Optimizing the Topology of Bayesian Network Classifiers by Applying Conditional Entropy to Mine Causal Relationships Between Attributes

LIMIN WANG\textsuperscript{1,2}, GAOJIE WANG\textsuperscript{1}, ZHIYI DUAN\textsuperscript{1}, HUA LOU\textsuperscript{3}, AND MINGHUI SUN\textsuperscript{1}

\textsuperscript{1}College of Computer Science and Technology, Jilin University, Changchun 130012, China
\textsuperscript{2}Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
\textsuperscript{3}Department of Software and Big Data, Changzhou College of Information Technology, Changzhou 213164, China

Corresponding author: Gaojie Wang (gjwang18@mails.jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61272209 and Grant 61872164.

ABSTRACT Due to the excellent classification performance and expressivity, the study of Bayesian network classifiers (BNCs) has attracted great attention ever since the success of Naive Bayes (NB). Information theory has established mathematical basis for the rapid development of BNC. In this paper we propose the definition of entropy function $H_B(D)$, which corresponds to the optimal number of bits encoded in the network structure $B$ and can roughly measure the amount of information implicated in training data $D$. Each factor in $H_B(D)$ explicitly represents statements about causal relationships. An efficient heuristic search strategy is introduced to minimize $H_B(D)$ and explore the optimal topology of BNC. Our extensive experimental evaluation on 40 datasets reveals that this out-of-core algorithm achieves competitive classification performance compared to state-of-the-art learners such as tree augmented Naive Bayes, $k$-dependence Bayesian classifier, support vector machine, logistic regression and neural network.

INDEX TERMS Bayesian network classifier, conditional entropy, causal relationships.

I. INTRODUCTION

Machine learning recently receives tremendous attention from both statistics and computer science communities. These algorithms learn from observed data and can be used to predict. Information theory, which was first introduced and developed by Shannon [1] to explain the principle behind point-to-point communication and data storing, has been widely applied in machine learning. Many criteria, e.g., joint mutual information [2], maximum-relevance minimum-redundancy [3] and etc. [4]–[6], have been proposed based on entropy, mutual information and conditional mutual information and their variations.

Classification is one of the most active research area in both machine learning and data mining. The goal is to “train” a model based on the training data and this model can have good generalization performance on unlabeled testing instances. The success of Naive Bayes (NB) has considerably strengthened the interest and enthusiasm for learning Bayesian network classifiers (BNCs) from data. For BNC learning, information theory plays an important role and many of its applications consider information-theoretic quantities. Mutual information is commonly applied for attribute sorting or selection [8]–[10], and conditional mutual information for measuring conditional dependence between predictive attributes [11]–[15]. Researchers also proposed some alternative measures (e.g., information gain [16], chi-squared score [17] and etc. [18], [19]) that can be used to measure the relevance.

BNCs, e.g., state-of-the-art tree augmented Naive Bayes (TAN) and $k$-dependence Bayesian classifier (KDB), provide a graphical representation for the probability distribution of the training data. However, conditional mutual information cannot be used to quantitatively measure the amount of information implicated in BNCs and describe the causal relationships due to its symmetry characteristic. In this paper, we seek to explore the relationship between entropy function and joint probability distribution in terms of log likelihood. On the basis of this, we clarify the reasonableness of applying conditional entropy to identify significant causal relationships, and then propose a heuristic search approach to find an appropriate network structure that may “best fit” training data.
The contributions of this paper are as follows:

- We introduce the definition of entropy $H_B$ to measures the amount of information implicated in the topology of BNC $B$. We then prove the correctness of applying conditional entropy $H(X_i|C, \Pi_i^B)$ to represent causal relationships between attributes. The resulting highly scalable algorithm combines the computational efficiency of classical generative learning with the low bias of discriminative learning.

- We compare the performance of our algorithm, i.e., FKDB, with other state-of-the-art classifiers on 40 datasets, ranging in size from 24 to 165 thousand instances and 4 to 166 attributes. We show that FKDB achieves comparable or lower error than a range of state-of-the-art BNCs (e.g., NB, TAN, KDB and AODE) and non-BNC classifiers (e.g., support vector machine, logistic regression and neural network).

The remainder of this paper is organized as follows: Section 2 reviews some state-of-the-art BNCs and investigates the difference between NB and these BNCs in terms of log likelihood. Section 3 presents a heuristic search strategy to build a high-order maximum weighted spanning tree by applying conditional entropy rather than conditional mutual information. Section 4 presents a set of comparisons for our proposed algorithm with other approaches, including logistic regression, support vector machine and neural network. To finalize, Section 5 shows the main conclusions and future work.

II. BAYESIAN NETWORK CLASSIFIERS

The supervised classification problem consists of inducing a classification model that is able to assign a label $c \in \Omega_C$, from a set of $|C|$ labels of the class variable $C$, to an unlabeled instance $x = (x_1, \ldots, x_n)$, with values for the $n$ attributes $X = \{X_1, \ldots, X_n\}$. Because the prior probability $P(x)$ is irrelevant to classification, a BNC assigns the most probable a posteriori (MAP) class to $x$ by applying the following classification rule based on Bayes theorem,

$$
c^* = \arg \max_c P(c|x) = \arg \max_c \frac{P(x|c) P(c)}{P(x)} = \arg \max_c P(x, c).
$$

(1)

BNCs approximate the estimate of $P(x, c)$ with a factorization according to the network topology [7]. The topology is a directed acyclic graph (DAG) in which vertices correspond to the random variables $\{X_1, \ldots, X_n, C\}$ and arcs encode the probabilistic dependencies among triplets of variables. For restricted BNC $\mathcal{B}$, class variable is the root node, thus $P(x, c)$ can be decomposed as Eq. (2) shows. Each factor in $P(x, c)$, i.e., $P(x_i|c, \pi_i^B)$, is a categorical distribution, where $\pi_i^B$ denotes the values of $\Pi_i^B$ and $\Pi_i^B$ is the parents of $X_i$ in $\mathcal{B}$.

$$
P(x, c) = P(c) \prod_{i=1}^{n} P(x_i|c, \pi_i^B).
$$

(2)

For specific instance $d_i$, the log likelihood $\log P_B(d_i)$ measures how many bits are needed to describe $d_i$ based on the probability distribution $P_B$. From the statistical viewpoint, the higher the log likelihood is, the closer $\mathcal{B}$ is to model the probability distribution $P_B(d_i)$. Suppose that, for $d_i$, there exist $N_i$ instances that have the same attribute values. Let $P(d_i)$ be the empirical distribution defined by frequencies of event $d_i$, namely, $P(d_i) = \frac{N_i}{N}$. Thus $N_i \log P_B(d_i)$ or $NP(d_i) \log P_B(d_i)$ measures how many bits are needed to describe $d_i$ if there exist duplicated instances. Given enough training data, the estimate of $P(d_i)$ will approximate the joint probability distribution $P(c, x)$. The entropy function $H_B(D)$, which is defined as follows, corresponds to the optimal number of bits needed to roughly measure the amount of information represented by $\mathcal{B}$.

$$
H_B(D) = -N \cdot \sum_{d_i \in D} P(d_i) \log P_B(d_i) \\
\approx - \sum_{c,x} P(c, x) \log P_B(c, x).
$$

(3)

Eq. (3) implies that, to maximize the description of $D$ we need to find a network structure that can maximize the log likelihood or minimize the entropy function $H_B(D)$. Because $N$ denotes the size of training data and doesn’t affect the comparison results of $H_B(D)$ for different BNCs. Thus based on Eq. (3), we introduce the definition of $H_B$ as follows,

$$
H_B = - \sum_{c,x} P(c, x) \log P_B(c, x) \\
= - \sum_{c,x} P(c) \log \left( P(c) \prod_{i=1}^{n} P(x_i|c, \pi_i^B) \right) \\
= - \sum_{c,x} P(c) \log P(c) - \sum_{i=1}^{n} \sum_{x_i, \pi_i} P(c, x_i, \pi_i) \log P(x_i|c, \pi_i^B) \\
= - \sum_{c} P(c) \log P(c) - \sum_{i=1}^{n} \sum_{x_i, \pi_i} H(X_i|C, \Pi_i^B).
$$

(4)

As shown in Fig. 1(a), NB [20]–[22] is the simplest BNC and assumes that every attribute is conditionally independent from the rest of the attributes given the class variable. This independence assumption can be described as

$$
P_{NB}(x_1, \ldots, x_n|c) = \prod_{i=1}^{n} P(x_i|c).
$$

(5)

For NB, Eq. (4) turns to be

$$
H_{NB} = - \sum_{c,x} P(c, x) \log P_{NB}(c, x) \\
= - \sum_{c,x} P(c, x) \log \left( P(c) \prod_{i=1}^{n} P(x_i|c) \right).
$$
However, this independence assumption rarely holds in practice. One approach to avoiding the negative effects caused is to approximate the independence assumption and improve the probability estimates made by NB. Weighted Naive Bayes [23], [24] adjusts the naive Bayesian probabilities that may significantly improve the classification accuracy. Semi-naive Bayes models [25] introduce new attributes obtained as the Cartesian product of two or more original predictor variables.

Another approach is augmenting the network structure of NB with edges between attributes. The mostly cited and studied augmenting technique for NB so far is TAN (see Fig. 1(b)), which is proposed by Friedman et al. [12]. TAN maintains the basic structure of NB and models 1-dependence relationships among the attributes. There has been much prior work that explores approaches to achieving the high accuracy improvement [26], [27]. For TAN, Eq. (4) turns to be

$$H_{TAN} = -\sum_{C,X} P(c, x) \log P_{TAN}(c, x)$$

$$= -\sum_{C,X} P(c, x) \log \{P(c) \prod_{i=1}^{n} P(x_i|c, \pi_i^{TAN})\}$$

$$= H(C) + \sum_{i=1}^{n} H(X_i|C, \Pi_i^{TAN}),$$  

(7)

where $\Pi_i^{TAN}$ denotes the parents of $X_i$ in TAN and $|\Pi_i^{TAN}| \leq 1$. Sahami [15] proposed to extend NB to allow for the modeling of arbitrarily complex dependencies, and the final BNC, KDB (see Fig. 1(c)), allows every attribute to be conditioned on the class variable and at most $k$ other attributes. For KDB, Eq. (4) turns to be

$$H_{KDB} = -\sum_{C,X} P(c, x) \log P_{KDB}(c, x)$$

$$= -\sum_{C,X} P(c, x) \log \{P(c) \prod_{i=1}^{n} P(x_i|c, \pi_i^{KDB})\}$$

$$= H(C) + \sum_{i=1}^{n} H(X_i|C, \Pi_i^{KDB}).$$  

(8)

where $\Pi_i^{KDB}$ is the parents of attribute $X_i$ in KDB and $|\Pi_i^{KDB}| \leq k$. There have also been some important refinements that improve KDB’s performance. Bouckaert [28] proposed to average all possible network structures for a fixed value of $k$ (including lower orders). Rubio and Gámez [29] presented a variant of KDB that employs a hill-climbing search. Blanco et al. [29] suggested to perform attribute subset selection within a KDB using filter and wrapper approaches. Xiao et al. [30] argued that $k$-graph as the predictor subgraph is also the result of a kind of evolutionary computation method, which is inspired by the so-called group method of data handling (GMDH) [31].

### III. KDB AND FKDB

Although different restricted BNCs, e.g., TAN and KDB, may apply different strategies to build their own structures, they all use the network structure of NB as the basic framework and can be regarded as different extensions of NB. By comparing the difference between any restricted BNC and NB in terms of log likelihood, from Eq. (4) and Eq. (6) we have

$$H_{NB} - H_B = \{H(C) + \sum_{i=1}^{n} H(X_i|C)\}$$

$$- \{H(C) + \sum_{i=1}^{n} H(X_i|C, \Pi_i^{B})\}$$

$$= \sum_{i=1}^{n} I(X_i; \Pi_i^{B}|C).$$  

(9)

Obviously, to encode more bits in the network structure $B$ or effectively represent more significant conditional dependencies between attributes, the summation of information quantity measured by $\sum I(X_i; \Pi_i^{B}|C)$ will help to achieve this goal, although indirectly. That can clarify why TAN or KDB uses conditional mutual information to measure the conditional dependence. From Eq. (9) we can see that, the attribute order is not an important issue, but how to identify the conditional dependencies is. Thus it is reasonable for TAN to build maximum weighted spanning tree to identify the significant 1-dependence relationships without any prior

![Example of (a) NB, (b) TAN and (c) KDB.](image)
### Algorithm 1: Learning Process of KDB

1. For each attribute, calculate the mutual information $I(X_i; C)$
2. Calculate the conditional mutual information $I(X_i; X_j|C)$ for each pairwise combination of attributes ($i \neq j$).
3. Let the selected attribute list, $L$, be empty.
4. Initialize the BN, $BN$, with a single class node, $C$.
5. repeat
   6. Select the attribute $X_{\text{max}}$, which is not in $L$ and has the maximum value $I(X_{\text{max}}; C)$.
   7. Add a node to $BN$ representing $X_{\text{max}}$.
   8. Add an arc from $C$ to $X_{\text{max}}$ in $BN$.
   9. Add $m = \min(|L|, k)$ arcs from $m$ distinct attributes $X_j$ in $L$ with the highest value for $I(X_{\text{max}}, X_j|C)$.
   10. Add $X_{\text{max}}$ to $L$.
7. until $L$ includes all attributes;

attribute order. We argue that the conditional dependencies of higher degree will be more closely to fit the training data and can help BNCs achieve better generalization performance. In contrast to TAN, KDB provides an effective way to construct classifiers at arbitrary points (values of $k$) along the attribute dependence spectrum. The attribute order, which is determined by comparing mutual information, seems unnecessary as discussed above. The learning procedure of KDB is described in Algorithm 1.

Conditional mutual information is often used to measure conditional dependence whereas it is rarely used to identify implicit conditional independence. For example, given attribute order $\{X_1, \ldots, X_n\}$, KDB should select $k$ attributes as parents for attribute $X_i$ from its candidate parents $\Pi_i = \{X_1, \ldots, X_{i-1}\}$ when $i > k$. That is, KDB implicitly assumes that $X_i$ is conditionally independent from its other candidate parents, then these attributes can be regarded as redundant and removed from $\Pi_i$. KDB doesn’t check whether this assumption really holds. The classification performance of KDB will be biased to some extent due to this passive learning strategy. From the definition of $I(X_i; X_j|C)$ we cannot tell which one is the cause and which one is the effect. For example, TAN builds a maximum weighted spanning tree, then transforms this undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it. For different root variable, the arc between $X_j$ and $X_i$ may be $X_j \rightarrow X_i$ or $X_j \leftarrow X_i$. Obviously, the arc $X_j \rightarrow X_i$ in TAN just implies that $X_j$ and $X_i$ are conditionally dependent, but not that $X_j$ is the cause and $X_i$ is the effect. Bayesian network is also called causal network, but state-of-the-art BNCs, e.g., TAN or KDB, only can represent conditional dependence rather than causality.

In contrast, FKDB takes an active learning strategy and uses conditional entropy to learn significant causal relationships from data. From Fig. 2 we can see that, the distributions of $I(X_i; X_j|C)$ and $H(X_i|C, X_j)$ may differ greatly. Thus if respectively using these two measures to identify parent attributes $\Pi_i$ for attribute $X_i$, we may get different results and that will lead to different network topologies. Subsumption resolution assumes that, if $P(x_i|x_j) = 1$ then

![Figure 2](image-url)
the attribute value $x_i$ is determined by $x_j$, and $x_i$ can be regarded as redundant if $x_j$ exists, i.e., $P(x_i|c, x_j(x_j) = P(x_i|c)$, that reflects the causal relationship $x_j \rightarrow x_i$ to some extent. Similarly, FKDB sorts attributes by comparing $H(X_i|C, \Pi_i)$. The lower the value of $H(X_i|C, \Pi_i)$ is, the more possibly that the values of $X_i$ are determined by the values of $\{C, \Pi_i\}$. FKDB selects the first attribute with the minimum of $H(X_i|C)$, and selects the second attribute with the minimum of $H(X_i|C, X_j)$, then selects the third attribute with the minimum of $H(X_i|C, X_j, X_k)$ and so on. Correspondingly, we will have a set of weak or strong causalities including $C \rightarrow X_i, \{C, X_i\} \rightarrow X_j$ and $\{C, X_j, X_k\} \rightarrow X_k$. After all attributes have been sorted in this way, each candidate parent in $\Pi_i = \{X_1, \cdots, X_{i-1}\}$ for attribute $X_i$ will show redundancy characteristics of different levels. When the structure complexity is restricted, we can select subset of $\Pi_i$ that can minimize the conditional entropy of attribute $X_i$. For example, attribute $X_k$ takes $X_j$ and $\{X_j, C\}$ as its parents, $X_j$ can be regarded as redundant to a certain extent and then removed from $\Pi_k$.

An ideal network topology should correspond to the minimum of the entropy function $H_B$. Based on the discussion above, we propose a heuristic search strategy to identify the conditional dependence by comparing high-order conditional entropy $H(X_i|C, \Pi_i^B)$, which is one of the factors in $H_B$ (see Eq. (4)), and then build a high-order maximum weighted spanning tree. The learning procedure of the proposed algorithm, called flexible KDB (FKDB), is described in Algorithm 2.

Algorithm 2: Learning Process of FKDB

1. Let the selected attribute list, $\mathcal{L}$, be empty.
2. Initialize the BN, $BN$, with a single class node, $C$.
3. repeat
   4. Select the attribute $X_i$, which is not in $\mathcal{L}$ and corresponds to the minimum of $H(X_i|\Pi_i, C)$, where $\Pi_i \subseteq \mathcal{L}$ and $|\Pi_i| = \min(|\mathcal{L}|, k)$;
   5. Add a node to $BN$ representing $X_i$;
   6. Add arcs from attributes in $\Pi_i$ to $X_i$ in $BN$;
   7. Add $X_i$ to $\mathcal{L}$;
4. until $\mathcal{L}$ includes all attributes;

IV. EXPERIMENTS

In this section, we compares our proposed FKDB algorithm’s results with other BNCs and some state-of-the-art algorithms, such as support vector machine, logistic regression and neural network. Each algorithm is tested on each dataset using 10 rounds of 10-fold cross validation. Probability estimates are smoothed using $m$-estimation with $m = 1$ [32].

To allow the proposed algorithm to be compared with Weka’s algorithms, missing values for qualitative attributes are replaced with modes and those for quantitative attributes are replaced with means from the training data. The performance is analyzed in terms of zero-one loss, bias and variance on 40 datasets from the UCI (University of California at Irvine) repository of machine learning [34]. We employ the Win/Draw/Loss (WDL) record to interpret the results, when two algorithms are compared, we count the number of datasets for which one algorithm performs better, equally well or worse than the other on a given measure. We assess a difference as significant if the outcome of a one-tailed binomial sign test is less than 0.05.

Table 1 describes the details of each dataset used, including the number of instances, attributes and classes. These datasets are categorized in terms of their sizes. That is, datasets with instances $< 1000$, $\geq 1000$ and $< 10000$, $\geq 10000$ are denoted as small size, medium size and large size, respectively. The experiments are conducted in the Weka work-branch (version 3.5.7) and the following algorithms are compared:

- NB, Naive Bayes.
- TAN, tree augmented Naive Bayes.
- AODE, averaged one-dependence estimators.
- K2DB, k-dependence Bayesian classifier with $k = 2$.
- FK2DB, flexible k-dependence Bayesian classifier with $k = 2$.
- SVM, support vector machine with default parameters.
- LR, logistic regression with default parameters.
- NN, neural network with default parameters.

A. ZERO-ONE LOSS

Zero-one loss is one of the most commonly used loss function to measure the classification performance. Table 2 reports

| Table 1. Datasets. |
|-------------------|
| No. | Dataset | Instance Attribute Class |
|-----|---------|------------------------|
| 1   | Contact-lenses | 24 | 5 |
| 2   | Lung-cancer | 32 | 6 |
| 3   | Liver-cancer | 57 | 6 |
| 4   | Tissue | 101 | 16 |
| 5   | PROMoters | 106 | 2 |
| 6   | Lymphography | 148 | 18 |
| 7   | Teaching-ass | 151 | 3 |
| 8   | Auto | 205 | 25 |
| 9   | Sonar | 208 | 60 |
| 10  | Glass | 214 | 9 |
| 11  | New-diabetes | 215 | 16 |
| 12  | Auto | 226 | 69 |
| 13  | Soothing-large | 307 | 35 |
| 14  | Ionosphere | 351 | 34 |
| 15  | Dermatology | 366 | 34 |
| 16  | Horse-colic | 368 | 21 |
| 17  | Horse-voices-84 | 435 | 16 |
| 18  | Musk1 | 476 | 166 |
| 19  | Chars | 551 | 39 |
| 20  | Credit-a | 690 | 15 |
| 21  | Breast-cancer-w | 699 | 9 |
| 22  | Annals | 898 | 38 |
| 23  | Tic-tac-toe | 958 | 9 |
| 24  | German | 1,000 | 20 |
| 25  | Led | 1,000 | 7 |
| 26  | Mfeat-ice | 2,000 | 6 |
| 27  | Hypothyroid | 3,163 | 25 |
| 28  | Hypo | 3,772 | 29 |
| 29  | Sick | 3,772 | 29 |
| 30  | Abalone | 4,177 | 8 |
| 31  | Thyroid | 9,169 | 29 |
| 32  | Prognosis | 10,992 | 16 |
| 33  | Sys | 12,546 | 8 |
| 34  | Nursery | 12,960 | 8 |
| 35  | Magic | 19,020 | 10 |
| 36  | Adult | 48,842 | 14 |
| 37  | Shuttle | 58,000 | 9 |
| 38  | Connect-4 | 67,557 | 42 |
| 39  | Waveform | 100,000 | 21 |
| 40  | Localization | 164,860 | 5 |

VOLUME 7, 2019
TABLE 2. Experimental results of zero-one loss.

| Dataset  | NB | TAN | AODE | KB2_DB | FK2_DB | SVM | NN | LR |
|----------|----|-----|------|--------|--------|-----|----|----|
| Contact-lenses | 0.3750 | 0.3750 | 0.3750 | 0.3750 | 0.3750 | 0.3750 | 0.3750 | 0.3750 |
| Letter | 0.1975 | 0.1975 | 0.1975 | 0.1975 | 0.1975 | 0.1975 | 0.1975 | 0.1975 |
| Glass | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 |
| Proximal | 0.7550 | 0.7550 | 0.7550 | 0.7550 | 0.7550 | 0.7550 | 0.7550 | 0.7550 |
| Synthep | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 |
| Treenning | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 |
| Astor | 0.3213 | 0.3213 | 0.3213 | 0.3213 | 0.3213 | 0.3213 | 0.3213 | 0.3213 |
| MB | 0.2638 | 0.2638 | 0.2638 | 0.2638 | 0.2638 | 0.2638 | 0.2638 | 0.2638 |
| Glass | 0.2817 | 0.2817 | 0.2817 | 0.2817 | 0.2817 | 0.2817 | 0.2817 | 0.2817 |
| New-thyroid | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 | 0.4646 |
| ADX | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 |
| Soybean-large | 0.5132 | 0.5132 | 0.5132 | 0.5132 | 0.5132 | 0.5132 | 0.5132 | 0.5132 |
| Synthep | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 | 0.2597 |
| Dermatology | 0.1313 | 0.1313 | 0.1313 | 0.1313 | 0.1313 | 0.1313 | 0.1313 | 0.1313 |
| Horse-colic | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 |
| Horse-colic | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 | 0.2174 |
| Musk | 0.1400 | 0.1400 | 0.1400 | 0.1400 | 0.1400 | 0.1400 | 0.1400 | 0.1400 |
| Glass | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 |
| Cond | 0.1046 | 0.1046 | 0.1046 | 0.1046 | 0.1046 | 0.1046 | 0.1046 | 0.1046 |

TABLE 3. Win/Draw/Loss comparison results of zero-one loss on all datasets.

| Classifier | NB | TAN | AODE | KB2_DB | FK2_DB | SVM | NN | LR |
|------------|----|-----|------|--------|--------|-----|----|----|
| TAN | - | - | - | - | - | - | - | - |
| AODE | 27/2/4 | 17/1/0 | 17/1/0 | 17/1/0 | 17/1/0 | 17/1/0 | 17/1/0 | 17/1/0 |
| KB2_DB | 22/3/1 | 14/1/1 | 14/1/1 | 14/1/1 | 14/1/1 | 14/1/1 | 14/1/1 | 14/1/1 |
| SVM | 0/4/7/ | 9/2/0 | 2/4/2 | 2/4/2 | 2/4/2 | 2/4/2 | 2/4/2 | 2/4/2 |
| NN | 24/3/1 | 17/1/1 | 17/1/1 | 17/1/1 | 17/1/1 | 17/1/1 | 17/1/1 | 17/1/1 |
| LR | 14/7/19 | 10/7/19 | 10/7/19 | 10/7/19 | 10/7/19 | 10/7/19 | 10/7/19 | 10/7/19 |
| FK2_DB | 26/3/1 | 25/3/1 | 25/3/1 | 25/3/1 | 25/3/1 | 25/3/1 | 25/3/1 | 25/3/1 |

FIGURE 3. The fitting curve of GD between FK2_DB and KB2_DB in terms of zero-one loss.

The zero-one loss results of the above algorithms on different datasets. Table 3 presents corresponding WDL records. From Table 3 we can see that, NB performs better than SVM and LR, but poorer than NN (13 wins and 24 losses).

As the structure complexity increases, the augmented edges will help relax the independence assumption of NB. TAN and KB2_DB shows competitive generalization performance when compared with NN, and they still retain the advantage over SVM and LR. In contrast, FK2_DB shows significant advantage over these classifiers, it respectively beats SVM, NN and LR on 31, 20 and 27 datasets.

To clarify the effectiveness of the measurement of conditional entropy, we compare the classification performance of FK2_DB with that of KB2_DB in terms of goal difference (GD) [35]. Given two classifiers A and B, the value of GD can be computed as follows:

\[ GDA(A; B| T) = |\text{win}| - |\text{loss}|, \]

where \( T \) denotes the set of datasets for comparison, \(|\text{win}|\) and \(|\text{loss}|\) represent the number of datasets on which A performs better or worse than B, respectively. Fig. 3 shows the fitting curve of \( GD(FK2_DB; KB2_DB|S_k) \) in terms of zero-one loss. The X-axis represents the indexes of different datasets, referred to as k, which correspond to the indexes described in Table 1, and the Y-axis represents the value of \( GD(FK2_DB; KB2_DB|S_k) \), where \( S_k = \{D_j | j < k \} \) and \( D_j \) denotes the dataset with index j.

From Fig. 3 we can see that, while dealing with datasets of different sizes, FK2_DB enjoys significant advantage over KB2_DB in terms of zero-one loss. A notable case is Tic-tac-toe, on which the zero-one loss result for KB2_DB is 0.2035 while that for FK2_DB is 0.0689. Conditional entropy is more reasonable for learning BNC but not definitely better than conditional mutual information. They both suffer from local optimal but global non-optimal solutions.

Given \( n \) attributes, there are \( n! \) alternative orders, thus before learning the network topology BNCs often impose an ordering on the attributes implicitly (like TAN) or explicitly (like KDB). However, because the attributes are sorted one by one, the m best attributes that rank first in the order may not be the best m attributes. Besides, the estimate of high-order conditional entropy may be biased due to limited number of training instance. From the experimental results we can see that, among all of the 40 datasets FK2_DB performs poorer than KB2_DB on 4 datasets, and 3 of them (i.e., datasets Contact-lenses, Promoters, Horse-colic) have less than 400 training instances.

B. BIAS AND VARIANCE

Bias-variance decomposition of zero-one loss is presented by Kohavi and Wolpert [33] from sampling theory statistics for analyzing different learning scenarios. Bias denotes the systematic component of error, which describes how closely the learner is able to describe the surfaces for a
TABLE 4. Experimental results of bias.

| Dataset          | NB   | TAN   | AODE  | K-DB  | PK-DB  | SVM   | NN    | LR    |
|------------------|------|-------|-------|-------|--------|-------|-------|-------|
| Concor-dearte    | 0.283| 0.285 | 0.280 | 0.315 | 0.305  | 0.482 | 0.454 | 0.423 |
| Lung-cancer      | 0.200| 0.197 | 0.194 | 0.218 | 0.215  | 0.501 | 0.472 | 0.441 |
| Liver-cancer     | 0.295| 0.298 | 0.302 | 0.331 | 0.332  | 0.491 | 0.460 | 0.431 |
| Breast-cancer-w  | 0.318| 0.318 | 0.318 | 0.345 | 0.343  | 0.491 | 0.460 | 0.431 |
| Breast-cancer-m  | 0.095| 0.095 | 0.095 | 0.105 | 0.105  | 0.095 | 0.095 | 0.095 |
| Adult             | 0.318| 0.318 | 0.318 | 0.345 | 0.343  | 0.491 | 0.460 | 0.431 |
|heart             | 0.269| 0.269 | 0.269 | 0.297 | 0.297  | 0.481 | 0.451 | 0.421 |
| Ionosphere       | 0.256| 0.256 | 0.256 | 0.285 | 0.285  | 0.471 | 0.441 | 0.411 |
| Statlog-german    | 0.231| 0.231 | 0.231 | 0.259 | 0.259  | 0.461 | 0.431 | 0.401 |
| Statlog-australian| 0.231| 0.231 | 0.231 | 0.259 | 0.259  | 0.461 | 0.431 | 0.401 |

TABLE 5. Experimental results of variance.

| Dataset          | NB   | TAN   | AODE  | K-DB  | PK-DB  | SVM   | NN    | LR    |
|------------------|------|-------|-------|-------|--------|-------|-------|-------|
| Concor-dearte    | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Lung-cancer      | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Liver-cancer     | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Breast-cancer-w  | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Breast-cancer-m  | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Adult             | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
|heart             | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Ionosphere       | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Statlog-german    | 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |
| Statlog-australian| 0.111| 0.111 | 0.111 | 0.141 | 0.141  | 0.321 | 0.291 | 0.261 |

TABLE 6. Win/Draw/Loss comparison results of bias and variance on all datasets.

| Classifier       | NB   | TAN   | AODE  | K-DB  | PK-DB  | SVM   | NN    | LR    |
|------------------|------|-------|-------|-------|--------|-------|-------|-------|
| TAN              | 22/0 |
| AODE             | 28/0 |
| K-DB             | 22/0 |
| PK-DB            | 28/0 |
| SVM              | 6/6/2 |
| NN               | 25/12 |
| LR               | 25/12 |
| TAN              | 22/0 |
| AODE             | 28/0 |
| K-DB             | 22/0 |
| PK-DB            | 28/0 |
| SVM              | 6/6/2 |
| NN               | 25/12 |
| LR               | 25/12 |
| TAN              | 22/0 |
| AODE             | 28/0 |
| K-DB             | 22/0 |
| PK-DB            | 28/0 |
| SVM              | 6/6/2 |
| NN               | 25/12 |
| LR               | 25/12 |
| TAN              | 22/0 |
| AODE             | 28/0 |
| K-DB             | 22/0 |
| PK-DB            | 28/0 |
| SVM              | 6/6/2 |
| NN               | 25/12 |
| LR               | 25/12 |
| TAN              | 22/0 |
| AODE             | 28/0 |
| K-DB             | 22/0 |
| PK-DB            | 28/0 |
| SVM              | 6/6/2 |
| NN               | 25/12 |
| LR               | 25/12 |

C. ANALYSIS OF CLASSIFICATION AND TRAINING TIME

The results of average classification and training time for all the compared algorithms are shown in Fig. 4, where the X-axis represents the algorithms, and the Y-axis corresponds to the time of the algorithm.

All experiments are conducted on a desktop computer with an Intel(R) Core(TM) i5-8300H CPU @ 2.3 GHz, 64 bits and 8 GB of memory. Each bar represents the sum of time on 40 datasets in the 10-fold cross validation. Fig. 4 indicates that the X-axis represents the algorithms, and the Y-axis corresponds to the time of the algorithm.
(e.g., TAN). FK2DB takes more time for training because it needs to build high-order maximum weighted spanning tree in terms of high-order conditional entropy. The ensemble learning strategy imposes high classification time on AODE. In contrast to BNCs, NN, SVM and LR take extremely more time for training.

V. CONCLUSION AND FUTURE WORK
In many real-world applications, although the application of conditional mutual information for learning BNC presents strong asymptotic guarantees, it does not necessarily optimize the classification performance. Our analysis suggests that using conditional entropy to measure the causal relationships between attributes may be more helpful to learn the network topology. From the experimental results, the final BNC achieves the trade-off between classification performance and structure complexity. To reduce search space, the attributes are sorted one by one by comparing conditional entropy, that may result in local optimal but global non-optimal solutions. It remains a direction for future research to explore techniques for sorting attribute values in each instance. Such a variant of FKDB would show excellent flexibility in representing the diversity of conditional dependencies in different situations.

REFERENCES
[1] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
[2] H. Yang and J. Moody, “Data visualization and feature selection: New algorithms for nongaussian data,” in Proc. NIPS, 2000, pp. 687–693.
[3] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[4] M. Vidal-Naquet and S. Ullman, “Object recognition with informative features and linear classification,” in Proc. ICCV, 2003, pp. 281–288.
[5] F. Fleuret, “Fast binary feature selection with conditional mutual information,” J. Mach. Learn. Res., vol. 5, pp. 1531–1555, Nov. 2004.
[6] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, “Normalized mutual information feature selection,” IEEE Trans. Neural Netw., vol. 20, no. 2, pp. 189–201, Feb. 2009.
[7] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Burlington, MA, USA: Morgan Kaufmann, 1988.
[8] M. Pazzani and D. Billsus, “Learning and revising user profiles: The identification of interesting Web sites,” Mach. Learn., vol. 27, no. 3, pp. 313–331, 1997.
[9] R. Blanco, I. Inza, M. Merino, J. Quiroga, and P. Larraga, “Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS,” J. Biomed. Inform., vol. 38, no. 5, pp. 376–388, 2005.
[10] J. Li, C. Zhang, T. Wang, and Y. Zhang, “Generalized additive Bayesian network classifiers,” in Proc. 20th IJCAI, Hyderabad, India, 2007, pp. 6–12.
[11] J. B. Kruskal, Jr., “On the shortest spanning subtree of a graph and the traveling salesman problem,” Proc. Amer. Math. Soc., vol. 7, no. 1, pp. 48–50, Feb. 1956.
[12] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” Mach. Learn., vol. 29, no. 2, pp. 131–163, Nov. 1997.
[13] F. Pernkopf and J. A. Bilmes, “Efficient heuristics for discriminative structure learning of Bayesian network classifiers,” J. Mach. Learn. Res., vol. 11, pp. 2323–2360, Aug. 2010.
[14] Y. Jing, V. Pavlović, and J. M. Rehg, “Boosted Bayesian network classifiers,” Mach. Learn., vol. 73, no. 2, pp. 155–184, Nov. 2008.
[15] M. Sahami, “Learning limited dependence Bayesian classifiers,” in Proc. 2nd SIGKDD, Portland, OR, USA, 1996, pp. 335–338.
[16] D. Mladenic and M. Grobelnik, “Feature selection for unbalanced class distribution and Naive Bayes,” in Proc. Int. Conf. Mach. Learn., 1999, pp. 258–267.
[17] E. R. Hruschka, Jr., and N. F. F. Ebecken, “Towards efficient variables ordering for Bayesian networks classifier,” Data Knowl. Eng., vol. 63, pp. 258–269, Nov. 2007.
[18] J. A. Bilmes, “Dynamic Bayesian multinet,” in Proc. 16th Conf. Uncertainty Artif. Intell., 2000, pp. 38–45.
[19] W. J. McGill, “Multivariate information transmission,” IEEE Trans. Inf. Theory, vol. IT-4, no. 4, pp. 93–111, Sep. 1954.
[20] M. E. Maron and J. L. Kuhns, “On relevance, probabilistic indexing and information retrieval,” J. ACM, vol. 7, no. 3, pp. 216–244, Jul. 1960.
[21] M. Minsky, “Steps toward artificial intelligence,” Proc. IRE, vol. 49, no. 1, pp. 8–30, Jan. 1961.
[22] R. Vilalta and I. Rish, “A decomposition of classes via clustering to explain and improve Naive Bayses,” in Proc. ECML. Berlin, Germany: Springer, 2003, pp. 444–445.
[23] A. Rubio and J. A. Gámez, “Flexible learning of K-dependence Bayesian classifiers,” in Proc. GECCO, 2011, pp. 207–219, 2011.
[24] M. Sahami, “Learning limited dependence Bayesian classifiers,” in Proc. 2nd SIGKDD, Portland, OR, USA, 1996, pp. 335–338.
[25] E. J. Keogh, “Dynamic Bayesian multinet,” in Proc. 16th Conf. Uncertainty Artif. Intell., 2000, pp. 38–45.
[26] J. A. Bilmes, “Dynamic Bayesian multinet,” in Proc. 16th Conf. Uncertainty Artif. Intell., 2000, pp. 38–45.
[27] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, “Alleviating the attribute independence assumption by attribute weighting,” J. Mach. Learn. Res., vol. 14, no. 1, pp. 1947–1988, Jul. 2013.
[28] G. I. Webb and M. J. Pazzani, “Adjusted probability Naive Bayesian induction,” in Proc. AICAJ. Berlin, Germany: Springer, 1998, pp. 285–295.
[29] F. Zheng, G. I. Webb, P. Suraweera, and L. Zhu, “Subsumption resolution: An efficient and effective technique for semi-naive Bayesian learning,” Mach. Learn., vol. 87, no. 1, pp. 93–125, Dec. 2012.
[30] M. Minsky, “Steps toward artificial intelligence,” Proc. IRE, vol. 49, no. 1, pp. 8–30, Jan. 1961.
[31] R. Vilalta and I. Rish, “A decomposition of classes via clustering to explain and improve Naive Bayes,” in Proc. ECML. Berlin, Germany: Springer, 2003, pp. 444–445.
[32] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, “Alleviating the attribute independence assumption by attribute weighting,” J. Mach. Learn. Res., vol. 14, no. 1, pp. 1947–1988, Jul. 2013.
[33] G. I. Webb and M. J. Pazzani, “Adjusted probability Naive Bayesian induction,” in Proc. AICAJ. Berlin, Germany: Springer, 1998, pp. 285–295.
[34] F. Zheng, G. I. Webb, P. Suraweera, and L. Zhu, “Subsumption resolution: An efficient and effective technique for semi-naive Bayesian learning,” Mach. Learn., vol. 87, no. 1, pp. 93–125, Dec. 2012.
[35] E. J. Keogh and M. J. Pazzani, “Learning the structure of augmented Bayesian classifiers,” Int. J. Artif. Intell. Tools, vol. 11, no. 4, pp. 587–601, 2002.
[36] F. Pernkopf and P. O’Leary, “Floating search algorithm for structure learning of Bayesian network classifiers,” Pattern Recogn. Lett., vol. 24, no. 15, pp. 2839–2848, 2003.
[37] R. R. Bouckaert, “Voting massive collections of Bayesian network classifiers for data streams,” in Proc. AICAJ. Berlin, Germany: Springer, 2006, pp. 243–252.
[33] R. Kohavi and D. H. Wolpert, “Bias plus variance decomposition for zero-one loss functions,” in Proc. ICML, 1996, pp. 275–283.

[34] P. M. Murphy and D. W. Aha. UCI Repository of Machine Learning Databases. Accessed: Jun. 11, 2019. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html

[35] Z. Duan and L. Wang, “K-dependence Bayesian classifier ensemble,” Entropy, vol. 19, no. 12, p. 651, Nov. 2017.

LIMIN WANG received the Ph.D. degree in computer science from Jilin University, China, in 2005, where he is currently a Professor with the College of Computer Science and Technology. His research interests include probabilistic logic inference and Bayesian networks.

GAOJIE WANG received the B.Eng. degree from Jilin University, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. His research interests include Bayesian networks and data analysis.

ZHIYI DUAN received the B.Eng. degree from Jilin University, China, in 2014, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. His research interests include data mining and Bayesian networks.

HUA LOU received the B.Eng. degree in Zhengzhou University, HeNan, China, in 2004. He is currently an Associate Professor with the Department of Software and Big Data, Changzhou College of Information Technology, China. His research interests include abnormal detection and data analysis.

MINGHUI SUN received the Ph.D. degree in computer science from the Kochi University of Technology, Japan, in 2011. He is currently an Assistant Professor with the College of Computer Science and Technology, Jilin University, China. He is interested in using HCI methods to solve challenging real-world computing problems in many areas, including tactile interface, pen-based interface, and tangible interface.

* * *