a generalized maximum a posterior inference notation as below:

$$NpLFM(D,G)=\sum_{i=1}^{n}\max\arg(X_i,Pa(X_i),C,D) \tag{23}$$

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 equation 22. 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 the factor of class variable is included, a minor change will be developed into $\sum_{j=1}^{qi}\sum_{k=1}^{r1}N_{ijck}$

Plugging this value into equation 23, it can be expressed as

$$NpLFM(D,G)=\sum_{c}\left[\frac{1}{|C_c|}\max\arg\left(\sum_{j=1}^{qi}\sum_{k=1}^{r1}N_{ijck}\right)\right] \tag{24}$$

If a link is introduced between node $X_i$ and node $X_j$ pointed towards $X_j$, then only the local value of NP-FiLM will be altered for the purpose of evaluating whether this addition gives any significant improvement in the structure being represented by G such that.

$$\Delta(X_pX_q)=\left\langle\sum_{i=1}^{n}\max\arg(X_q,Pa(X_q),C,D)\cup\{X_p\}\right\rangle -\sum_{i=1}^{n}\max\arg(X_q,Pa(X_q),C,D) \tag{25}$$

Equation 24 and 25 indicates that simple frequency calculation of the feature node, non class node and class node can result into a numerical value. Hence it can be concluded that NP-FiLM 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.

Whilst reviving 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 node $X_i$ 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 begets some critical observation. Let us consider feature set and class variable as defined in equation 16 and 17. Let us 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 of 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. It has been already 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.

**Proposition 3.2.** 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 26.

$$\omega_F = \lambda_{C,F} - \lambda_{C,F} \tag{26}$$

The terms $\lambda_{C,F}$ and $\lambda_{C,F}$ plays the role of existence restrictions. Let us consider both of them as existence restrictions such that $(F,C) \in \lambda_{C,F}$: the link $F \to C$ explains the discriminant objective with respect to the class such as: $(F,C) \in \lambda_{F,C}$: the link $C \to F$ means the discriminant score with respect to the feature. In our earlier research (Naeem et al., 2013) the correct topological ordering between two features was highlighted. This was shown by an earlier version of the proposed scoring function Integration to Segregation (I2S) in which it was emphasized 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 neighbour and parents becoming unreasonable. However I2S is capable of rightly identifying it in majority of the cases as compared to BIC, MDL, BDeu, Entropy and many more. Moreover, (Madden, 2009) 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, 2009). 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 27.

$$\lambda_{F,C} = \sum_f \left[ \max \arg \left( \sum_c \frac{P(f,c)}{\vartheta(C)} \right) \right] \tag{27}$$

See equation 22 and 23 for detail of equation 25. A function for simple descending order is applied to the weights achieved from the equation 24 which results into an ordered list of input variables.

$$\overleftarrow{F} = \{\overset{i}{\omega}_f \mid i = 1...n\} \tag{28}$$

Plugging this ordered set into the equation 24 will give result in

$$NpLFM(D,G,\overleftarrow{X}) = \sum_c \left[ \frac{1}{|C_c|} \max \arg \left( \sum_{j=1}^{qi} \sum_{k=1}^{r1} N_{ijck} \right) \right] \tag{29}$$

**Lemma 3.4.** The ordered set initialized by an intelligent heuristic may convert NP-FiLM into a well behaved scoring metric.

**Proof.** Let us consider a set of n un-sorted features $F = \{f_1, f_2, f_3, ...f_n\}$. Let us begin from any of the succeeding feature say jth feature $f_j$ such that it lies somewhere in the trivial un-ordered list denoted by $\{\rho, f_j, \varsigma\} \therefore \rho \cap \varsigma = \phi$ where $\rho$ is the set of predecessor and $\varsigma$ is the set of successor nodes. It is already stated 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 search algorithm can choose any of the parent-set before $f_j$. If a feature $f_s$ exists such that it can significantly contribute towards score of structure, then following expression must hold $\forall(f_s,c) \in \rho, f_s \to f_j : true$ and otherwise the expression holds $\forall(f_s,c) \in \varsigma, f_s \to f_j : false$.

A careful consideration of expression of NP-FiLM (see equation 29), one 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.

NP-FiLM 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 influences 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, 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 specific distributions only. Secondly, they 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, 2012) 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.

## 5. EMPIRICAL VALIDATION OF NP-FiLM

A number of benchmark datasets have been used for evaluation purpose in this study. These include dataset with binary classification problems as well as multivariate classification problems obtained from the UCI data repository (Frank et al., 2010). These dataset are processed into *weka support format (arff) available at sears project ((Hall et al., 2009; Sears, 2013). These data sets were randomly selected so as to choose from various real-world domains with varying characteristics. The random sampling of the dataset result into diversified attributes count, number of rows (cases) and classes. It is preferred selecting dataset with

variety of information under these categories to avoid any prior bias factor in favour 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. 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:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{30}$$

The results are illustrated showing eleven tree classifiers, three function based classifiers and six well known scoring function based NB system. The scoring function comparison has been made using various parent values of 4, 3 and 2 (polytree) within the searching algorithm of K2.

In all of these experiments, ten fold cross validation was exercised, which 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 the figures in this section. Firstly the proposed measure is compared with every other classifier in terms of average (see figure 1,3 and 5) and in term of win/neutral/lose that the proposed scoring function based classifier wins or loses from the specific algorithm. Where as in the neutral case, no significant statistical difference was found. That is, any other classifier exhibited statistically better than the proposed technique according to corrected t-test with $p < 0.05$ (Nadeau et al., 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 et al., 2003).

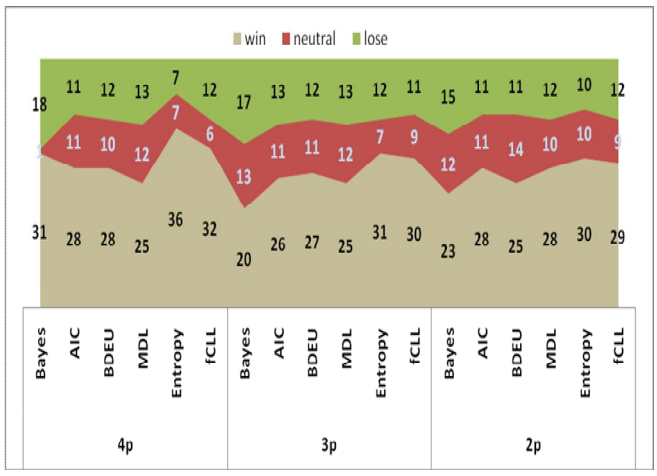


Fig. 1. Comparison of accuracy of NP-FiLM vs. other classifiers over 50 dataset (win/neutral/lose).

Although, the results of NP-FiLM with respect to other classifiers have been obtained which were either tree based classifiers or neural network based learner and one regression model. However, it is preferable to draw results while keeping maximum number of external and internal parameters quite same. This includes the same searching algorithm, the same number of potential candidates for parental node, estimation of frequencies, the pre-processing steps such as deciding what to do with missing samples and discretization of continuous data. Let us keep 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 influences 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 poses 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. In this study, three sets of experiments have been performed to validate the effectiveness of our NP-FiLM. In the 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. The setting of markove blanket is set to false, initNaiveBayes value set to true and random order set to false.

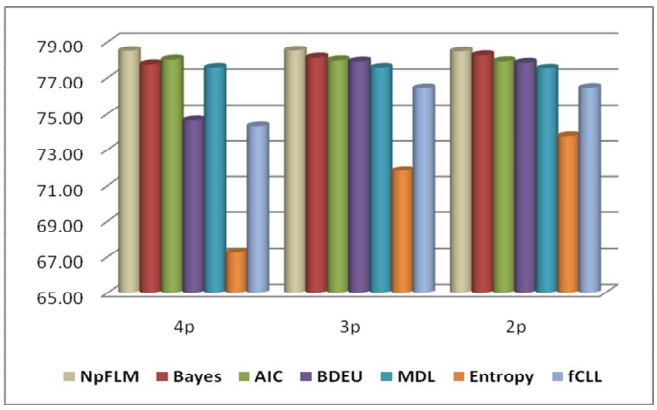


Fig. 2. Average accuracy of classifiers over 50 dataset.

This session of experiment was repeated by keeping parent set value of three and two respectively as indicated by the figure 1 and figure 2. A careful examination of these figures indicates that the best performance of the NP-FiLM was obtained in the relatively dense graph. Secondly an important observation was noticed that in all of three cases, the proposed scoring function exhibited almost same accuracy (see figure 2). It means the proposed scoring function usually generates a polytree whatever the value of the maximum number of parent is set to (greater 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 and unless another node with best characteristics is not found, it is not conditioned with

that specific node. A recently introduced scoring function fCLL by (Carvalho et al., 2011) has also been included. 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 this study. The scoring function fCLL was evolved in the background of improvement in TAN, however, its functionality was exploited in context of general Bayesian network with maximum parent set of four. When the results were examined from the perspective of average accuracy, again NP-FiLM outperforms the other scoring function.

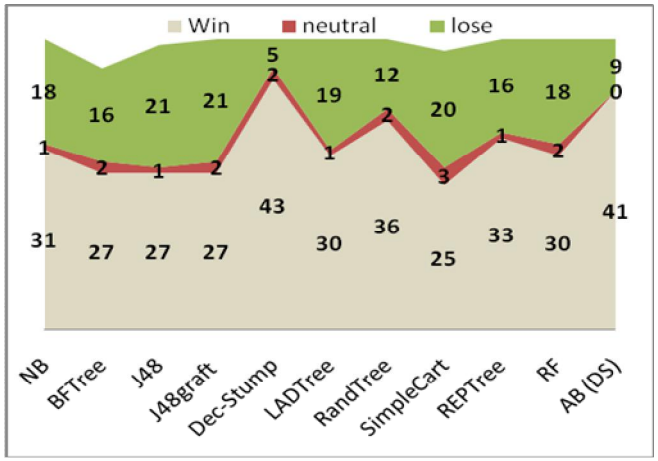


Fig. 3. Comparison of accuracy of NP-FiLM vs. other classifiers over 50 dataset.

So far, the results have been examined by comparing NP-FiLM with respect to its peer scoring metrics, however it is quite essential to examine its performance to other classifiers such as tree based classifiers or neural network based learner and regression model while keeping all of the default parameter values for these classifiers (usually fixed with optimized setting). Figure 3 and 4 represent the results of tree based classifiers including Simple NB, Breadth First Tree (BFTree), J48 (implementation of C4.5 in weka), J48graft, Decision Stump, LAD Tree, Radom Tree, Simple Cart, Random Forest (RF) and Decision Stump with Adaboost ensembler (AB(DS)) (see Pang-Ning, 2006)) for detail). In comparative results, NP-FiLM gives best result in 14 dataset followed by RF for which RF delivers best result for 9 datasets. 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 deliver highest result for data set 'trains' and 'mushroom'.

Some dataset were too large in number of features that a few of the classifiers did not give result in reasonable time, thus the results of these dataset have been excluded from average performance comparison. On the other hand, when the results are observed from different perspective of win/neutral/lose, then it is the comparison of NP-FiLM with respect to other tree classifier. In these comparisons, one can observe that Decision Stump and Adaboost Decision Stump both exhibit poor results in comparison to NP-FiLM while J48 and its modified version J48graft were in close competition; albeit NP-FiLM outperforms all of these eleven tree classifiers. The second dimension of comparison is achieving average accuracy over set of all 50 datasets.

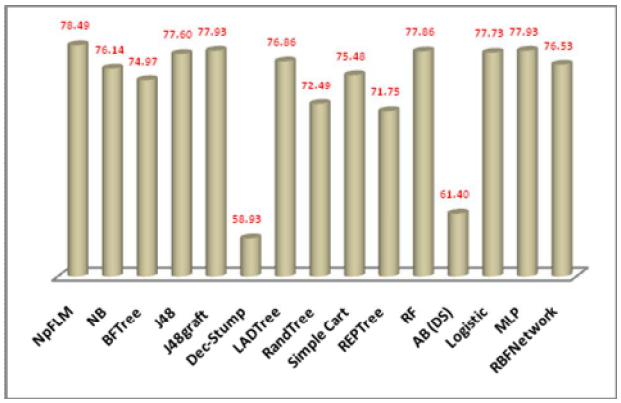


Fig. 4. Average accuracy of classifiers over 50 dataset.

It is noteworthy that while calculating average, missing cells were omitted for comparison on equalitarian basis. It is evident from figure 4 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%. Our finding reveals that on the overall, the tree classifiers are comparatively well suited for 'thin network', where 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. Explaining it by an example, the dataset arrhythmia and audiology contains 280 and 70 features respectively, the class size is also 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 indicate the complexity of the size. (See proposition 1 for detail). The dataset in which the performance of tree classifiers is relatively better posses very simple structure (thin network) such as in case of balance-scale, hayes-roth_test or some others.

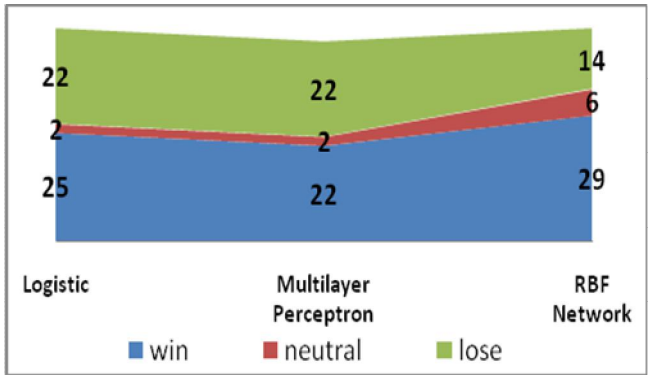


Fig. 5. Accuracy of NP-FiLM vs. other classifiers over 50 dataset.

Figure 5 is another comparison of NP-FiLM towards function classifiers which include Logistic (a regression model), Multilayer Perceptron (MLP) and Radial Basis Function (RBF) Network. These classifiers in general have high time

complexity as compared to their peer classification system. Specially, Multilayer perceptron consumes exceptionally outstanding time resulting into excluding some larger dataset cylinder-bands, kdd_synthetic_control, mfeat-pixel and splice. It can be observed that although Multilayer Perceptron delivers some comparable results to NP-FiLM in which NP-FiLM wins over 22 dataset and also lose over other 22 dataset. However, the time complexity of our technique 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%.

**Table 1. NP-FiLM vs. 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** |
| **Absolute Win** | **1** | **8** | **5** | **3** | **11** |

When it came to question of number of parents for a non class node in Naïve Bayesian networks, 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., 1997) 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). 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 among three of these type of network and deliver some 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 variants of GBN have been proposed preferably suitable in various domains of interest while the other two networks were usually 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.

(Madden, 2009) challenged some existing challenges 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 shown in the column next to the dataset in table 1. 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 the nodes are annotated with the conditional mutual information between all pairs of non class variables but conditioned on the class node, as shown by the equation.

$$I(x_1, x_2 \mid c) = \sum_{x_1, x_2, c} \left[ P(X_1, X_2, C) \log \frac{P(X_1, X_2 \mid C)}{P(X_1 \mid C) P(X_2 \mid C)} \right] \tag{31}$$

(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 experimental analysis reveal out that the poor performance which was earlier reported by (Friedman et al., 1997) about GBN has its roots in simple empirical frequencies (parameter smoothening) in order to estimate General Bayes Network parameters. It may be concluded that parameter smoothing plays important role in improvement of a classifier. It can be pointed out that GBN has much more potential to be considered for any specific domain because of its diverse nature in drawing structure. 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 was employed. Moreover, missing attributes were ignored in both of the studies. Table 1 illustrates that NP-FiLM delivers better results as compared to others as it outperformed in 11 datasets out of 26 datasets. Moreover, the average of the classification accuracy was also highest towards NP-FiLM.

**Table 2. NP-FiLM vs. Kabir et al., (2011).**

| Dataset | ECNBDMM-I | ECNBDMM-II | 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 comparions of NP-FiLM to a technique forwarded by Kabir et al., (2011) is described. (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 is 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 arguable; albeit authors produce a criteria of weighted training error such that.

$$training\_error = \sum_{i=1}^{k}\left( cluster\_error\_of(Ci) \times \frac{n_i}{N} \right) \quad (32)$$

Where $C = \{c_1, c_2, c_3, ...c_n\}$ 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 increases it gradually till it reaches a specific stop threshold. The stop threshold is marked by continuous increase of weighted training error after a few observations. This generates an optimal value of K.

The published result presented by (Kabir et al., 2011) shows that ECNBDMM-II is somewhat efficient with good accuracy on its best dataset; albeit the dataset is quite limited in size (only four datasets in all) raising an argument in generalize their techniques (See table 2). One dataset Iris is particularly a short data and the number of states in each of its features is below medium in size. It can be suggested that their technique might be well suited for thin networks. However, there are many issues arguable in models ECNBDMM-I and ECNBDMM-II. In this technique, training data set is limited enough to build a "correctly represented" model in each run. Although during clustering whole of actual dataset, there are n numbers of clusters; but only a single cluster is used for training model whereas the test data is assumed to be fixed. It means n numbers 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 biases in dataset.

## 6. CONCLUSION

In classification, structure learning and inference from Bayesian model is a well renowned practice for the purpose of mining hidden but useful information out of large amount of data. Generally, this process comprised of two phases. First phase addresses the construction of best representative structure from the dataset. The second phase conducts the inference using this learnt structure. This study has tweaked out the first phase. The central crux in the designing first phase of a BBN classifier is to bring out a discriminant metric functioning within vector space of query variables through exercising of a prior knowledge. The effectiveness of the Bayesian belief network using K2 greedy heuristics searching mechanism has qualified its splendid place in the field of classification systems. Analysis were made for numerous well established scoring metrics including Bayes, BDeu, MDL, AIC, Entropy, and fCLL in the perspective of balance between over-fitting and under-fitting. This study presented a novel parameter free, penalty-less and decomposeable metric in the domain of structure learning. The introduced measure, known as Non Parametric Factorized Likelihood Metric (NP-FiLM) is characterized by the mutual dependence approximated by maximizing marginal and joint probability distribution. The novel metric is especially designed for discriminative learning because it is decomposable with the potential to forward efficient estimation of structure learning. The accuracy merit of NP-FiLM is empirically evaluated and compared to six well known state of-the-art scoring metrics given a reasonable size of benchmark data sets. Furthermore, exhaustive experimentation of comparison to numerous tree classifiers, two neural networks and one regression model were presented. The study delivered the analysis and comparison from different angle using Greedy search with various setting of parental set. NP-FiLM performed significantly better than others in all of these comparisons. The proposed measure is extrapolated to construct the realistic network which is likely to tally with the practical gist of domain experts.

## REFERENCES


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

Buntine, W. L. (1999). Theory refinement on Bayesian networks, *Proc*. UAI', 52–60.

Carvalho, A.M., Roos, M.T.T., Oliveira, A.L., Myllymäki, P. (2011). Discriminative learning of Bayesian networks via factorized conditional log-likelihood, *Journal of machine learning research*, Vol. 12, 2181-2210.

Cooper, G.F., (1992). Herskovits, E., A Bayesian Method for the induction of probabilistic networks from data, *Machine Learning*, 9, 309-347.

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

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

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.

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,.

Kabir, M.F., Rahman, C. Hossain, M.,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.

Lam, W., Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle, *Comp. Intell*., Vol. 10, 269-294.

Liu, Z. (2012). Empirical evaluation of scoring functions for Bayesian network model selection, *BMC bioinformatics*, 13(Suppl 15), S14.

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

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

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

Pang-Ning, T., Steinbach, M., Kumar, V. (2006). Introduction to data mining, *Library of Congress*.

Sears Project, http://repository.seasr.org/Datasets/UCI/arff/, accessed March 2013.