PLOS ONE

RESEARCH ARTICLE

RDE: A novel approach to improve the classification performance and expressivity of KDB

Hua Lou[1]*, LiMin Wang[2], DingBo Duan[1], Cheng Yang[1], Musa Mammadov[3]

1 Changzhou College of Information Technology, ChangZhou, China, **2** College of Computer Science and Technology, Jilin University, ChangChun, China, **3** Faculty of Science and Technology, Federation University, Ballarat, Australia

* ccit-louhua@139.com

Citation: Lou H, Wang L, Duan D, Yang C, Mammadov M (2018) RDE: A novel approach to improve the classification performance and expressivity of KDB. PLoS ONE 13(7): e0199822. https://doi.org/10.1371/journal.pone.0199822

Editor: Lars Kaderali, Universitatsmedizin Greifswald, GERMANY

Received: December 21, 2017

Accepted: June 14, 2018

Published: July 23, 2018

Data Availability Statement: The data used in this study are third party and are publicly available from the UCI repository of machine learning databases: http://archive.ics.uci.edu/ml/datasets.html.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Bayesian network classifiers (BNCs) have demonstrated competitive classification performance in a variety of real-world applications. A highly scalable BNC with high expressivity is extremely desirable. This paper proposes Redundant Dependence Elimination (RDE) for improving the classification performance and expressivity of *k*-dependence Bayesian classifier (KDB). To demonstrate the unique characteristics of each case, RDE identifies redundant conditional dependencies and then substitute/remove them. The learned personalized *k*-dependence Bayesian Classifier (PKDB) can achieve high-confidence conditional probabilities, and graphically interpret the dependency relationships between attributes. Two thyroid cancer datasets and four other cancer datasets from the UCI machine learning repository are selected for our experimental study. The experimental results prove the effectiveness of the proposed algorithm in terms of zero-one loss, bias, variance and AUC.

Introduction

Data mining is the analysis step of the "knowledge discovery in databases" process and its goal is the extraction of patterns and knowledge from large amounts of data. During the past decades, statistical models, such as Bayesian network, neural network and support vector machine, have been proposed and applied in many real life applications, e.g. precision medicine. Due to the high prediction performance of these statistical models, researchers would like to gain an understanding of the reasons behind such a prediction, especially when the prediction contradicts their intuition. For example, physicians are typically not only interested in the final prediction, but also like to understand the underlying inference procedure that may help explain why the system makes a certain recommendation. An explanatory, causal and graphical model is more desirable to visualize and mine previously undiscovered knowledge from data [1].

Bayesian network classifiers (BNCs) have long been a popular tool for graphically representing the probabilistic dependencies and inferring under conditions of uncertainty [2–5]. Numerous BNCs (e.g., Naive Bayes (NB) [6], tree augmented Naive Bayes (TAN) [7],

Averaged One-Dependence Estimators (AODE) [8] and k-dependence Bayesian classifier (KDB) [9–11] have been proposed to mine dependency relationships from data. Among them, KDB can generalize to describe any higher degrees of attribute dependence. KDB provides the "average network" to express significant dependencies and this "one size fits all" solution obviously cannot apply to all cases. Patients with similar symptoms may have different kinds of diseases. For example, because of low incidence rate, AIDS (Acquired Immune Deficiency Syndrome) at early stage is often diagnosed as influenza [12]. How to enable person to have "personalized network", which can describe the dependency relationships among specific characteristics or attributes for each case, is still challenging. Local graph structure $KDB_{\mathcal{P}}$ [2] takes each case or unlabeled testing instance $\mathcal{P}$ as a target and can describe local causal relationships implicated. However, the number of conditional dependencies in $KDB_{\mathcal{P}}$ is determined by user-specified parameter k. Some redundant dependencies should be replaced with more meaningful or "personalized" dependencies that only hold in specific instances.

In this paper, a new approach, called Redundant Dependency Elimination (RDE), is proposed to identify redundant conditional dependencies in $KDB_{\mathcal{P}}$ and then substitute/remove them at classification time. The resulting optimized network structure of $KDB_{\mathcal{P}}$, denoted by KDB_O, can increase the confidence level of conditional probabilities. The final personalized classifier, PKDB, is an ensemble of KDBs learned from training data and testing instance respectively. PKDB combines the computational efficiency of classical generative learning with the control of bias/variance trade-off. Two thyroid disease datasets and four other cancer datasets from the UCI machine learning repository are selected for our experimental study. The experimental results show the advantages of PKDB over other classifiers.

Materials and methods

Classifiers

LibSVM [13] and Random forest [14] are introduced in this paper for comparison study. We use Weka's implementations and default settings of Random forest with the exceptions of 20 decision trees. We use Weka's implementations and default settings of LibSVM and performing a "grid-search" on C and γ for the RBF kernel using 5-fold cross-validation. Each pair of (C, γ) is tried ($C = 2^{-5}, 2^{-3}, \cdots, 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, \cdots, 2^{3}$) and the one with the lowest cross-validation zero-one loss is selected. For clarity, the abbreviation of algorithms mentioned above is shown in Table 1.

Table 1. Abbreviation of algorithms introduced in this paper.

Index	Description	Abbreviation
1	Redundant dependency elimination	RDE
2	KDB learned from training data	KDB
3	KDB learned from testing instance $\mathcal{P}$	$KDB_{\mathcal{P}}$
4	Ensemble of KDB and $KDB_{\mathcal{P}}$	AKDB
5	$KDB_{\mathcal{P}}$ optimized by RDE	KDB_O
6	Ensemble of KDB and KDB_O	PKDB
7	A Library for Support Vector Machines	LibSVM
8	Random forest	RF

https://doi.org/10.1371/journal.pone.0199822.t001

Table 2. Description of data sets.

Index	Data set	Case	Att	Class
1	Dis	3772	29	2
2	Hypothyroid	3163	25	2
3	Breast-cancer-w	699	9	2
4	Haberman	306	3	2
5	Heart-disease-c	303	13	2
6	Pima-ind-diabetes	768	8	2

https://doi.org/10.1371/journal.pone.0199822.t002

Data

Six datasets from UCI machine learning repository [15] are selected in this paper for case study. The detailed introduction of these datasets are shown in Table 2, which summarizes the characteristics of each dataset, including the numbers of instances, attributes and classes. For each benchmark dataset, we use MDL discretization [16] to discretize quantitative attributes using 3-bin equal frequency discretization.

Metrics

Zero-one loss is one of the most commonly used metrics to measure the classification performance of a classifier. Zero-one loss can measure how well a classifier correctly identifies or discriminate an unlabeled instance. Let $\mathbf{X}$ and Y be the input and output spaces respectively, and elements $\mathbf{x}$ and y respectively. The zero-one loss function for instance $\mathbf{x}$ is defined as [17]:

$$\xi(\mathbf{x}) = 1 - \delta(y, \hat{y}),$$

where $\delta(y, \hat{y}) = 1$ if $\hat{y} = y$ and zero otherwise, y and $\hat{y}$ are respectively the true class label and predicted label of $\mathbf{x}$. Kohavi and Wolpert presented a bias-variance decomposition of the zero-one loss function [17]. The bias term measures the squared difference between the average output of the target and the algorithm, and it is defined as follows [17]:

$$bias(\mathbf{x}) = \frac{1}{2}\sum_{y' \epsilon Y}[P(y'|\mathbf{x}) - P(y|\mathbf{x})]^2,$$

The variance term measures the sensitivity of the algorithm to the changes in the training set, and it is defined as follows [17]:

$$variance(\mathbf{x}) = \frac{1}{2}\left[1 - \sum_{y' \epsilon Y} P(y'|\mathbf{x})^2\right].$$

In machine learning, the bias-variance tradeoff is a central problem for supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously. High-variance learning methods (e.g., high-dependence BNCs) are usually more complex, enabling them to capture more complex multivariate relationships, but at risk of overfitting to noisy or unrepresentative training data. In contrast, high-bias component of zero-one loss is highly appealing to simpler models that don't tend to overfit, but may underfit their training data, failing to capture important regularities.

The statistical hypothesis test, e.g. Friedman test [18], can test the null hypothesis of no differences between algorithms. Friedman test ranks the algorithms for each data set separately:

the best performing algorithm getting the rank of 1, the second best ranking 2, and so on. In case of ties, average ranks are assigned. The Friedman statistic can be computed as follows [18]:

$$X_F^2 = \frac{12}{Nt(t+1)} \sum_{j=1}^{n} R_j^2 - 3N(t+1), \tag{1}$$

where $R_j = \sum_i r_i^j$ and r_i^j is the rank of the j-th of t algorithms on the i-th of N datasets.

Sensitivity measures the proportion of actual positives that are correctly identified and specificity measures the proportion of actual negatives that are correctly identified. Receiver operating characteristic curve, i.e. ROC curve, is a powerful tool to illustrate the diagnostic ability of a binary classifier by plotting the true positive rate (Sensitivity) against the false positive rate (100-Specificity) for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The ROC curve graphically displays the trade-off between sensitivity and specificity and is useful in assigning the best cut-offs. The area under the ROC curve (AUC) [19] provides a simple numeric measure indicating the performance over the visual comparison of ROC curves.

Bayesian network classifiers

Given class variable Y and a set of discrete attributes $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ (In the following formulas, all variables are assumed to be discrete.) the aim of supervised learning is to predict the discrete class label y of a testing instance $\mathbf{x} = (x_1, \cdots, x_n)$, where x_i is the value of attribute X_i and y is the value of class variable Y. The restricted BNCs, e.g., KDB, model joint probability distribution $P(\mathbf{x}, y)$ according to chain rule, which can be described in the form of a product of a set of conditional probabilities.

$$P(\mathbf{x}, y) = P(y) \prod_{i=1}^{n} P(x_i | \Pi_i, y), \tag{2}$$

where Π_i represents the parent attribute set of X_i.

From the definition of conditional probability, we use the following Formula to classify

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}, y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}, y)}{\sum_y P(\mathbf{x}, y)}. \tag{3}$$

When attribute number n is high and/or data size N is relatively small, it would be difficult to obtain a sufficiently accurate estimate of $P(x_i|\Pi_i, y)$ from the sample frequencies. One popular solution is to restrict the number of parents of each attribute while trying to retain accurate estimate of $P(x_i|\Pi_i, y)$. That is, given attribute subset $\hat{\Pi}_i \subset \Pi_i$, $P(x_i|\hat{\Pi}_i, y) \approx P(x_i|\Pi_i, y)$ holds. Sahami [11] proposed the notion of k-dependence BNC, which allows each attribute X_i to have a maximum of k attribute nodes as parents.

NB is the simplest of the BNCs, assuming that all attributes are independent given the class. There exist no dependency relationships between attributes and thus NB is a 0-dependence BNC. AODE utilizes a restricted class of one-dependence estimators (ODEs) and aggregates the predictions of all qualified estimators within this class. TAN relaxes NB's independence assumption by allowing every attribute to have at most one other attribute as parent. Its basic structure extends the Chow-Liu tree [20] to a maximum spanning tree. The arc or conditional dependence between attributes X_i and X_j is measured by conditional mutual information

(CMI) $I(X_i; X_j|Y)$ given class variable, which is defined as follows [21],

$$I(X_i; X_j|Y) = \sum_{x_i}\sum_{x_j}\sum_{y} P(x_i, x_j, y) log \frac{P(x_i, x_j|y)}{P(x_i|y)P(x_j|y)} \tag{4}$$

KDB further relaxes NB's independence assumption by allowing any attribute X_i to be conditioned on at most k other attributes, i.e., at most k arcs from other attributes to X_i. Unlike TAN, KDB requires to determine the attribute order by comparing the mutual information (MI) $I(X_i; Y)$ between attribute X_i and class Y, which is defined as follows [21],

$$I(X_i; Y) = \sum_{x_i}\sum_{y} P(x_i, y) log \frac{P(x_i, y)}{P(x_i)P(y)} \tag{5}$$

The learning procedures of KDB is described in Algorithm 1.

Algorithm 1: Structure learning of KDB

Input: Training set $\mathcal{T}$, parameter $k = 2$, crosstab $CMI = \{I(X_i, X_j|Y)|1 \leq i \neq j \leq n\}$ (see formula (4)) and vector $MI = \{I(X_i; Y)|1 \leq i \leq n\}$ (see formula (5)).
Output: Network structure $\text{KDB}_{\mathcal{T}} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set.

Let $\mathcal{L}$ be a list of all X_i in descending order of $I(X_i; Y)$.
$\mathcal{V} = \{Y\}$; $\mathcal{E} = \emptyset$;
for $i = 1 \rightarrow n$ **do**
  $\mathcal{V} = \mathcal{V} \cup \mathcal{L}[i]$;
  $\mathcal{E} = \mathcal{E} \cup (Y \rightarrow \mathcal{L}[i])$;
end
for $i = 1 \rightarrow n$ **do**
  $S = \emptyset$;
  $\hat{k} = k$;
  while $(\hat{k} > 0)$ **do**
    $m = \text{argmax}_j\{I(\mathcal{L}[i]; \mathcal{L}[j]|Y) : 1 \leq j < i), j \notin S\}$;
    $\mathcal{E} = \mathcal{E} \cup (\mathcal{L}[m] \rightarrow \mathcal{L}[i])$;
    $\hat{k} = \hat{k} - 1$;
    $S = S \cup \{m\}$;
  end
end
return $\text{KDB}_{\mathcal{T}}$

Algorithm 2: Structure learning of $\text{KDB}_{\mathcal{P}}$

Input: testing instance $\mathcal{P}$, parameter $k = 2$, vector $LMI = \{I(x_i; Y)|1 \leq i \leq n\}$, crosstab $CLMI = \{I(x_i, x_j|Y)|1 \leq i \neq j \leq n\}$ (see formula (6)).
Output: Network structure $\text{KDB}_{\mathcal{P}} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set.

Let $\mathcal{L}$ be a list of all x_i in descending order of $I(x_i; Y)$.
$\mathcal{V} = \{Y\}$; $\mathcal{E} = \emptyset$;
for $i = 1 \rightarrow n$ **do**
  $\mathcal{V} = \mathcal{V} \cup \mathcal{L}[i]$;
  $\mathcal{E} = \mathcal{E} \cup (Y \rightarrow \mathcal{L}[i])$;
end
for $i = 1 \rightarrow n$ **do**
  $S = \emptyset$;

```
9   k̂ = k;
10   while (k̂ > 0) do
11     m = argmax_j{I(L[i]; L[j]|Y) : 1 ≤ j < i), j ∉ S};
12     E = E ∪ (L[m] → L[i]);
13     k̂ = k̂ − 1;
14     S = S ∪ {m};
15   end
16 end
17 return KDB_P
```

KDB can represent the "average knowledge" or "expert knowledge" mined from data, that roughly describes the dependency relationships between different inputs, e.g., the dependency relationship between {*Gender*, *Age*} and *TSH*. However, KDB cannot finely describe the dependency relationships in different patient records, e.g., the relative independency relationship between {*Gender* = "*male*", *Age* = 20} and *TSH* = "*yes*", or the relative dependency relationship between {*Gender* = "*female*", *Age* = 45} and *TSH* = "*yes*". In contrast, $KDB_{\mathcal{P}}$ represents "personalized knowledge" mined from instance $\mathcal{P}$. The "average knowledge" learned from labeled training data and the "personalized knowledge" learned from unlabeled testing instance are complementary in nature. Thus they should be considered simultaneously for classification. To achieve this goal, $KDB_{\mathcal{P}}$ applies the same learning strategy that KDB uses. Given testing instance $\mathcal{P} = (x_1, \cdots, x_n)$, $KDB_{\mathcal{P}}$ sorts attributes by comparing local mutual information (LMI) $I(x_i; x_j|Y)$ and choose appropriate conditional dependencies by comparing conditional local mutual information (CLMI). LMI and CLMI are defined as follows [2],

$$\begin{cases} I(x_i; Y) = \sum_y P(x_i, y) \log \dfrac{P(x_i, y)}{P(y)P(x_i)} \\ I(x_i; x_j|Y) = \sum_y P(x_i, x_j, y) \log \dfrac{P(x_i, x_j|y)}{P(x_i|y)P(x_j|y)} \end{cases} \tag{6}$$

From the viewpoint of information theory, MI or $I(X_i; Y)$ can measure the uncertainty reduction in Y given the information from X_i. The attributes corresponding to greater reduction will get higher rank and added to the network structure in priority. By comparing formulas (5) and (6) we can see that, $I(X_i; Y) = \sum_{x_i} I(x_i; Y)$. MI refers to the average of all possible events, and it is the expected value of LMI over all possible values of X_i. LMI can be used to measure the uncertainty reduction in Y given the information from $X_i = x_i$. Because $I(X_i; X_j|Y) = \sum_{x_i, x_j} I(x_i; x_j|Y)$, we can get similar results that $I(x_i; x_j|Y)$ can measure the conditional dependence between X_i and X_j when they take specific values.

The ensemble of KDB and $KDB_{\mathcal{P}}$, i.e., AKDB [2], has better overall prediction accuracy, on average, than any individual member. KDB and $KDB_{\mathcal{P}}$ apply the same learning strategy whereas model different data spaces (training data and testing instance). It is difficult to judge which output from these two classifiers should be considered in priority. The linear combiner is used for models that output real-valued numbers, so is applicable for BNC. In practice, it is inappropriate to pre-determine the weight of subclassifier. Thus in practice AKDB uses the uniformly rather than nonuniformly weighted average. The ensemble probability estimate is

$$\hat{P}(y|x, \text{AKDB}) = \frac{P(y|x, \text{KDB}) + P(y|x, \text{KDB}_{\mathcal{P}})}{2}. \tag{7}$$

Given m class labels, the class label y^* of unlabeled instance $\mathbf{x}$ corresponds to the highest value of posterior probability of $\hat{P}(y|x, \text{AKDB})$, where $y \in \{y_1, \cdots, y_m\}$, i.e.,

$$y^* = \arg\max \hat{P}(y|x, \text{AKDB}). \tag{8}$$

The classification procedure of AKDB is shown in Algorithm 3.

Algorithm 3: Classification procedure of AKDB

Input: testing instance $\mathcal{P} = (x_1, \cdots, x_n)$, KDB learned from Algorithm 1 and $\text{KDB}_\mathcal{P}$ learned from Algorithm 2.
Output: Class label y^*.
1 Compute the joint probability $P(y, \mathbf{x}|\text{KDB})$ and $P(y, \mathbf{x}|\text{KDB}_\mathcal{P})$ by Formula (2);
2 Compute the conditional probability $P(y|\mathbf{x}, \text{KDB})$ and $P(y|\mathbf{x}, \text{KDB}_\mathcal{P})$ by Formula (3);
3 Compute the conditional probability $P(y|\mathbf{x}, \text{AKDB})$ by Formula (7);
4 Compare and predict the class label y^* for $\mathcal{P}$ by Formula (8);
5 **Return** y^*;

Redundant dependency elimination

Suppose that $\Pi_i = \{X_1, \cdots, X_{i-1}\}$, from the chain rule of mutual information we have [21]

$$\begin{aligned} I(X_i; \Pi_i, Y) \quad = \quad & I(X_i; Y) + I(X_i; X_1|Y) + I(X_i; X_2|X_1, Y) + \\ & \cdots + I(X_i; X_{i-1}|X_1, \cdots, X_{i-2}, Y) \end{aligned} \tag{9}$$

KDB implicitly reduces $I(X_i; X_j|X_1, \cdots, X_{j-1}, Y)$ to $I(X_i; X_j|Y)$ when $j > 1$. The same strategy is also applicable to $\text{KDB}_\mathcal{P}$. Obviously, the dependency relationships between the parent attributes of X_i are neglected, that will inevitably result in estimation bias. For different instances, the dependency relationships may differ. Here, we introduce Pointwise mutual information (PMI) $I(x_i; x_j)$ and Pointwise conditional mutual information (PCMI) $I(x_i; x_k|x_j)$ to address this issue. The definitions of PMI and PCMI are as follows [22],

$$\begin{cases} I(x_i; x_j) = log \dfrac{P(x_i, x_j)}{P(x_i)P(x_j)} \\ \\ I(x_i; x_k|x_j) = log \dfrac{P(x_i, x_k|x_j)}{P(x_i|x_j)P(x_k|x_j)} \end{cases} \tag{10}$$

The dependency relationships in $\text{KDB}_\mathcal{P}$ that are relevant or irrelevant to class labels are respectively measured by formulas (6) and (10), the confidence levels of which are determined by the estimation of probability distributions. The probability distributions have to be estimated from training data before structure learning. For small datasets, the sparsely distributed attribute values make the estimation of lower-order probability estimations much more reliable than that of the higher-order ones. If the probability distributions learned from training data are not reliable, the resulting non-robust classifier will make wrong prediction. That may be the main reason why NB offers competitive performance with high efficiency, strong robustness and loose coupling on some small datasets.

PMI and PCMI refer to single events. Like MI, PMI also follows the chain rule, i.e.,

$$I(x_i; x_1, \cdots, x_{i-1}) = I(x_i; x_1) + I(x_i; x_2|x_1) + \cdots + I(x_i; x_{i-1}|x_1, \cdots, x_{i-2}) \tag{11}$$

In computational linguistics, PMI has been used for finding co-occurrences of words in a text corpus and to approximate the probabilities $P(x)$ and $P(x, y)$ respectively. MI can roughly

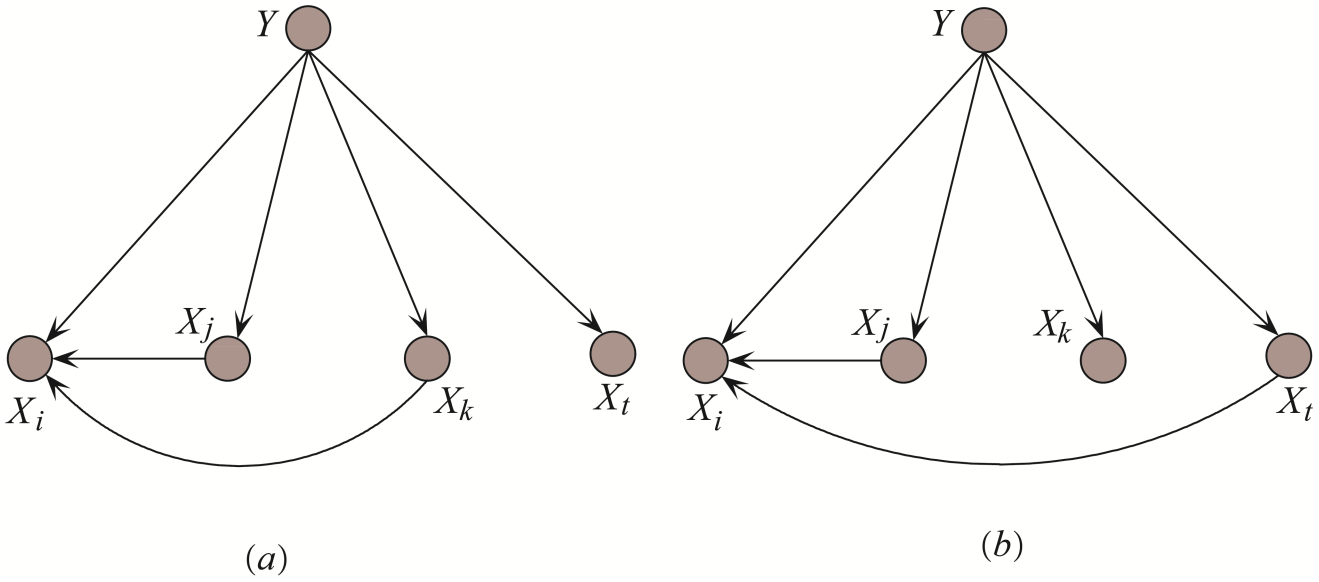

Fig 1. Example: Conditional dependencies between X_i and its parents. (a) X_i has two parent attributes X_j and X_k. (b) Parent attribute X_k is substituted with X_t.

https://doi.org/10.1371/journal.pone.0199822.g001

measure the dependency relationship between the associated variables, but cannot measure the inherent relational mapping between specific variable values. Given two attributes X_i, X_j, each having two values and $<X_i, X_j>$ = {(1, 1), (1, 2), (2, 2)} for example, obviously when X_i = 1 the uncertainty of X_j reaches a maximum, whereas when X_i = 2 the uncertainty of X_j is reduced to zero. In practice, it may be the case that certain values are more significant than others, or that certain patterns of association are more semantically important than others. Further, it is desirable to obtain reasonable causal relationships for causality analysis rather than a simple classification result. Considering an example of a substructure shown in Fig 1(a). Corresponding training data is presented in Table 3. In this example, X_i has two parents X_j, X_k and its conditional probability is $P(x_i|x_j, x_k, y)$.

KDB or KDB$_p$ just consider the conditional dependence between X_i and its parents, and the relationships among parents are neglected, that may not help to increase the confidence level of $P(x_i|\Pi_i, y)$ or reduce the uncertainty of X_i when it takes specific values. For testing instance $\mathcal{P}$, its class label is unknown thus Redundant Dependency Elimination (RDE) just considers the dependency relationships between attribute values. For example, given {$X_i = b$,

Table 3. An example of training data with four attributes, in which the mapping relationships between {X_i, X_j, X_k} are shown.

X_i	X_j	X_k	X_t
a	*c*	*e*	*b*
b	*d*	*e*	*b*
b	*d*	*e*	*c*
b	*d*	*e*	*c*
a	*d*	*e*	*c*
a	*c*	*f*	*d*

https://doi.org/10.1371/journal.pone.0199822.t003

$X_j = d, X_k = e\}$ in Table 3, from the chain rule of PMI we will have

$$\begin{aligned} I(x_i; x_j, x_k) &= I(x_i; x_j) + I(x_i; x_k|x_j) = \log \frac{P(b,d)}{P(b)P(d)} + \log \frac{P(b,e|d)}{P(b|d)P(e|d)} \\ &= \log \frac{P(b,d)}{P(b)P(d)} + \log \frac{P(b,d,e)P(d)}{P(b,d)P(d,e)} \\ &= \log \frac{\frac{1}{2}}{\frac{1}{2}\cdot\frac{4}{6}} + \log \frac{\frac{1}{2}\cdot\frac{4}{6}}{\frac{1}{2}\cdot\frac{4}{6}} \\ &= \log \frac{6}{4} + 0 = \log \frac{6}{4} \end{aligned} \tag{12}$$

Thus $I(x_i; x_j, x_k) = I(x_i; x_j)$, i.e., x_k does not provide any extra valuable information to reduce the uncertainty of x_i. To further increase the conditional probability of x_i, we should select another attribute value, e.g., x_t, to take the place of x_k. If $I(x_i; x_t|x_j) > 0$, then

$$I(x_i; x_j, x_t) - I(x_i; x_j, x_k) = [I(x_i; x_j) + I(x_i; x_t|x_j)] - I(x_i; x_j) = I(x_i; x_t|x_j) > 0 \tag{13}$$

Thus the larger the difference is, the more appropriate X_t is as the parent of X_i. If the attributes are sorted by comparing $I(x_i; Y)$ and the resulting order is $\{x_1, \cdots, x_n\}$, then x_i can select at most k parents from $i - 1$ attributes that ranks higher. Suppose that its parents are sorted by comparing $I(x_i; x_j|Y)(j < i)$ and the order is $\{\hat{x}_1, \cdots, \hat{x}_{i-1}\}$, RDE first operates by iteratively identifying redundant parents of each attribute. It uses the criterion

$$\frac{I(x_i; \hat{x}_j|\hat{x}_1)}{I(x_i; \hat{x}_1)} \geq \delta \tag{14}$$

to infer that except the information $\hat{x}_1$ provides to x_i, $\hat{x}_j$ can provide extra information to x_i, where $\hat{X}_j \in \Pi_i$ and $1 < j \leq i - 1$, δ is a minimum redundancy ratio. If there exist attribute values that make formula (14) hold, then an appropriate parent of x_i should be selected from them. This process is terminated if there is no redundancy or no substituted attribute available. We keep the attribute value with the smallest index and disregard the other attribute values. For instance, if $\hat{x}_2, \hat{x}_3$ and $\hat{x}_4$ hold for formula (14), we only take $\hat{x}_2$ as the parent of x_i.

Starting from the basic network structure learned from testing instance, KDB_O repairs "harmful" interdependencies by applying RDE to remove highly correlated attribute values in classification time. Note that attribute selection approaches, such as Backwards sequential elimination (BSE, [23, 24]), simply remove attributes to achieve zero-one loss improvement. BSE operates by iteratively removing successive attributes until no zero-one loss improvement. According to Formula 2, attribute X_i can have at most $i - 1$ parents, i.e., there exists $i - 1$ conditional dependencies between X_i and its parents. If X_i is removed from Bayesian network structure, then $i - 1$ conditional dependencies will be implicitly removed correspondingly. That will result in great change in network structure and classification bias. In contrast, RDE retains all attributes and resolve such interdependencies with much more flexible strategy and finer tuning, as for some test instances one conditional dependence may be identified as redundant and then substituted or removed, for other test instances it may hold.

One effective way of resolving the trade-off between bias and variance is to use ensemble learning [9, 25]. For example, boosting combines many "weak" (high bias) models in an ensemble that has lower bias than the individual models, while bagging combines "strong" learners in a way that reduces their variance. KDB and KDB_O are both "strong" learners. KDB takes training set as a target and build general BNC for it. KDB_O takes testing instance $\mathcal{P}$ as a target and build a specific BNC for $\mathcal{P}$. In contrast to KDB, KDB_O is defined by the conditional

dependencies at the attribute values in $\mathcal{P}$. Obviously, for different testing instances, KDB remains the same while KDB_O may differ greatly. RDE identifies and then substitutes/removes the redundant dependencies in $KDB_{\mathcal{P}}$, that will make the conditional dependencies in KDB_O much more reasonable.

The final model, PKDB, is an ensemble of KDB and KDB_O. The ensemble probability estimate for PKDB is

$$\hat{P}(y|x, \text{PKDB}) = \frac{P(y|x, \text{KDB}) + P(y|x, \text{KDB}_O)}{2}.$$

PKDB can represent arbitrary k-dependence relationships. It seems that PKDB with higher degree of attribute dependence will more closely fit the training data and can achieve better generalization performance than those with lower degree of attribute dependence. However, higher degree of attribute dependence needs more training instances to ensure more accurate estimation of conditional probability. From Table 2, the thyroid disease datasets for experimental study contain relatively small number (< 3800) of instances but large number (≥ 25) of attributes. To make resulting algorithm combine the computational efficiency of classical generative learning with the control of bias/variance trade-off, in the following discussion we restrict PKDB to be 2-dependence, i.e., $k = 2$, as used in [2]. Since attribute X_i can have k parent attributes with higher ranks, the problem of redundant dependency arises when $i \geq k + 2$. The detailed learning procedure of PKDB is presented in Algorithm 4.

Algorithm 4: Redundant Dependency Elimination for KDB_O when $k = 2$

Input: Network structure $KDB_{\mathcal{P}}$, parameter k, testing instance $\mathcal{P}$.
Output: KDB_O, network structure after applying RDE.
1 Transform $KDB_{\mathcal{P}}$ to a set of children-parent pairs $\{x_1, \Pi_1\} \cdots, \{x_n, \Pi_n\}$.
2 Let $\mathcal{L}$ be a list of all x_i in descending order of $I(x_i; Y)$.
3 **for** $i = k + 2 \rightarrow n$ **do**
4 Let $\mathcal{L}'$ be a list of all $x_j (x_j \in \Pi_i)$ in descending order of $I(x_i; x_j | Y)$;
5 $\Pi_i = \{\mathcal{L}'[1]\}$;
6 **for** $j = 2 \rightarrow i - 1$ **do**
7 **if** $(I(\mathcal{L}[i]; \mathcal{L}'[j] | \mathcal{L}'[1]) \geq \delta \cdot I(\mathcal{L}[i]; \mathcal{L}'[1]))$ (see formula (14)) **then**
8 $\Pi_i = \{\mathcal{L}'[1], \mathcal{L}'[j]\}$;
9 **end**
10 **end**
11 **end**
12 Transform revised children-parent pairs $\{x_1, \Pi_1\} \cdots, \{x_n, \Pi_n\}$ to KDB_O.
13 **return** KDB_O

During training PKDB generates a three-dimensional table of co-occurrence counts for each pair of attribute values and each class value to estimate the probabilities $P(y)$, $P(x_i, y)$, $P(x_i, x_j, y)$, $P(x_i, x_j)$ and $P(x_i, x_j, x_k)$. KDB requires $O(Nm(nv)^2)$ time (dominated by calculating CMI) [11] to build the network structure, where v is the average number of discrete values that an attribute may take. The basic structure of KDB_O only considers the attribute values in testing instance and thus requires $O(Nmn^2)$ time. RDE requires $O(Nn^2)$ time to calculate PCMI, then an extra pass is needed to perform identification and then substitute/remove redundant conditional dependencies. The final time complexity for building KDB_O is $O(Nmn^2) + O(Nn^3)$. The time complexities of classifying a single instance for KDB and KDB_O are the same, $O(mnk)$.

Results

The experimental system is implemented in C++. The experiments are conducted on a desktop computer with an Intel(R) Core(TM) i5-7200 CPU @3.20GHz, 64 bits and 12,288 MB of memory. For the BNCs to be compared, 10-fold cross validation is applied to obtain an accurate estimation of the average performance. For each fold, leave-one-out cross validation zero-one loss [26] [27] is used as selection criterion to determine δ in Formula (14). Table 4 summarizes the experimental results in terms of zero-one loss, bias, variance and AUC. The Friedman statistic is distributed according to X_F^2 with $t-1$ degrees of freedom. Thus, for any pre-determined level of significance α, the null hypothesis will be rejected if $X_F^2 > X_F^\alpha$. The critical value of X_F^α for $\alpha = 0.05$ with seven degrees of freedom is 14.07. The Friedman statistic of zero-one loss in Table 4 is 15.32, which is larger than 14.07. Hence, the null-hypotheses is rejected and these classifiers are different.

Quinlan believed that the two relatively large datasets, i.e. `Dis` and `Hypothyroid`, have been corrupted [28] and many missing values exist (6064 missing values in dataset `Dis` and 5329 missing values in dataset `Hypothyroid`). When we substitute these missing values with a specific value, i.e., "?" or unknown, noise is artificially introduced and the performance of learned classifier may be degraded. For the other four small datasets with less than 800 instances, the training data provided only accounts for a small portion of the full dataset. Thus the estimation of conditional probability will be of low-confidence. Relatively simple structure resulted from underfitting rather than overfitting may help to improve the classification performance of learning algorithm. From the experimental results of zero-one loss in Table 4 we can see that, classifiers with complex structure don't necessarily enjoy significant advantage over classifiers with simple structure. For example, KDB, LibSVM and RF perform poorer than NB on datasets `Breast-cancer-w` and `Heart-disease-c`. However, $KDB_{\mathcal{P}}$ provides an effective way to learn high-confidence dependency relationships implicated in testing instance. RDE can remove the redundant dependency relationships that are irrelevant to class label and add high-confidence conditional dependencies. The negative effect caused by noise and insufficient data will be mitigated to some extent. AKDB performs better than KDB. PKDB even performs the best among all classifiers in terms of zero-one loss.

We then clarify from the viewpoint of bias-variance decomposition. The experimental results of variance are reasonable that AODE achieves higher variance than NB because of its complex structure. However, AODE achieves higher bias on dataset `Dis`, which means underfitting to some extent. Since AODE indiscriminately represents all $29^*28 = 812$ conditional dependencies, some weak dependencies may represent a large noise component in the training set and counteract the effect the strong dependencies, making it underfit dataset `Dis` and its prediction less accurate than NB. For dataset `Hypothyroid` AODE only needs to represent $25^*24 = 600$ conditional dependencies and negative effect of weak dependencies can be mitigated. When $k = 2$, KDB can represent $0 + 1 + 2 \cdots + 2 = 49$ conditional dependencies (as shown in Fig 2 whereas TAN only needs to represent 28 conditional dependencies. Thus KDB achieves higher variance since it fits training set well even there exists noise. As a result, the KDB does not fit the testing instance much better than TAN. Noisy training data will reduce the confidence level of the classification model. For classifiers learned from the other four small datasests, overfitting is almost inevitable. KDB, LibSVM and RF perform poorer than NB on datasets `Breast-cancer-w` and `Heart-disease-c` in terms of variance. How to reduce variance is a crucial point for improving classification accuracy. RDE helps to mitigate the negative effect of overfitting, thus the variance for PKDB is always lower than that for AKDB and KDB.

Table 4. The comparison of classification performance between classifiers in terms of zero-one loss, bias, variance and AUC.

	Dataset	NB	AODE[17]	TAN[16]	KDB[18]	AKDB[11]	PKDB	LibSVM	Random Forest
Zero-one loss	Dis	0.0234 ± 0.0127	0.0241 ± 0.0126	0.0198 ± 0.0101	0.0202 ± 0.0105	0.0193 ± 0.0092	0.0182 ± 0.0087	0.0215 ± 0.0106	0.0185 ± 0.0065
	Hypothyroid	0.0147 ± 0.0095	0.0128 ± 0.0082	0.0138 ± 0.0071	0.0122 ± 0.0062	0.0112 ± 0.0065	0.0093 ± 0.0076	0.0165 ± 0.0054	0.0146 ± 0.0093
	Breast-cancer-w	0.0255 ± 0.0223	0.0383 ± 0.0248	0.0534 ± 0.0204	0.0845 ± 0.0248	0.0531 ± 0.0212	0.0482 ± 0.0105	0.0406 ± 0.0126	0.0395 ± 0.0087
	Haberman	0.2856 ± 0.1052	0.3337 ± 0.0835	0.3812 ± 0.0975	0.3345 ± 0.1025	0.3054 ± 0.0924	0.2656 ± 0.0987	0.2765 ± 0.1541	0.3106 ± 0.0912
	Heart-disease-c	0.1751 ± 0.0694	0.1668 ± 0.0811	0.1876 ± 0.0863	0.2038 ± 0.0933	0.1848 ± 0.0948	0.1732 ± 0.0672	0.2056 ± 0.0974	0.2116 ± 0.0104
	Pima-ind-diabetes	0.2568 ± 0.0745	0.2303 ± 0.0677	0.2850 ± 0.0755	0.3170 ± 0.0586	0.2425 ± 0.0649	0.2178 ± 0.0765	0.2552 ± 0.0835	0.2713 ± 0.0953
Bias	Dis	0.0165 ± 0.0092	0.0174 ± 0.0086	0.0193 ± 0.0058	0.0191 ± 0.0062	0.0191 ± 0.0082	0.0181 ± 0.0063	0.0157 ± 0.0078	0.0177 ± 0.0037
	Hypothyroid	0.0116 ± 0.0101	0.0094 ± 0.0092	0.0104 ± 0.0082	0.0096 ± 0.0065	0.0082 ± 0.0046	0.0075 ± 0.0041	0.0115 ± 0.0035	0.0078 ± 0.0014
	Breast-cancer-w	0.0187 ± 0.0091	0.0243 ± 0.0104	0.0143 ± 0.0052	0.0449 ± 0.0102	0.0196 ± 0.0052	0.0201 ± 0.0029	0.0136 ± 0.0058	0.0181 ± 0.0036
	Haberman	0.2332 ± 0.1118	0.2375 ± 0.1091	0.2298 ± 0.0962	0.2301 ± 0.0765	0.2127 ± 0.1053	0.2010 ± 0.1053	0.2236 ± 0.1021	0.2235 ± 0.0924
	Heart-disease-c	0.1368 ± 0.0946	0.1414 ± 0.0652	0.1426 ± 0.0118	0.1697 ± 0.0763	0.1656 ± 0.0932	0.1602 ± 0.0842	0.1563 ± 0.0256	0.1769 ± 0.0245
	Pima-ind-diabetes	0.1957 ± 0.1043	0.1935 ± 0.0973	0.1917 ± 0.0972	0.1944 ± 0.0916	0.1964 ± 0.1096	0.2053 ± 0.0725	0.2015 ± 0.0635	0.2274 ± 0.0626
Variance	Dis	0.0069 ± 0.0033	0.0071 ± 0.0027	0.0005 ± 0.0003	0.0011 ± 0.0003	0.0002 ± 0.0001	0.0002 ± 0.0001	0.0082 ± 0.0002	0.0002 ± 0.0001
	Hypothyroid	0.0031 ± 0.0015	0.0034 ± 0.0012	0.0034 ± 0.0009	0.0024 ± 0.0008	0.0025 ± 0.0009	0.0021 ± 0.0003	0.0057 ± 0.0021	0.0087 ± 0.0002
	Breast-cancer-w	0.0014 ± 0.0003	0.0118 ± 0.0081	0.0207 ± 0.0006	0.0504 ± 0.0102	0.0255 ± 0.0093	0.0248 ± 0.0072	0.0213 ± 0.0026	0.0204 ± 0.0026
	Haberman	0.0325 ± 0.0103	0.0312 ± 0.0121	0.0317 ± 0.0093	0.0333 ± 0.0101	0.0322 ± 0.0112	0.0309 ± 0.0097	0.0211 ± 0.0082	0.0324 ± 0.0055
	Heart-disease-c	0.0443 ± 0.0103	0.0463 ± 0.0101	0.0497 ± 0.0092	0.0914 ± 0.0082	0.0744 ± 0.0084	0.0718 ± 0.0045	0.0702 ± 0.0045	0.0823 ± 0.0029
	Pima-ind-diabetes	0.0715 ± 0.0110	0.0729 ± 0.0103	0.0751 ± 0.0097	0.0689 ± 0.0101	0.0661 ± 0.0082	0.0525 ± 0.0064	0.0516 ± 0.0095	0.0689 ± 0.0073
AUC	Dis	0.9828 ± 0.2112	0.9773 ± 0.1066	0.9933 ± 0.0924	0.9912 ± 0.1076	0.9905 ± 0.0824	0.9998 ± 0.1142	0.6228 ± 0.1043	0.9666 ± 0.0972
	Hypothyroid	0.7235 ± 0.1237	0.9756 ± 0.2062	0.9807 ± 0.2112	0.9818 ± 0.1637	0.9396 ± 0.1253	0.9887 ± 0.1512	0.7482 ± 0.0624	0.9882 ± 0.1125
	Breast-cancer-w	0.5150 ± 0.0827	0.4572 ± 0.1213	0.7823 ± 0.1046	0.8238 ± 0.1237	0.8480 ± 0.0975	0.9157 ± 0.0882	0.9557 ± 0.1074	0.9877 ± 0.1067
	Haberman	0.7823 ± 0.1162	0.8124 ± 0.0912	0.8637 ± 0.0824	0.8831 ± 0.0967	0.9237 ± 0.1057	0.9742 ± 0.1169	0.5234 ± 0.0824	0.8632 ± 0.0913
	Heart-disease-c	0.5332 ± 0.0474	0.5134 ± 0.0421	0.4528 ± 0.0474	0.4425 ± 0.0518	0.4724 ± 0.0436	0.5153 ± 0.0517	0.8224 ± 0.0613	0.8661 ± 0.1092
	Pima-ind-diabetes	0.6793 ± 0.0626	0.7141 ± 0.0631	0.7367 ± 0.0732	0.7445 ± 0.0974	0.7632 ± 0.0635	0.8124 ± 0.0842	0.7225 ± 0.0623	0.8081 ± 0.1064

https://doi.org/10.1371/journal.pone.0199822.t004

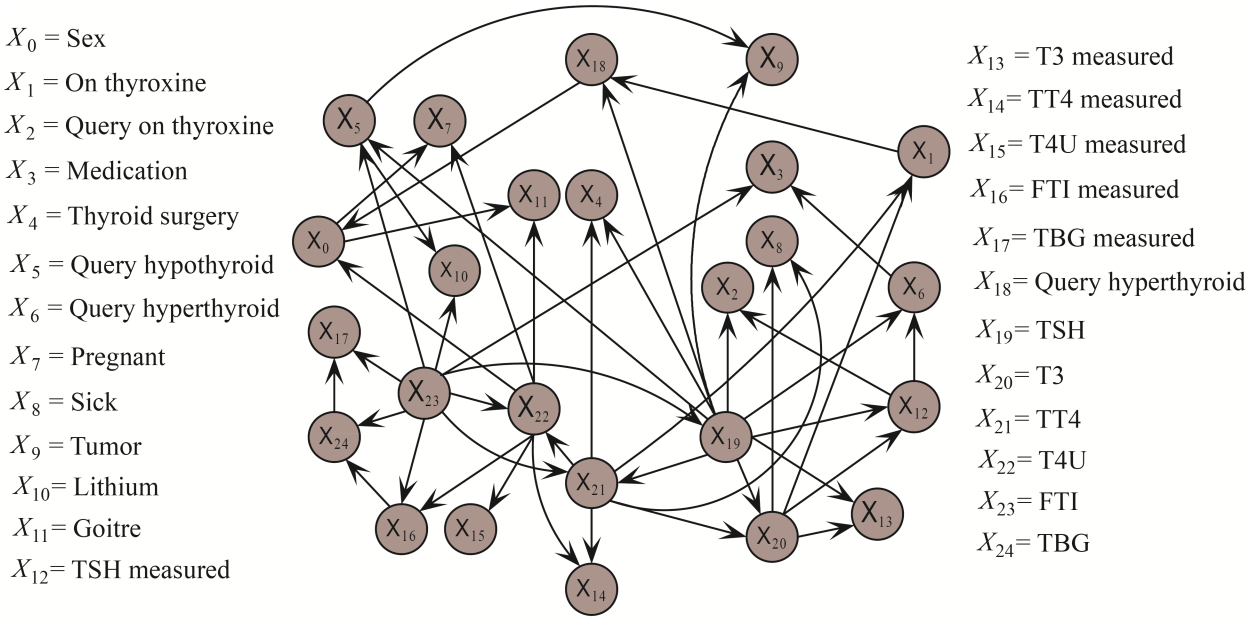

Fig 2. The network structure of KDB($k = 2$) on dataset `Hypothyroid`. Class variable Y is not included for simplicity. Only conditional dependencies between attributes are shown.

https://doi.org/10.1371/journal.pone.0199822.g002

AUC is often used to evaluate the classification performance while dealing with imbalanced data. From Table 4 we can see that, TAN and KDB perform better than NB more often than not on small datasets. That indicates although the negative effect caused by overfitting may reduce the classification accuracy, the dependency relationships implicated will help to improve the the discriminatory power of BNCs. The definitions of LMI and CLMI considers all possible values of class variable, thus $KDB_{\mathcal{P}}$ cannot overfit the given testing instance $\mathcal{P}$, but provides a possible dependence tree structure to describe the relationships among attribute values in $\mathcal{P}$. The advantage of PKDB over other classifiers in AUC is especially obvious on datasets `Dis` and `Hypothyroid`. In contrast, AKDB also uses the personalized $KDB_{\mathcal{P}}$, it performs much worse. This can be attributed to the low-confidence dependency relationships mined from these small datasets. LibSVM performs poorer on datasets `Dis` and `Hypothyroid` but better on the other four small datasets. RF demonstrates significant robustness while dealing with relatively large or small datasets.

Discussion

Doctors may need to determine if blood tests are necessary for patients due to their respective risk factors, e.g., family history of goitres, Gender or Age. By computing LMI, CLMI from the local perspective, KDB_O, which learns from individual testing instance, is obviously an example of learners for precision medicine. PKDB can utilize the information provided by the training set and testing instances with the help of the aggregating mechanism. To prove this, we take two cases for example from `Hypothyroid` dataset, which take different class labels. The

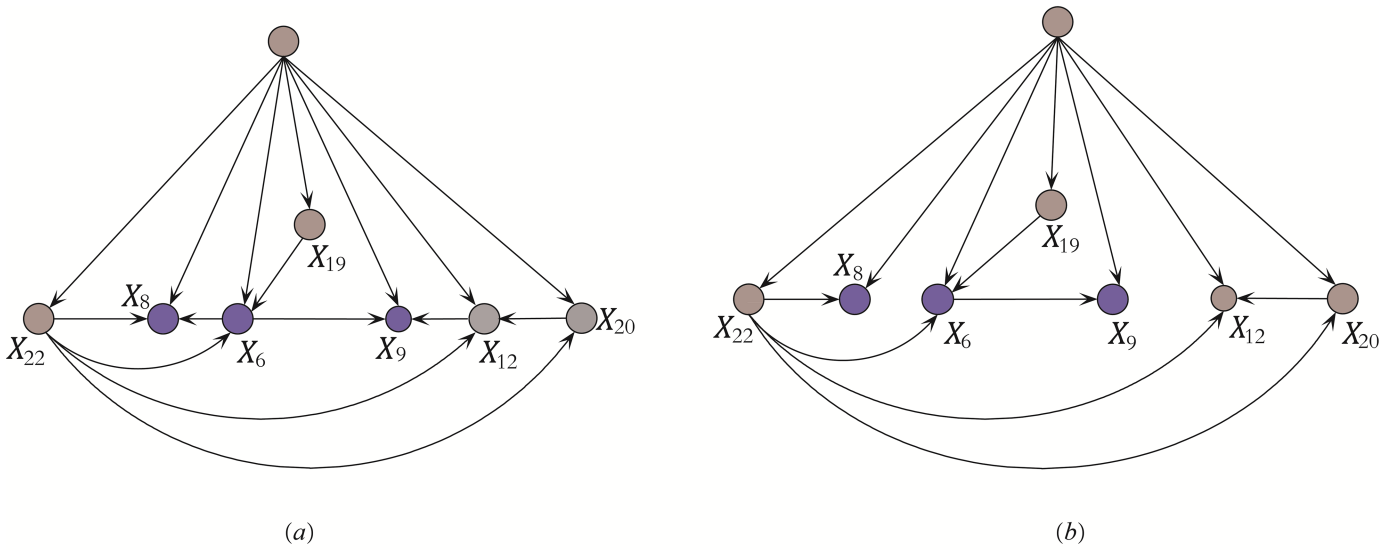

Fig 3. The substructures of KDB$_\mathcal{P}$ (a) and KDB$_O$ (b) for {X_8, X_6, X_9} learned from *Case*1 = (x_{19} = 43, x_{23} = 47, x_{22} = 1.26, x_{20} = 2, x_{21} = 59, x_{12} = *y*, x_{13} = *y*, x_6 = *t*, x_{14} = *y*, x_{15} = *y*, x_{16} = *y*, x_5 = *f*, x_{24} = ?, x_{17} = *n*, x_1 = *f*, x_0 = *F*, x_{18} = 28, x_4 = *f*, x_2 = *f*, x_8 = *f*, x_{11} = *f*, x_7 = *f*, x_9 = *t*, x_3 = *f*, x_{10} = *f*). The arcs X_6(Query hyperthyroid) → X_8(Sick), X_{12}(TSH measured) → X_9(Tumor) in (a) are identified as redundant and removed. As shown in (b), no more attributes with higher ranks are considered as possible parents of X_8 and X_9.

https://doi.org/10.1371/journal.pone.0199822.g003

first case that is diagnozed as "hypothyroid" is shown as follows

$$\begin{aligned} Case1 \quad = \quad & (x_{19} = 43, x_{23} = 47, x_{22} = 1.26, x_{20} = 2, x_{21} = 59, x_{12} = y, x_{13} = y, x_6 = t, \\ & x_{14} = y, x_{15} = y, x_{16} = y, x_5 = f, x_{24} = ?, x_{17} = n, x_1 = f, x_0 = F, x_{18} = 28, \\ & x_4 = f, x_2 = f, x_8 = f, x_{11} = f, x_7 = f, x_9 = t, x_3 = f, x_{10} = f) \end{aligned} \quad (15)$$

where '?' is used to denote a value that is missing or unknown. The attribute values in *case* 1 have been sorted by comparing $I(x_i; Y)$. Among them, x_{19} or TSH ranks the highest, thus the level of TSH is closely related to some definite results and further tests will be needed. The full network structure with 25 attributes are too complex (47 arcs or conditional dependencies) to explain, so we just select one substructure to clarify. The conditional dependencies in KDB$_\mathcal{P}$ and KDB, which focus on attributes {X_8, X_6, X_9}, are respectively shown in Figs 3(a) and 4. In Fig 3(a), the testing result of X_{22}(T4U) can explain why the patient does not feel sick($X_8 = f$), thus X_6(query on hyperthyroid) does not provide valuable information. The arc $X_6 \rightarrow X_8$ is removed. By comparing KDB shown in Fig 4 and KDB$_O$ shown in Fig 3(b), the limitation of KDB in precise representation is obvious. Hyperthyroidism is a condition in which thyroid gland produces too much of the hormone thyroxine. One symptom for hyperthyroid is an enlarged thyroid gland, which may appear as a swelling at the base of one's neck. It is reasonable in Fig 3(b) that X_9(tumor) is related to X_6(query on hyperthyroid) whereas in Fig 4 X_9(tumor) is related to X_5(query on hypothyroid). To judge the possibility of hypothyroidism, blood tests (including TSH(X_{12}), T3(X_{20}) and T4U(X_{22})) are needed. The close relationships can be clearly seen in Fig 3.

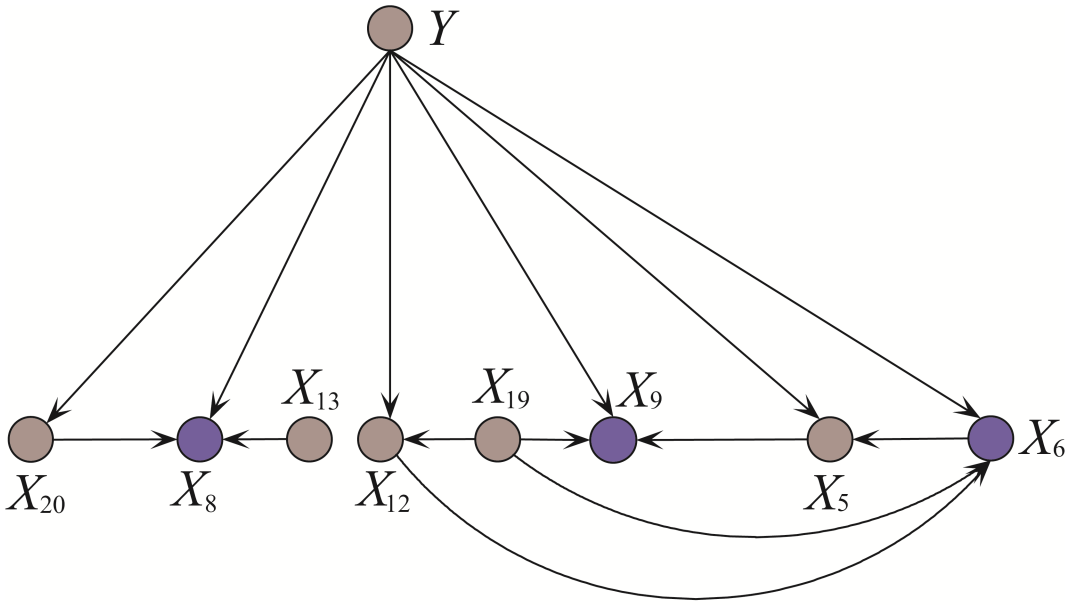

Fig 4. The substructure of KDB for $\{X_8, X_6, X_9\}$ learned from training set. The arc X_5(TSH measured) → X_9(Tumor) is not reasonable when $X_9 = t$ (i.e., 'true').

https://doi.org/10.1371/journal.pone.0199822.g004

The detail of the second instance that is diagnozed as "negative" is shown as follows,

$$\begin{aligned} Case2 \quad = \quad & (x_{23} = 51, x_{21} = 37, x_{20} = 0.5, x_{19} = 9.7, x_{22} = 0.72, x_{13} = y, x_{12} = y, x_1 = t, \\ & x_{14} = y, x_{15} = y, x_{16} = y, x_5 = f, x_0 = F, x_{24} = ?, x_{17} = n, x_4 = f, x_{18} = 46, \\ & x_6 = f, x_8 = f, x_2 = f, x_{11} = f, x_9 = f, x_7 = f, x_3 = f, x_{10} = f) \end{aligned} \tag{16}$$

The conditional dependencies in $KDB_{\mathcal{P}}$ and KDB, which focus on attributes $\{X_1, X_{15}, X_{16}\}$, are respectively shown in Figs 5(a) and 6. The information implicated in some attribute values may overlap or even cover that in other attribute values. For example, "TSH measured = *y*" is a premise of "TSH = 4.6". "Sex = *F*" is a premise of "Pregnant = *t*". Although there exist strong dependencies between these attribute values and they may appear simultaneously as the co-parents of some attributes, this kind of dependencies are redundant and should be substituted. The arc $X_{12} \rightarrow X_1$ is removed from Fig 5(a) and we should find another parent for X_1 as shown in Fig 5(b). To provide accurate diagnosis for hypothyroid, the blood tests of TT4 and FTI are always used simultaneously. Thus the arc $X_{15} \rightarrow X_{16}$ is also redundant and should be removed. The limitation of KDB in scalability is obvious. As shown in Fig 6, the value of X_{13}(T3 measured) is a premise of the value of X_{20}(T3). When they appear as the co-parents of some other attribute, e.g., X_1, the conditional probability $P(x_1|x_{13}, x_{20}, y)$ will approximate the estimate of $P(x_1|x_{20}, y)$. X_{13}(T3 measured) cannot provide any valuable information to X_1.

Conclusion and future work

$KDB_{\mathcal{P}}$ takes instance $\mathcal{P}$ as the target and its network structure describes the dependency relationships in $\mathcal{P}$. Because of the computational overhead, only a limited number of dependencies, which are determined by parameter *k*, can be described by $KDB_{\mathcal{P}}$. The proposed approach, RDE, is a filter that transforms the testing instance to substitute these redundant

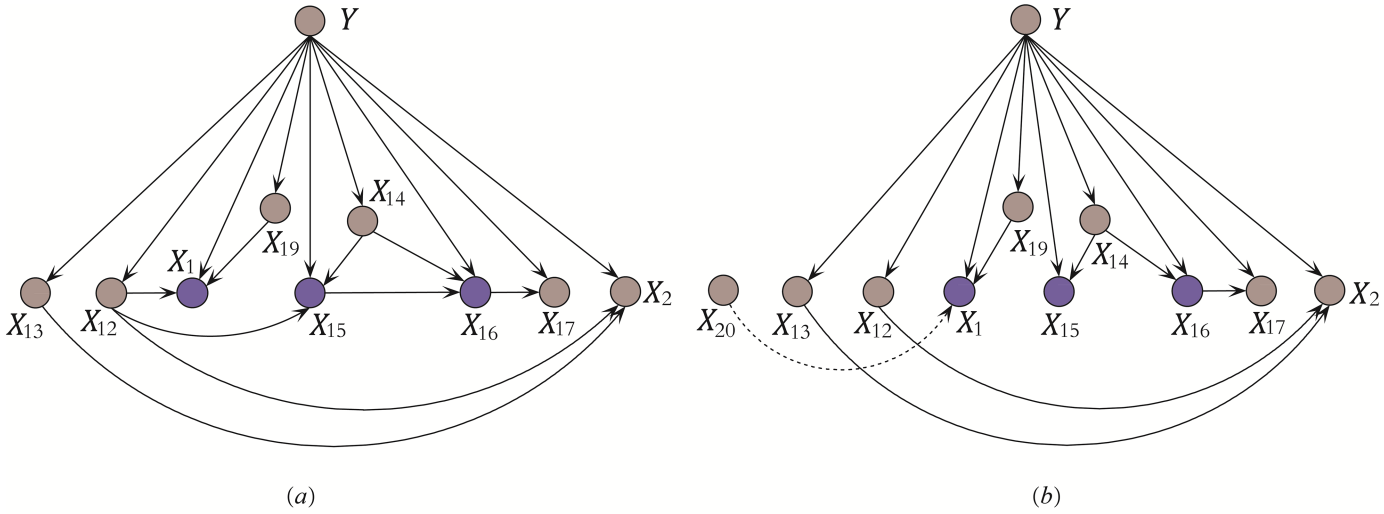

Fig 5. The substructures of KDB_P (a) and KDB_O (b) for $\{X_1, X_{15}, X_{16}\}$ learned from Case2 = ($x_{23} = 51$, $x_{21} = 37$, $x_{20} = 0.5$, $x_{19} = 9.7$, $x_{22} = 0.72$, $x_{13} = y$, $x_{12} = y$, $x_1 = t$, $x_{14} = y$, $x_{15} = y$, $x_{16} = y$, $x_5 = f$, $x_0 = F$, $x_{24} = ?$, $x_{17} = n$, $x_4 = f$, $x_{18} = 46$, $x_6 = f$, $x_8 = f$, $x_2 = f$, $x_{11} = f$, $x_9 = f$, $x_7 = f$, $x_3 = f$, $x_{10} = f$). The arcs X_{12}(TSH measured) → X_{15}(Query hypothyroid) and X_{15}(T4U measured) → X_{16}(FTI measured) in (a) are identified as redundant and removed. No more attributes with higher ranks are considered as possible parents of X_{15} and X_{16}. Arc X_{12}(TSH measured) → X_1(On thyroxine) is substituted with arc X_{20}(T3) → X_1(On thyroxine).

https://doi.org/10.1371/journal.pone.0199822.g005

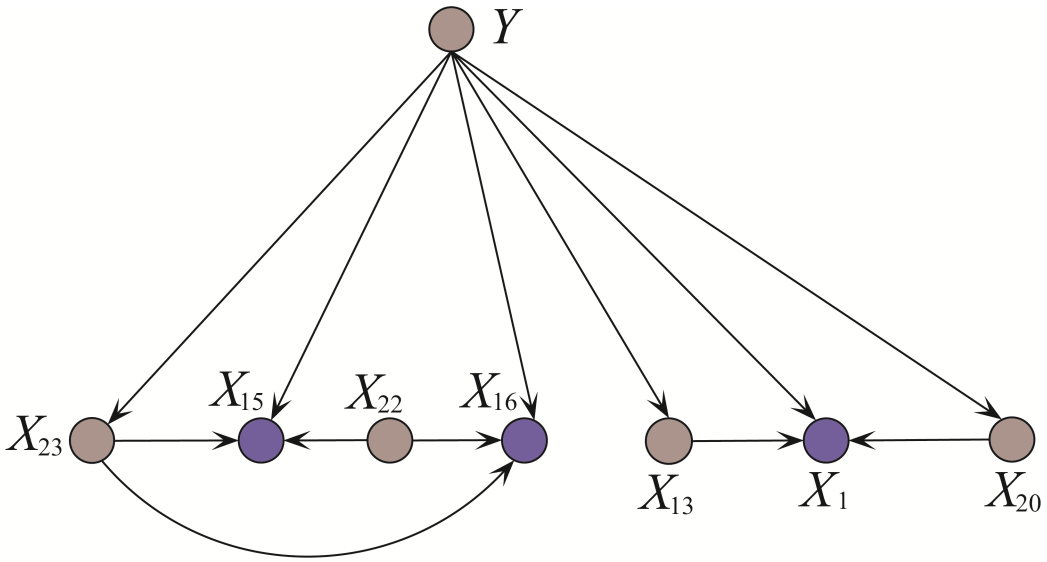

Fig 6. The substructure of KDB for $\{X_1, X_{15}, X_{16}\}$. The arc X_{13} (T3 measured) → X_1 (On thyroxine) is redundant since the information provided by X_{20} (T3) includes the information provided by X_{13} (T3 measured).

https://doi.org/10.1371/journal.pone.0199822.g006

dependencies with other dependencies at classification time. The experimental results show that the classification accuracy (or zero-one loss) and robustness (bias and variance) are significantly enhanced by the addition of RDE. Besides, the dependency relationships that RDE identified in testing instance are irrelevant to class label, thus it is especially applicable to imbalanced data, e.g. `Dis` and `Hypothyroid`. That may be the main reason why RDE obtains the highest AUC values among all the BNCs on the datasets `Dis` and `Hypothyroid`.

RDE searches for the mapping relationships between specific attribute values and then identifies redundant ones. Thus it is suited to probabilistic techniques which deal with discrete attributes, such as KDB. RDE can also be extended to deal with continuous attributes. One possible solution is that, if the conditional probability density function $p(x_j|x_i)$ is relatively high (or greater than a specified value δ) then the mapping relationship $x_i \rightarrow x_j$ is supposed to exist and x_j is redundant. The estimation of $p(x_j|x_i)$ should be learned reliably from training data and the data size should be very large. Although the estimation of $p(x_j|x_i)$ will be time-consuming and more experimental study is needed to determine the value of δ for different attributes, the research work on extending RDE is still very promising.

Author Contributions

Formal analysis: DingBo Duan.

Investigation: Hua Lou.

Methodology: Hua Lou.

Resources: LiMin Wang, Musa Mammadov.

Software: LiMin Wang, DingBo Duan, Cheng Yang.

Validation: Cheng Yang, Musa Mammadov.

Writing – original draft: Hua Lou, LiMin Wang, Musa Mammadov.

References

1. Riccard B, Blaz Z. Predictive data mining in clinical medicine: Current issues and guidelines, International Journal of Medical Informatics, 2008; 77(2): 81–97. https://doi.org/10.1016/j.ijmedinf.2006.11.006
2. Wang LM, Zhao HY, Sun MH, Ning Y. General and Local: Averaged *k*-Dependence Bayesian Classifiers. ENTROPY, 2015; 17(6): 4134–4154. https://doi.org/10.3390/e17064134
3. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Palo Alto, CA. 1988.
4. Wu J, Cai Z. A naive Bayes probability estimation model based on self-adaptive differential evolution. Journal of Intelligent Information Systems, 2014; 42(3): 671–694. https://doi.org/10.1007/s10844-013-0279-y
5. Zheng F, Webb GI. Subsumption resolution: an efficient and effective technique for semi-naive Bayesian learning. Machine Learning, 2011; 87(1): 1947–88.
6. Park SH, Rnkranz J. Efficient implementation of class-based decomposition schemes for Naive Bayes. Machine Learning, 2014; 96(3): 295–309. https://doi.org/10.1007/s10994-013-5430-z
7. Jiang LX, Cai ZH, Wang DH. Improving tree augmented naive bayes for class probability estimation. Knowledge-Based Systems, 2011; 26: 239–45. https://doi.org/10.1016/j.knosys.2011.08.010
8. Zheng F, Geoffrey W. Efficient lazy elimination for averaged one-dependence estimators. in Proceedings of the Twenty-third International Conference on Machine Learning, 2006; 1113–1120.
9. Francisco L, Anderson A. Bagging *k*-dependence probabilistic networksAn alternative powerful fraud detection tool. Expert Systems with Applications, 2012; 39(14): 11583–92. https://doi.org/10.1016/j.eswa.2012.04.024
10. Taheri S, Mammadov M. Structure learning of Bayesian Networks using global optimization with applications in data classification. Optimization Letters, 2015; 9(5): 931–948. https://doi.org/10.1007/s11590-014-0803-1
11. Sahami M. Learning limited dependence Bayesian classifiers. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996: 335–338.
12. https://www.healthline.com/health/hiv-aids/early-signs-hiv-infection#stages-of-hiv
13. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995; 20(3): 273–297. https://doi.org/10.1007/BF00994018

14. Breiman L. Random Forests. Machine Learning, 2001; 45(1): 5–32. In Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1993; 1022–1029.
15. Murphy PM, Aha DW. UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/datasets.html, 1995.
16. Fayyad UM, Irani KB. Multi-interval discretization of continuousvalued attributes for classification learning.
17. Kohavi R, Wolpert D. Bias plus variance decomposition for zero-one loss functions. In Proceedings of the 13th International Conference on Machine Learning, 1996; 275–283.
18. Friedman M. A comparison of alternative tests of significance for the problem of *m* rankings. Journal of the American Statistical Association, 1940; 11(1): 86–92.
19. Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. 1997; 30(7): 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2
20. Chow C. Liu C. Approximating discrete probability distributions with dependency trees. IEEE Transactions on Information Theory. 1968; 14: 462–467. https://doi.org/10.1109/TIT.1968.1054142
21. Shannon CE. The Mathematical Theory of Communication. University of Illinois Press, 1949.
22. Kenneth WC, Patrick H. Word association norms, mutual information, and lexicography. Meeting on Association for Computational Linguist, 1989; 16(1): 22–29.
23. Kittler J. Feature selection and extraction. Handbook of pattern recognition and image processing. New York: Academic Press, 1986.
24. Blanco R, Inza I, Merino M and Quiroga J. Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS. Journal of Biomedical Informatics, 2005; 8: 376–388. https://doi.org/10.1016/j.jbi.2005.05.004
25. Bouckaert RR. Voting massive collections of Bayesian network classifiers for data streams. In Proceedings of the 19th Australian joint conference on Artificial Intelligence: advances in Artificial Intelligence, Berlin, Heidelberg, Springer-Verlag, 2006; 243–252.
26. Chen SL, Nayyar AZ. Scalable learning of Bayesian network classifiers. Journal of Machine Learning Research, 2013; 1–30.
27. Chen SL, Martinez AM, Webb GI, Wang LM. Sample Based Attribute Selective AnDE for Large Data. IEEE Transactions on Knowledge and Data Engineering, 2017; 29(1): 172–185. https://doi.org/10.1109/TKDE.2016.2608881
28. http://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/HEL-LO