Model Weighting for One-Dependence Estimators by Measuring the Independence Assumptions

HUA LOU1, GAOJIE WANG2, LIMIN WANG2,3, AND MUSA MAMMADOV4

1Department of Software and Big Data, Changzhou College of Information Technology, Changzhou 213164, China
2College of Computer Science and Technology, Jilin University, Changchun 130012, China
3Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
4School of Information Technology, Deakin University, Burwood, VIC 3125, Australia

Corresponding author: Limin Wang (wanglim@mail.jlu.edu.cn)

This work was supported by the Development Project of Jilin Province under Grant 20150101014JC and Grant 20200201281JC.

ABSTRACT The superparent one-dependence estimators (SPODEs) is a popular family of semi-naive Bayesian network classifiers, and the averaged one-dependence estimators (AODE) provides efficient single pass learning with competitive classification accuracy. All the SPODEs in AODE are treated equally and have the same weight. Researchers have proposed to apply information-theoretic metrics, such as mutual information or conditional log likelihood, for assigning discriminative weights. However, while dealing with different instances the independence assumptions for different SPODEs may hold to different extents. The quest for highly scalable learning algorithms is urgent to approximate the ground-truth attribute dependencies that are implicit in training data. In this study we set each instance as the target and investigate extensions to AODE by measuring the independence assumption of SPODEs and assigning weights. The proposed approach, called independence weighted AODE (IWAODE), is validated on 40 benchmark datasets from the UCI machine learning repository. Experimental results reveal that, the resulting weighted SPODEs delivers computationally efficient low-bias learning, proving to be a competitive alternative to state-of-the-art single and ensemble Bayesian network classifiers (such as tree-augmented naive Bayes, k-dependence Bayesian classifier, WAODE-MI and etc).

INDEX TERMS Averaged one-dependence estimators, discriminative weights, conditional independence.

I. INTRODUCTION Learning classifiers from data is one of the most important research topics in data mining. Bayesian network (BN) [1]–[3] has long been considered a popular medium for graphically representing the probabilistic relationships among variables of interest in an annotated directed graph. Meanwhile, learning the topology of BN has been proven to be an NP-hard problem [4]. During the past decades, BN has elicited considerable attention from researchers as a competitive alternative to other state-of-the-art classifiers particularly after the success of naive Bayes (NB) [5]–[9], which is an extremely simple and remarkably effective approach to classification. NB assumes that the attributes \(X_1, \ldots, X_n\) are independent of one another given the class variable \(Y\) [10], and this unrealistic assumption would harm the performance of NB in research domains with complex attribute dependencies.

Bayesian network classifiers (BNCs) can be roughly divided into two categories, single BNCs and ensemble BNCs. To weaken NB’s independence assumption, single BNCs, such as tree-augmented naive Bayes (TAN) [11] and \(k\)-dependence Bayesian classifier (KDB) [12], represent complex dependency relationships among attributes by adding augmented edges to NB-based topology [10]–[15]. Ensemble BNCs, such as averaged one-dependence estimators (AODE) [14], focus on the diversity of conditional dependencies in the relatively simple topologies of committee members. Kohavi and Wolpert [16] proposed the bias-variance decomposition to provide valuable insights into the classification performance of learned BNCs. Single BNCs with simple topology (e.g., NB) exhibit high bias and low variance, and those with high-dependence relationships (e.g., KDB) exhibit low bias and high variance. In contrast, AODE aggregates a restricted class of superparent...
one-dependence estimators (SPODEs), with each SPODE setting a different attribute as the superparent and assuming that the rest of the attributes are conditionally independent of each other given the class variable \( Y \) and the superparent [17]. The unrealistic independence assumption helps to reduce variance, and the diversity in independence assumptions of different SPODEs helps to reduce bias. Thus AODE demonstrates significant advantage in the trade-off between bias and variance.

Although SPODEs in AODE correspond to different independence assumptions, they are treated equally and are assigned with the same weight. Model weighting has primarily been viewed as a means of increasing the influence of highly predictive models and discounting models that have little predictive value. For example, the information-theoretic metrics, e.g., mutual information [18] or information gain [19], are introduced to directly measure how predictive each individual SPODE is. Most BNCs assume that the training instances are drawn independently and identically from an unknown joint probability distribution. However, this i.i.d assumption is often violated in practice, and the dependency relationships between attributes may vary when they take different values. Ever increasing data quantity makes ever more urgent the need for highly scalable learners. We argue that, the primary value of model weighting is its capacity to mitigate the negative effect caused by violations of the independence assumption of SPODE members. If the weights are irrelevant to the independence assumption of SPODE members while classifying specific unlabeled testing instances, this non-scalable weighting strategy will degrade the generalization performance of AODE.

In this article we propose a novel weighting approach for improving AODE, independence weighting (IW), which takes each instance as the target and measure the independence assumption in terms of log likelihood. The weights assigned to SPODEs will be self-adaptive for different instances, that will help improve both the interoperability of these SPODEs and the generalization performance of the final ensemble BNC. The experimental results reveal that IW is effective at reducing AODE’s bias and increasing its classification accuracy.

The rest content of the paper is arranged as follows: Section 2 reviews some state-of-the-art BNCs, and describes the relationship between explicit (or implicit) independence assumptions and the learned topology. Section 3 introduces the basic idea and learning procedure of the proposed algorithm, independence weighted AODE (IWAODE). Section 4 presents the experimental setup and comparison results for IWAODE on 40 datasets with related approaches. The final section draws conclusions and outlines some directions for further research.

II. RELATED WORK
The task of learning BNC can be divided into two subtasks: (1) structure learning, i.e., identification of the topology \( G \), and (2) parameter learning, i.e., estimation of the conditional probabilities \( \Theta \) according to the topology \( G \). The most challenging task is how to learn \( G \) from data. The topology \( G \) is a directed acyclic graph where the nodes correspond to domain variables \( \{X_1, \ldots, X_n, Y\} \) and the arcs between nodes represent direct dependencies between these variables. Given instance \( d = (x, y) = (x_1, \ldots, x_n, y) \), as shown in Fig.1, for full BNC the joint probability can be factorized to the product of the individual conditional probabilities as follows,

\[
P(x, y) = P(y) \prod_{i=1}^{n} P(x_i|x_1, \ldots, x_{i-1}, y)
\]

\[
= P(y) \prod_{i=1}^{n} P(x_i|x_{\Pi_i}, y),
\]

where \( \Pi_i = \{X_1, \ldots, X_{i-1}\} \) denotes the parent attributes of attribute \( X_i \). By mining significant conditional dependencies between \( X_i \) and \( \{X_1, \ldots, X_{i-1}\} \) to reduce structure complexity and variance, the topology \( G \) of BNC learned from training data \( D \) often chooses a subset of \( \{X_1, \ldots, X_{i-1}\} \) as \( \Pi_i \) in practice. Ideally the log likelihood of the BNC should be maximized by using the empirical estimates of \( P(x_i|\Pi_i, y) \).

As shown in Fig.2, NB assume that each attribute is independent from the rest of attributes given the class, i.e., \( \Pi_i = \emptyset \). The explicit independence assumption for \( X_i \) can be described by

\[
P(x_1, \ldots, x_n|y) = \prod_{i=1}^{n} P(x_i|y).
\]

Or more precisely, the conditional probability of \( X_i \) for full BNC turned from \( P(x_i|x_1, \ldots, x_{i-1}, y) \) to be \( P(x_i|y) \)
for NB, and corresponding independence assumption can be described as follows,

$$P(x_1, \ldots, x_n|y) = P(x_1, \ldots, x_{i-1}|y)P(x_i|y).$$

(3)

As shown in Fig.3, TAN [9], [11] is a structural augmentation of NB, and it assumes that each attribute can have at most one other attribute as its parent. By utilizing conditional mutual information to measure the conditional dependence between attributes and find a maximum spanning tree, the topology of TAN is regarded as an extension of the Chow-Liu tree [20]. Suppose that $X_1$ is the only parent attribute of attribute $X_i$ for TAN, thus the conditional probability of $X_i$ turns to be $P(x_i|x_1, y)$, and the implicit independence assumption is

$$P(x_2, \ldots, x_n|x_1, y) = P(x_2, \ldots, x_{i-1}|x_1, y)P(x_i|x_1, y).$$

(4)

As shown in Fig.4, KDB [12] further allows each attribute to have a maximum of $k$ attributes as parents apart from the class. The inclusion order of the predictor attribute $X_i$ in the model is given by $I(X_i; Y)$, starting with the highest. Suppose that the final order is $\{X_1, \ldots, X_n\}$, then the candidate parent attributes of attribute $X_i$ is $\Pi_i = \{X_1, \ldots, X_{i-1}\}$. According to the topology of the learned KDB, if the parent attributes of attribute $X_i$ is $\Pi_i = \{X_{i_1}, \ldots, X_{i_k}\} (1 \leq n_k \leq k)$, the conditional probability of $X_i$ will turn to be $P(x_i|\Pi_i, y)$, and the implicit independence assumption is

$$P(x_i, \Pi_i/\Pi_i|\Pi_i, y) = P(\Pi_i/\Pi_i|\Pi_i, y)P(x_i|\Pi_i, y).$$

(5)

Complex topology can help reduce bias, whereas limited number of training instances may lead to the increase in variance and biased estimate of conditional probability. TAN or KDB, provides an intermediate bias-variance trade-off, standing between NB with strict independence assumption on one hand and full BNC with no assumption on the other.

AODE retains the simplicity and direct theoretical foundation of NB. It builds a collection of $n$ SPODEs by letting each of the attributes be superparent in one of the SPODEs [14]. Thus AODE doesn’t need to learn the network topology of each SPODE, thereby decreasing the variance component of the classifier and not incurring the same order of computational overhead as TAN and KDB. For SPODE$^a$ that takes $X_a$ as the superparent, its independence assumption can be described as follows,

$$P_{\text{SPODE}^a}(x|x_a, y) = \prod_{i=1}^n P(x_i|x_a, y).$$

(6)

To factor out the biased estimate of probability distributions from Eq.(6), while classifying instance $x = \{x_1, \ldots, x_n\}$ AODE excludes SPODE with superparent $X_a$ if the training data contains fewer than $m$ instance with attribute value $x_a$, and $m = 30$ is widely utilized as the minimum on sample size for statistical inference purposes. The prediction of AODE is produced by averaging the predictions of all qualified SPODEs, and the estimate of joint probability for AODE is

$$P_{\text{AODE}}(x, y) = \frac{\sum_{\alpha=1 \land F(x_a) \geq 30} P(y, x_a)P_{\text{SPODE}^a}(x_1, \ldots, x_n|x_a, y)|\alpha : 1 \leq \alpha \leq n \land F(x_a) \geq 30)}{\sum_{\alpha=1 \land F(x_a) \geq 30} W_\alpha},$$

(7)

Weighting has been proven to be a simple, efficient, and effective solution to alleviate the conditional independence assumption and improve the estimate of probability distribution. For NB, correlation-based feature weighting takes the attributes with maximum mutual relevance and minimum average mutual redundancy as highly predictive ones and assign greater weights to them [8]. Deep feature weighting approach incorporated the attribute weights into the conditional probability estimates [7]. For AODE, WAODE [18] sets the weight $W_\alpha$ as the mutual information $I(X_a; Y)$, where $X_a$ is the root attribute. WAODE is effective at reducing the bias of AODE with minimal computational overhead. The required joint probability is estimated using

$$P_{\text{AODE}}(x, y) = \frac{\sum_{\alpha=1 \land F(x_a) \geq 30} W_\alpha P(y, x_a)P_{\text{SPODE}^a}(x|x_a, y)}{\sum_{\alpha=1 \land F(x_a) \geq 30} W_\alpha},$$

(8)
where \( F(x_a) \) is the frequency of attribute-value \( x_a \) in the training data and is used to enforce the limit that we place on the support needed in order to accept a conditional probability estimate. For different instances, the correlation or conditional dependence between attribute values may vary greatly [15], [21]. Different SPODEs may fit the same instance to different extents, thus they should be treated differently and assigned with distinctive weights. Take dataset Vowel (see Table 2 for detail) as an example. Dataset Vowel has 13 attributes and only 990 instances, thus the independence assumptions seem to hold to greater extents. If \( X_1 \) and \( X_2 \) are conditionally independent given \( \{ X_i, Y \} \), then
\[
P(x_1, x_2 | x_i, y) = P(x_1 | x_i, y)P(x_2 | x_i, y)
\] (9)
or
\[
P_{1,2}^a = P(x_1, x_2 | x_i, y)P(x_1 | x_i, y)P(x_2 | x_i, y) = 1
\] (10)
The distributions of the values of \( P_{1,2}^a \) for SPODEs are shown in Fig.6. If the extents to which the independence assumptions for different SPODEs hold remain the same while dealing with different instances, it is appropriate to assign the fixed weights to individual SPODEs. However, from Fig 6 we can see that, the distributions of \( P_{1,2}^a \) varies greatly for SPODE\(^{X_1} \), SPODE\(^{X_2} \) and SPODE\(^{X_3} \), and while dealing with different instances the independence assumption for the same SPODE will not hold to the same extent. To address this issue, a feasible solution is to weight SPODEs by identifying the variation in attribute values of different instances. Consider the correlation between specific root attribute value and class variable, AVWAODE-IG and AVWAODE-KL [19] respectively use the Kullback-Leibler (KL) measure and information gain (IG) to compute the weights.

III. INDEPENDENCE WEIGHTING FOR SUPERPARENT ONE-DEPENDENCE ESTIMATORS

The independence assumption of SPODE may not be suitable for all instances, and that may result in biased estimate of joint probability distribution \( P(x, y) \) and harm AODE’s generalization performance. If the probability distribution of unlabeled instance \( x = \{ x_1, \ldots, x_n \} \) approximates the independence assumption of SPODE\(^a \), then it will be more possible for SPODE\(^a \) to fit \( x \) and then achieve the right classification result. High variance incurred by overfitting labeled training data may results in degradation in classification accuracy, whereas that incurred by overfitting unlabeled testing instance may results in improvement in classification accuracy.

Information theory [22] was first introduced and developed by Shannon to explain the principle behind point-to-point communication and data storing. Entropy function \( H(X) \), conditional entropy function \( H(X|Y) \) and mutual information \( I(X; Y) \) have been applied to measure uncertainty, conditional uncertainty and mutual dependence for BNC learning. These information-theoretic metrics take whole training data as the learning object and consider all possible values of \( X \) and \( Y \). Similar to the basic idea of target learning [21], independence weighting (IW) also takes each unlabeled instance \( x = \{ x_1, \ldots, x_n \} \) as the target or learning object. \( H(x), H(x|y) \) and \( I(x; y) \) are respectively variants of \( H(X), H(X|Y) \) and \( I(X; Y) \). They take specific instance as the learning object and are applied to identify the relationships between attribute values.

Definition 1: Given discrete random variable \( X \) and its possible value \( x \), value-based entropy \( H(x) \) is a function that measures the uncertainty regarding whether the event \( X = x \) happens or not in terms of log likelihood, and it is defined as follows,
\[
H(x) = - \log P(x)
\] (11)

Definition 2: Given discrete random variables \( X \) and \( Y \) and their possible values \( x \) and \( y \), value-based conditional entropy \( H(x|y) \) is defined as follows,
\[
H(x|y) = - \log P(x|y)
\] (12)

Definition 3: The value-based mutual information \( I(x; y) \) measures the mutual dependence between two attribute
values \( x \) and \( y \) and is defined as follows:

\[
I(x; y) = H(x) - H(x|y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{13}
\]

\( \log P(x|x_a, y) \) measures the number of bits needed to describe \( x \) when the values of superparent \( \{x_a, y\} \) are known. If the independence assumption of SPODE is known or Eq.(8) holds, then

\[
\log P(x|x_a, y) = \sum_{i=1}^{n} \log P(x_i|x_a, y). \tag{14}
\]

The log likelihood function \( \mathcal{L}_{\alpha} \), which is defined as follows, is proposed to measure the extent to which the independence assumption of SPODE is suitable to fit the probability distribution of \( (x, y) \).

\[
\mathcal{L}_{\alpha} = \sum_{i=1}^{n} \log P(x_i|x_a, y)
= \sum_{i=1}^{n} \log \frac{P(x_i, x_a, y)}{P(x_a, y)} = \sum_{i=1}^{n} \log \left( \frac{P(x_i, x_a, y)}{P(x_a, y) \cdot P(x, y)} \right) 
= \sum_{i=1}^{n} \log P(x_i|x_a) - \sum_{i=1}^{n} \log P(x_a) 
+ \sum_{i=1}^{n} \log P(x_i|y) 
= -\sum_{i=1}^{n} H(x_a|x_i, y) + \sum_{i=1}^{n} H(x_a|y) - \sum_{i=1}^{n} H(x_i|y) 
= \sum_{i=1}^{n} \{H(x_a) - H(x_a|x_i, y)\} - \sum_{i=1}^{n} H(x_i|y) \tag{15}
\]

Because the class label of testing instance \( x \) is unknown, considering all the possible values of class variable \( Y \), from Eq.(15) the independence assumption implicated in \( x \) can be measured by

\[
\mathcal{W}_\alpha = \sum_y \log P(x|x_a, y) 
= \sum_y \{\sum_{i=1}^{n} I(x_a; x_i, y) - I(x_a; y)\} - \sum_{i=1}^{n} H(x_i|y) 
= \sum_y \{\sum_{i=1}^{n} I(x_a; x_i, y) - I(x_a; y)\} - \sum_{i=1}^{n} \sum_{y} H(x_i|y) 
= I_\alpha - H_Y. \tag{16}
\]

From Eq.(16), \( H_Y \) considers all the attribute values in instance \( x \) and can be regarded as a constant for different SPODEs, thus

\[
I_\alpha = \mathcal{W}_\alpha + H_Y \quad \text{or} \quad I_\alpha \propto \mathcal{W}_\alpha \tag{17}
\]

Algorithm 1 IWAODE

**Input:** Training dataset \( \mathcal{D} \) with \( N \) instances, testing instance \( x = \{x_1, x_2, \ldots, x_n\} \).

**Output:** Class label for \( x \).

1. Generate a three-dimensional table of co-occurrence counts for each pair of attribute values and each class label.
2. According to the attribute values in testing instance \( x \), for each class label \( y_j \), compute \( \hat{P}(y_j|x_i, x_a) \) based on the estimates of \( \hat{P}(y_j|x_a) \) and \( \hat{P}(x_i, y_j, x_a) \).
3. For SPODE that takes \( x_a \) as the super-parent, compute \( I_\alpha \) according to Eq.(16). If \( I_\alpha < 0 \), then set \( I_\alpha = 0 \).
4. Compute \( P_{\text{IWAODE}}(x, y) \) according to Eq.(18) and output the class label by selecting

\[
y^* = \arg \max_y P_{\text{IWAODE}}(x, y).
\]

For simplicity, in practice we choose \( I_\alpha \) rather than \( \mathcal{W}_\alpha \) as the weight of SPODE. For explanatory models (SPODEs in our context) and for the specific unlabeled instance, independence weighting aims to find the trade-off between independence assumptions and model diversity, and thereby achieve good classification performance. The final classifier, IWAODE, calculates the weight associated with each SPODE to linearly combine their probability estimates \( P(x, y) \) of as follows,

\[
P_{\text{IWAODE}}(x, y) = \frac{\sum_{\alpha=1}^{n} I_\alpha \bar{P}(y|x_i, x_a) \prod_{i=1}^{n} \bar{P}(x_i,y,x_a)}{\sum_{\alpha=1}^{n} I_\alpha}.
\tag{18}
\]

Note that, for different unlabeled instances the attribute values may vary greatly and the weight or \( I_\alpha \) may also vary correspondingly, that is, the weights assigned to different SPODEs will adaptively change over different instances. That makes independence weighting more powerful than non-independence weighting. Note that \( \mathcal{W}_\alpha < 0 \) always holds whereas this is not the case for \( I_\alpha \) because \( H(Y) > 0 \) holds in Eq.(17). In this article SPODEs are required to be assigned with positive weights only, when \( I_\alpha < 0 \) holds the weight for SPODE is set to be zero instead. Thus IWAODE has some characteristics of both model selection and model weighting.

The learning procedure of IWAODE as shown follows. Table 1 summarizes the time complexity of each BNC discussed. At training time, AODE needs to generate a three-dimensional table of co-occurrence counts for each pair of attribute values and each class label, and the time complexity is \( O(m^2) \) [14]. At classification time, for estimating the conditional probabilities in Eq.(8) AODE needs to consider each pair of qualified parent and child attributes within each class, and the time complexity is \( O(mn^2) \), where \( t \) is the number of training instances, \( n \) is the number of attributes and \( m \) is the number of class labels. IWAODE has identical training time complexity to AODE because it behaves identically to AODE at training time. At classification time, IWAODE needs to check each attribute-value pairs and all possible
IV. EXPERIMENTS

We compare the performance of our proposed algorithm IWAODE with state-of-the-art classifiers, including semi-naive Bayes algorithms and weighted AODE algorithms. Jiang et al. [18] proposed four metrics for measuring weights, including mutual information, classification accuracy, conditional log likelihood and area under the ROC curve, among which mutual information is the most effective metric. The performance is analyzed in terms of zero-one loss, bias, variance and Nemenyi test on 40 datasets from the UCI machine learning repository [23]. Table 2 presents the details of each dataset, including the number of instances, attributes and class labels.

The following algorithms are compared:
- NB [5], naive Bayes.
- TAN [11], Tree-augmented naive Bayes.
- K2DB [12], k-dependence Bayesian classifier with k = 2.
- AODE [14], Averaged One-dependence Estimators.
- WAODE-MI [18], AODE which uses mutual information (MI) for weighting.
- AVWAODE-IG [19], AODE which uses the information gain (IG) for weighting.
- AVWAODE-KL [19], AODE which uses the Kullback-Leibler (KL) measure for weighting.
- IWAODE.

For fair comparison between IWAODE and other BNCs, e.g., WAODE-MI and AVWAODE-KL, in our experiments the missing values of any attributes are incorporated in probability computation, and they are replaced with the modes and means of the corresponding attribute values from the available data. Quantitative attributes are discretized using Minimum Description Length (MDL) discretization [24]. Probability estimates are smoothed using m-estimation with m = 1 [25]. Consider the “noise” caused by the above data pre-processing steps, we hypothesize that there is sufficient data present for every possible combination of attribute values, and direct estimation of each relevant multi-variate probability will still be reliable for BNC learning.

A. STATISTICS EMPLOYED

Each algorithm is tested on each dataset using 10 rounds of 10-fold cross validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same testing sets. We conducted one-tailed binomial sign test to compare all the related algorithms. The following statistics are employed to interpret the experimental results:
- Win/Draw/Loss (WDL) Record - The WDL record counts the number of datasets for which classifier A performs better, equally well or worse than classifier B on a given metric. The difference is identified as significant if the outcome of a one-tailed binomial sign test is less than 5%.
**TABLE 3.** WDL records for all BNCs in terms of zero-one loss, bias and variance.

| Classifier   | NB | TAN | KDB | AODE | WAODE-MI | AVWAODE-IG | AVWAODE-KL |
|--------------|----|-----|-----|------|----------|------------|------------|
| zero-one loss |    |     |     |      |          |            |            |
| TAN          | 23-6-11 | -   | -   | -    | -        | -          | -          |
| KDB          | 21-5-14 | 18-7-15 | - | -    | -        | -          | -          |
| AODE         | 24-11-5 | 18-12-10 | 22-8-10 | - | -    | -          | -          |
| WAODE-MI     | 28-9-3 | 24-10-6 | 23-9-8 | 8-28-4 | -        | -          | -          |
| AVWAODE-IG   | 23-8-9 | 19-8-13 | 22-4-14 | 13-17-10 | 8-18-14 | -          | -          |
| AVWAODE-KL   | 29-7-4 | 23-11-6 | 27-5-8 | 12-25-3 | 6-30-4 | 12-23-5 | -          |
| IWAODE       | 28-9-3 | 25-10-5 | 27-6-7 | 14-26-0 | 9-31-0 | 15-22-3 | 8-33-1     |
| bias         |    |     |     |      |          |            |            |
| TAN          | 23-7-10 | -   | -   | -    | -        | -          | -          |
| KDB          | 22-9-9 | 18-15-7 | - | -    | -        | -          | -          |
| AODE         | 28-10-2 | 17-11-12 | 16-5-19 | - | -    | -          | -          |
| WAODE-MI     | 29-6-5 | 20-12-8 | 16-8-16 | 7-29-4 | -        | -          | -          |
| AVWAODE-IG   | 26-10-4 | 16-10-14 | 14-7-19 | 9-19-12 | 5-25-10 | -          | -          |
| AVWAODE-KL   | 29-7-4 | 20-14-6 | 15-11-14 | 12-25-3 | 4-31-5 | 13-22-5 | -          |
| IWAODE       | 28-9-3 | 19-12-9 | 17-8-15 | 10-27-3 | 9-27-4 | 15-20-5 | 8-28-4     |
| variance     |    |     |     |      |          |            |            |
| TAN          | 4-3-33 | -   | -   | -    | -        | -          | -          |
| KDB          | 6-2-32 | 6-8-26 | - | -    | -        | -          | -          |
| AODE         | 7-6-27 | 30-4-6 | 30-5-5 | - | -    | -          | -          |
| WAODE-MI     | 8-5-27 | 30-3-7 | 29-6-5 | 6-22-12 | -        | -          | -          |
| AVWAODE-IG   | 4-4-32 | 24-6-10 | 29-3-8 | 2-8-30 | 2-15-23 | -          | -          |
| AVWAODE-KL   | 8-4-28 | 28-5-7 | 30-5-5 | 5-17-18 | 6-18-16 | 14-21-5 | -          |
| IWAODE       | 9-4-27 | 30-6-4 | 30-4-6 | 7-21-12 | 9-23-8 | 20-18-2 | 14-21-5   |

**FIGURE 7.** The comparison results of the Nemenyi test in terms of (a) zero-one loss, (b) bias and (c) variance on 40 datasets. CD = 1.6601.

- **Zero-one Loss** - Suppose that \( y \) and \( \hat{y} \) are the true class label and that predicted by classifier \( A \), respectively. Zero-one loss measures the extent to which classifier \( A \) wrongly predicts the class labels of unlabeled instances, and is defined as
  \[
  \xi(A) = \frac{1}{M} \sum_{i=1}^{M} g(y_i, \hat{y}_i),
  \]
  where \( g(\cdot) \) is a binary function that is zero if its two parameters are identical and one otherwise, and \( M \) denotes the number of unlabeled testing instances. A lower value of zero-one loss indicates the better performance.

- **Bias-variance Decomposition of Zero-one Loss** [16] - Given instance \( x \), bias measures how closely the classifier can describe the decision boundary and is defined as
  \[
  \text{bias}^2 = \frac{1}{2} \sum_{\hat{y}, y \in Y} [P(\hat{y} | x) - P(y | x)]^2.
  \]
  Variance measures the sensitivity of the classifier to variations in the training data and is defined as
  \[
  \text{variance} = \frac{1}{2} \left[ 1 - \sum_{\hat{y} \in Y} P(\hat{y} | x)^2 \right].
  \]
• **Friedman Test** [26] and **Nemenyi Test** [27] - Friedman and Nemenyi Test are effective for comparing multiple classifiers across multiple datasets. Classifiers will be ranked by comparing their classification performance and the Friedman statistic is defined as follows:

\[ X_F^2 = \frac{12N}{k(k+1)} \sum_{j=1}^{k} R_j^2 - 3N(k+1) \] (19)

where \( R_j = \sum_i r_{ij} \) and \( r_{ij} \) is the rank of the \( j \)-th of \( k \) algorithms on the \( i \)-th of \( N \) datasets. The Friedman statistic is distributed according to \( X_F^2 \) with \( k-1 \) and \( (k-1)(N-1) \) degrees of freedom for \( \alpha = 0.05 \). Thus, for any pre-determined level of significance \( \alpha \), the null hypothesis will be rejected if \( X_F^2 > X^2_{\alpha} \). The Nemenyi test is used to further analyze which pairs of algorithms are significantly different in terms of average ranks of the Friedman test. The performance of two classifiers is significantly different if their corresponding average ranks of the Friedman test differ by at least the critical difference (CD). CD is calculated as follows:

\[ CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \] (20)

where \( q_{\alpha} \) are the critical values for \( \alpha = 0.05 \).

**B. ANALYSIS OF THE ZERO-ONE LOSS RESULTS**

Tables 4, 5 and 6 in the Appendix respectively show the experimental results in terms of zero-one loss, bias and variance. WDL records summarizing the relative zero-one loss, bias and variance are shown in Table 3, and Cell\([i; j]\) in each table contains the number of datasets on which classifier on row \( i \) performs better, equally well or worse than the classifier on column \( j \). Only when the outcome of a one-tailed binomial sign test is less than 0.05, the difference between algorithms is supposed to be significant.

Complex topologies of single BNCs can help mitigate the negative effect caused by the unrealistic independence assumption of NB, but at the same time increase the risk of overfitting. Limited number of training instance also increase the risk of biased estimates of conditional probabilities. Ensemble BNCs, e.g., AODE, exhibit excellent generalization ability by using multiple “weak” learners, and large number of dependency relationships in one single BNC can be distributed throughout these committee members. The weighting metric for WAODE-MI measures the relationship between all the possible root attribute values in training data and class labels. In contrast, the other three weighting metrics (i.e., the Kullback-Leibler measure, information gain and log likelihood) measure the relationship between the attribute values in specific testing instance \( x \) and possible class labels. Thus these three weighting metrics can help ensemble BNCs better fit testing instance rather than the training data.

From Table 3 we can see that, high-dependence BNCs enjoy significant advantage over low-dependence BNCs in term of zero-one loss. For example, TAN and K₂DB respectively beat NB on 23 and 21 datasets, and K₂DB performs better than TAN although the advantage is less significant (18 wins and 15 losses). Ensemble BNCs (including AODE and weighted AODEs) all enjoy advantages over single BNCs, and the advantages of weighted AODEs are even more significant. Compared to AODE, WAODE-MI, AVWAODE-KL and AVWAODE-IG respectively wins on 8, 13 and 12 datasets. In contrast, IWAODE significantly outperforms AODE with 14 wins and surprisingly 0 loss.

**C. ANALYSIS OF THE BIAS-VARIANCE DECOMPOSITION**

Bias can measure how closely the BNC is able to describe the decision surfaces for a problem domain. High-dependence BNCs or ensemble BNCs can represent more conditional dependencies among attributes, thus they commonly achieve bias advantage. From Table 3 we can observe that, in terms of bias K₂DB enjoys significant advantage over TAN (18 wins and 7 losses), and TAN over NB (23 wins and 10 losses). AODE performs similarly to K₂DB, and it beats K₂DB on 16 datasets and loses on 19. Weighting by applying different metrics is more capable of reducing bias. WAODE-MI beats AODE on 7 datasets and loses on 4, AVWAODE-KL beats
AODE on 15 datasets and loses on 3. IWAODE significantly outperform the other weighted AODEs on bias reduction.

Variance can measure the sensitivity of the BNCs to variations in the training sample [16]. Due to its changeless topology, NB is ranked the best among all rival single BNCs on reducing variance. For single BNCs, simple topology will help reduce the risk of overfitting and achieve variance advantage. Thus TAN beats K₂DB on 26 datasets and loses on only 6. AODE simply averages every SPODE's probability estimate without incurring sophisticated weighting. This simplicity turns out to be the best scheme in terms of variance reduction, and AODE is ranked the best among all rival ensemble BNCs. Among all the weighted AODEs, IWAODE wins against WAODE-MI, AVWAODE-IG and AVWAODE-KL on reducing variance (WDL records respectively being 9-23-8, 20-18-2 and 14-21-5).

In general, lower bias means that the BNC is able to better fit the training data. However, overfitting may result in greater changes in the topology learned from sample to sample, and hence higher variance [28]. Although weighting reduces the variance advantage of AODE, that can be mitigated by bias reduction. Thus the bias-variance tradeoff is such that bias typically decreases when variance increases. As a result, weighted AODE can significantly outperform
AODE on error reduction. To avoid the appearance of negative weights, for some specific instances the weights assigned to some SPODEs are zero, and thus IWAODE performs an implicit model selection, that can be expected to help reduce the computation overhead and variance to some extent.

### D. FRIEDMAN TEST

With 8 algorithms and 40 datasets, the critical value of $\chi^2$ for $\alpha = 0.05$ with $7 \times 39 = 273$ degrees of freedom is 14.07. The Friedman statistics of experimental results are 48.9 for zero-one loss, 40.57 for bias and 101.3 for variance, which are all larger than 14.07. Hence, the null-hypotheses is rejected. The Critical Difference $CD$ can be calculated by Eq.(20) and is equal to 1.6601. Following the graphical presentation proposed by Demšar [26], we compare the algorithms against each other with the Nemenyi test on zero-one loss, bias and variance in Fig.7. The algorithms are plotted on a vertical line according to their average ranks, and the lower position of an algorithm corresponds to its lower rank and better performance. Ranks are also displayed on a parallel vertical line.

If the difference between the average ranks of two algorithms is less than the $CD$ value, then the two algorithms is connected by a line.

From Fig.7(a), IWAODE achieves the lowest mean zero-one loss rank (3.3375), followed by WAODE-MI (3.525). TAN has lower mean zero-one loss rank than $K_2$-DB.
but not significantly so. The Nemenyi test differentiates ensemble BNCs from single BNCs. Although WAODE-IG ranks the highest among all the weighted AODEs, it enjoys significant zero-one loss advantage over NB, TAN and $K^2$DB. Due to the low power of the Nemenyi test when a large number of algorithms are compared [15], these results differ from those of Section 4.2, in which weighting has proved its effectiveness in zero-one loss reductions in AODE and the application of all four metrics (mutual information, Kullback-Leibler measure, information gain and log likelihood) to AODE significantly improves upon the zero-one loss of AODE.

When bias is compared, From Fig.7(b) there are two clear groups. AODE, Weighted AODEs and $K^2$DB deliver significantly lower mean bias ranks than all the other algorithms. IWAODE and AVWAODE-KL achieve the lowest and second lowest mean bias ranks (3.5125 and 3.7875 respectively). The differences of bias ranks between IWAODE, AVWAODE-KL, AVWAODE-IG and $K^2$DB are relatively large, ranging from 3.5125 to 4.6.
From Fig. 7(c) NB obtains the lowest mean variance rank (2.4875), followed by AODE, WAODE-MI and IWAODE (3.375, 3.6 and 3.6375 respectively). To compute metric for assigning weights, IWAODE, AVWAODE-KL and AVWAODE-IG all use the attribute values from unlabeled testing instance. IW has the largest effect on reducing the variance of AODE, and variance advantages of IWAODE relative to AVWAODE-KL and AVWAODE-IG are very clear. K2DB achieves significantly higher mean variance rank than other BNCs due to its high-dependence topology and single model learning strategy.

V. COMPARISON OF TRAINING AND CLASSIFICATION TIME

All the experiments have been conducted on a desktop computer with an Intel(R) Core(TM) i5-8300H CPU @ 2.3 GHz, 64 bits and 8 GB of RAM memory. BNCs will run on the C++ software specifically designed for classification tasks. Fig. 8 shows the mean training and classification empirical time comparisons of the different out-of-core BNCs relative to IWAODE. Each bar represents the sum of all 40 datasets in a 10-fold cross validation experiment.

At training time, NB and AODE don’t need to learn the network topology and thus their training times are the least among all the BNCs. Among single BNCs, TAN and K2DB need to compute conditional mutual information for learning conditional dependencies between attributes, and K2DB also needs to compute mutual information for sorting attributes. Among ensemble BNCs, WAODE-MI computes mutual information for assigning weights to different SPODEs. AVWAODE-KL and AVWAODE-IG require some additional training time to compute the Kullback-Leibler measure or information gain, and we need to consider any possible root attribute value and all class labels. IWAODE computes weights at classification time rather than training time, so that to compute weights we only need to consider the attribute values in testing instances, thus IWAODE performs identically to AODE at training time. The graph shown in Fig. 8(a) reinforces what the orders of complexity described in Table 1, that is, K2DB requires a bit more time for training, and the training times for AODE and IWAODE are the same.

At classification time, among single BNCs high-dependence BNCs need a bit more time to compute the conditional probabilities than low-dependence BNCs. Among ensemble BNCs, AODE needs to aggregate the estimates of joint probabilities of all qualified SPODEs, and weighted AODEs further need to compute the weighted joint probabilities. As can be seen from Fig. 8(b), AODE is extremely computationally expensive compared to single BNCs, including NB, TAN and K2DB. All the weighted AODEs need more time for classification than AODE. IWAODE considers all attribute values in testing instance to compute weights, whereas AVWAODE-IG and AVWAODE-KL just consider the root attribute values. Because IW performs an implicit model selection and IWAODE doesn’t need to compute the weighted joint probabilities of all the qualified SPODEs, from Fig. 8(b) the classification time difference between IWAODE and its two competitive alternatives, i.e., AVWAODE-IG and AVWAODE-KL, is not significant as supposed.

VI. CONCLUSION

In this article we have introduced weighting schemes to incorporate weights in AODE. Our work has been primarily motivated by the observation that the SPODEs in AODE are treated equally whereas the conditional independence assumptions are often violated to different extents. Weighting scheme is a feasible approach to addressing this issue. In current research, weighting in AODE focuses on the dependency relationship between the root attribute (or root attribute value) and the class variable. We argue that measuring the independence assumptions implicated in different SPODEs and assigning corresponding weights to them provide a natural framework for alleviating the independence assumptions.

Our proposed algorithm, IWAODE, learns weights from specific instance rather than training dataset by applying the independence metric $I_{xy}$. The non-negative characteristic of the weights assigned results in implicit model selection. Our experiments suggest IWAODE has substantially lower bias than AODE at the cost of a small increase in variance. IWAODE retains the simplicity and direct theoretical foundation of AODE while alleviating the limitations of its independence assumptions. Because this scheme performs implicit model selection and hence reduces variance, it might be profitable to explore approaches to perform explicit model selection to aggregate a limited number of SPODEs, and that can retain the low bias advantage while further reducing the high variance.

ACKNOWLEDGMENT

This work is supported by the Research Fund of Higher Education Reform in Jiangsu Province (No.2019JSJG467) and Development Project of Jilin Province (No.20200201281JC), and partially supported by The Provincial Engineering Research Center of Jiangsu and the Innovation and Application of Jiangsu Small Business Industrial Internet Engineering Research Center.

APPENDIX

DETAILED EXPERIMENTAL RESULTS

See Tables 4–6.

REFERENCES

[1] J. Pearl, Causality: Models, Reasoning Inference, vol. 29. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[2] I. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Amsterdam, The Netherlands: Elsevier, 2014.
[3] D. Heckerman, D. Geiger, and D. M. Chickering, “Learning Bayesian networks: The combination of knowledge and statistical data,” Mach. Learn., vol. 20, no. 3, pp. 197–243, Sep. 1995.
[4] D. M. Chickering, Learning Bayesian Networks is NP-Complete. New York, NY, USA, Springer, 1996.
[5] P. Langley, W. Iba, and K. Thompson, “An analysis of Bayesian classifiers,” in Proc. AAAI, vol. 90, 1992, pp. 223–228.
[6] D. D. Lewis, “Naive (Bayes) at forty: The independence assumption in information retrieval,” in Proc. ECML, Berlin, Germany: Springer, 1998, pp. 4–15.
[7] L. Jiang, C. Li, S. Wang, and L. Zhang, “Deep feature weighting for naive bayes and its application to text classification,” Eng. Appl. Artif. Intell., vol. 52, pp. 26–39, Jun. 2016.
[8] L. Jiang, L. Zhang, C. Li, and J. Wu, “A correlation-based feature weighting filter for naive bayes,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 2, pp. 201–213, Feb. 2019.
[9] L. Jiang, Z. Cai, D. Wang, and H. Zhang, “Improving tree augmented naive bayes for class probability estimation,” Knowl.-Based Syst., vol. 26, pp. 239–245, Feb. 2012.
[10] P. Langley and S. Sage, Induction of Selective Bayesian Classifiers. San Mateo, CA, USA: Morgan Kaufmann, 1994.
[11] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” Mach. Learn., vol. 29, no. 2, pp. 131–163, Nov. 1997.
[12] M. Sahami, “Learning limited dependence Bayesian classifiers,” in Proc. 2nd SIGKDD, Portland, OR, USA, 1996, pp. 335–338.
[13] J. Cerequides and R. L. De Mántaras, “Robust Bayesian linear classifier ensembles.” in Proc. ECML, Berlin, Germany: Springer, 2005, pp. 72–83.
[14] G. Webb, J. Boughton, and Z. Wang, “Not so Naive Bayes: Aggregating one-dependence estimators,” Mach. Learn., vol. 58, no. 1, pp. 5–24, 2005.
[15] 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, Apr. 2012.
[16] R. Kohavi and D. H. Wolpert, “Bias plus variance decomposition for zero-one loss functions,” in Proc. ICML, 1996, pp. 275–283.
[17] E. J. Keogh and M. J. Pazzani, “Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches,” in Proc. AILStats, 1999, pp. 1–6.
[18] L. Jiang, H. Zhang, Z. Cai, and D. Wang, “Weighted average of one-dependence estimators,” J. Exp. Theor. Artif. Intell., vol. 24, no. 2, pp. 219–230, 2012.
[19] L. Yu, L. Jiang, D. Wang, and L. Zhang, “Attribute value weighted average of one-dependence estimators,” Entropy, vol. 19, no. 9, pp. 501–518, 2017.
[20] C. Chow and C. Liu, “Approximating discrete probability distributions with dependence trees,” IEEE Trans. Inf. Theory, vol. 14, no. 3, pp. 462–467, May 1968.
[21] L. Wang, S. Chen, and M. Mammadov, “Target learning: A novel framework to mine significant dependencies for unlabeled data,” in Proc. PAKDD, Cham, Switzerland: Springer, 2018, pp. 106–117.
[22] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
[23] P. M. Murphy and D. W. Aha. UCI Repository of Machine Learning Databases. Accessed: Jun. 11, 2019. [Online]. Available: http://www.ics.uci.edu/mlrepository.html
[24] U. Fayyad and K. Irani, “Multi-interval discretization of continuous-valued attributes for classification learning,” in Proc. 13th IJCAI, Chamberry, France, 1993, pp. 1022–1029.
[25] B. Cestnik, “Estimating probabilities: A crucial task in machine learning,” in Proc. ECAI, 1990, pp. 147–149.
[26] J. Demsár, “Statistical comparisons of classifiers over multiple data sets,” J. Mach. Learn. Res., vol. 7, pp. 1–30, Jan. 2006.
[27] P. Nemenyi, “Distribution-free multiple comparisons,” Ph.D. dissertation, Statist. Dept., Princeton Univ., Princeton, NJ, USA, 1963.
[28] F. Zheng and G. I. Webb, “Efficient lazy elimination for averaged one-dependence estimators,” in Proc. 23rd Int. Conf. Mach. Learn. (ICML), 2006, pp. 1113–1120.