Integrated Computer-Aided Engineering 24 (2017) 17–26
DOI 10.3233/ICA-160514
IOS Press

Learning cost-sensitive Bayesian networks via direct and indirect methods

Eman Nashnush and Sunil Vadera*
The School of Computing, Science and Engineering, University of Salford, Manchester, UK

Abstract. Cost-sensitive learning has become an increasingly important area that recognizes that real world classification problems need to take the costs of misclassification and accuracy into account. Much work has been done on cost-sensitive decision tree learning, but very little has been done on cost-sensitive Bayesian networks. Although there has been significant research on Bayesian networks there has been relatively little research on learning cost-sensitive Bayesian networks. Hence, this paper explores whether it is possible to develop algorithms that learn cost-sensitive Bayesian networks by taking (i) an indirect approach that changes the data distribution to reflect the costs of misclassification; and (ii) a direct approach that amends an existing accuracy based algorithm for learning Bayesian networks.An empirical comparison of the new approaches is carried out with cost-sensitive decision tree learning algorithms on 33 data sets, and the results show that the new algorithms perform better in terms of misclassification cost and maintaining accuracy.

Keywords: Cost-sensitive classification, Bayesian learning, decision trees

1. Introduction

The ability to learn classifiers has been one of the major success stories of AI [13,36], with a wide range of methods such as neural networks, decision tree induction, support vector machines, and Bayesian networks utilised in many real world applications such as fraud detection, assessing credit, cyber security and medical diagnosis [1,24,37].

Early machine learning algorithms, now termed cost-insensitive learning algorithms, focused on maximizing accuracy but did not take any type of costs into account [20]. Several authors have noted that this is not adequate for practical applications [7,18]. For example, in medical diagnosis applications, the cost of a false positive error includes unnecessary treatment and unnecessary worry while the cost of a false negative error includes postponed treatment or failure to treat and death or injury [28]. In fraud detection applications, a false positive error can lead to resources being wasted investigating non-frauds and reducing the benefits; while a false negative error such as a failure to detect fraud could be very expensive [25].

Table 1
A cost matrix for two-class problems

Predicting class	Actual class	
	Actual positive	Actual negative
Predicting positive	TP = 0	FP = £1
Predicting negative	FN = £50	TN = 0

Hence, in recent years, a significant level of attention has been paid to cost-sensitive learning, including making accuracy-based learners cost-sensitive [18]. Cost-sensitive learning algorithms take costs into consideration and aim to minimize expected cost [16]. The following example illustrates the use of a cost matrix together with some of the key ideas. Table 1 presents an example cost matrix, where $C_{i,j}$ is the cost of predicting an example to be in class i when it is actually in class j [9,34].

A classification scheme, when applied to some data, will lead to outcomes that are correct or incorrect, resulting in what is known as a confusion matrix. For example, suppose we have two different classifiers, C1 and C2, induced by two different learning algo-

*Corresponding author: Sunil Vadera, The School of Computing, Science and Engineering, Salford University, Manchester, UK. E-mail: S.Vadera@salford.ac.uk.

ISSN 1069-2509/17/$35.00 © 2017 – IOS Press and the author(s). All rights reserved
This article is published online with Open Access and distributed under the terms of the Creative Commons Attribution Non-Commercial License (CC BY-NC 4.0).

Table 2a
Outcomes of decision tree classifier (J48) on Hepatitis test set

Predicting class	Actual class	
	Actual die	Actual live
Predicting die	4	7
Predicting live	5	25

Table 2b
Outcomes of Bayesian network classifier on Hepatitis test set

Predicting class	Actual class	
	Actual die	Actual live
Predicting die	8	2
Predicting live	1	30

rithms; say a Decision tree classifier and a Bayesian network classifier. Applying these classifiers to a Hepatitis dataset and evaluating the supplied test set in the model may give the results in Tables 2(a) and 2(b) respectively.

Given the outcomes in Tables 2(a) and 2(b), we can compute the accuracy and misclassification costs of the two classifiers by using the following measures:

$$\text{Accuracy} = \frac{\text{No. of correct examples}}{\text{Total number of examples}} \tag{1}$$

$$\text{Cost} = \sum_{j=0}^{N} \text{No Missclassified}_j * \text{Cost}_j \tag{2}$$

Where N is the number of classes, No Misclassified$_j$ is the number of class j examples that are misclassified, and Cost$_j$ is the cost of misclassifying examples of class j. Using these equations, we obtain the following accuracies and costs for the decision tree (DT) and the Bayesian network (BN):

DT Accuracy $= 70.73\%$
DT Misclassification cost $= 257$
BN Accuracy $= 92.68\%$
BN Misclassification cost $= 52$

Thus, in this example, applying the Bayesian network classifier will entail less cost than applying the Decision tree classifier on the Hepatitis dataset. The task of cost-sensitive learning is to induce classifiers that minimize cost. Most of the work on inducing classifiers has focused on decision tree learning and, to the best knowledge of the authors, there has been no attempt to assess whether the use of Bayesian networks can produce better cost-sensitive classifiers than decision tree induction. Hence, this study aims to explore the use of Bayesian networks (BNs) for cost-sensitive classification.

The study builds upon a paper presented at the First International Conference on Soft Computing and Data Mining (SCDM-2014) in which the authors presented the initial results from using a sampling method [21]. In this paper, the authors present a new direct approach to learn cost-sensitive Bayesian networks, and present new results in comparison to the use of sampling.

This paper is organized as follows. Section 2 presents some background on Bayesian networks. Section 3 provides a number of definitions and a summary of related work and background information on cost-sensitive learning algorithms. Section 4 presents two alternative approaches for learning cost-sensitive Bayesian networks: one by using a direct approach that amends an algorithm and a second that uses an indirect approach that uses sampling to amend the distribution of the training data to reflect costs of misclassification as previously presented in [21]. Section 5 shows the results obtained by carrying out an empirical evaluation on data from the UCI repository [3]. Finally, Section 6 provides a conclusion and summary of the main contribution of this paper.

2. Background on Bayesian networks

A Bayesian network (BN) can be used as a classifier by computing the posteriori probability of a set of labels given the observable features [24]. According to Neapolitan [22] there are two aspects to constructing a BN: *(i) learn the graphical structure* (topology), that is the relationships between the variables; and *(ii) learn the parameters* (conditional probability estimation) which quantify the extent of the relationships.

Learning Bayesian networks can be super exponential in the number of nodes and is known to be an NP-hard problem [6,19], and hence several algorithms have been developed that reduce the size of the search space by limiting the type of topology that is learned. One of the first was due to Chow and Liu [5], who in 1968, proposed a method for learning a *Bayesian tree* (also called a Chow-Liu Tree) based on approximating the joint distribution of a set of discrete variables using the products of distributions involving no more than pairs of variables as shown in Fig. 1(a). This was extended by Pearl [24], in 1988, to learn singly-connected graphs; which are Directed Acyclic Graphs (DAGs) where any two nodes only have at most one unique path as shown in Fig. 1(c). In contrast, in 1992, Langley et al. [15] developed an algorithm for learning a simpler structure known as a *Naive*

Bayes structure, where all attributes are represented as independent nodes that have one parent (class node). A Naive Bayes classifier, as shown in Fig. 1(b), assumes conditional independence of the features given the class. Naive Bayes is easy to construct and it has been used as a classifier for many years, especially where the features are not strongly correlated. More recently, in 1997, Friedman et al. [11] developed a natural extension to the Naive Bayes classifier and the Chow-Liu algorithm, where they introduce the *Tree Augmented Naive Bayes* (TAN) structure. In contrast to Naive Bayes, where the assumption is that all attributes are independent, in a TAN all attributes are conditionally independent given the value of the class. Thus in a TAN, the correlations between attributes can be captured by adding additional edges between attributes, as shown in Fig. 1(d).

Learning network structure requires searching for the best network according to a score. Many scoring criteria have been proposed such as: the *Bayesian Dirichlet scoring function (BD)* [14], the *Bayesian Information Criterion (BIC)* [29], the *Minimum Description Length (MDL)* [27], and the *Akaike's Information Criterion (AIC*) [2]. All these measures have different characteristics and the reader can refer to [12] for details. The MDL measure is used in the algorithm we adapt and is described below.

Let us assume that $B = <$G, O$>$ is a Bayesian network, where G is an acyclic graph and O denotes the parameters consisting of the conditional probabilities.

Let D be a training set, then the MDL score can be defined by [11,22]:

$$\text{MDL(B|D)} = 1/2\text{logN} * |\text{B}| - \text{LL(B|D)} \qquad (3)$$

There are two parts to this definition. The first part, $^1/_2$ log N * |B| denotes the number of bits required to represent the network, where N is the number of instances; |B| is the number of parameters in the network; and $^1/_2$ log N represents the number of bits that are used for each parameter. The second part, LL(B|D), represents the log likelihood of B given D, and denotes how many bits are needed to describe the data D based on the probability distribution P_B and is given by [11,19,22]:

$$\text{LL(B|D)} = \sum_{i=1}^{N} P_B(u_i) * \log(P_B(u_i)) \qquad (4)$$

In particular, the highest log likelihood refers to the closest model B with the probability distribution of the data D. The MDL score focuses on combining the length of the network description and encoding the data to be minimized.

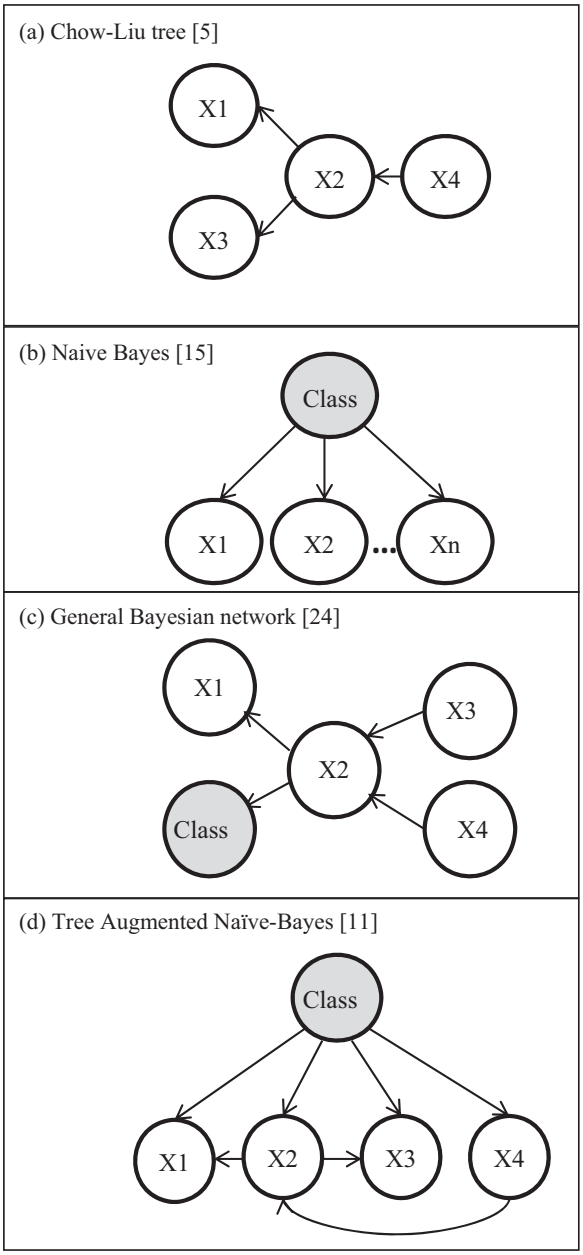

Fig. 1. Bayesian structures.

3. An overview of approaches to cost-sensitive learning

As illustrated by the example in Section 1, the aim of a cost-sensitive classifier is to minimize the expected cost of classification [9].

Several authors have categorized cost-sensitive induction algorithms. According to Zadrozny et al. [34], cost-sensitive classifiers can be divided into two cat-

egories: *Black Box* and *Transparent Box.* Black box methods use a closed box without changing the classifier behavior and can work for any classifier. On the other hand, transparent box methods require knowledge of the particular learning algorithm and are based on changing the algorithm to include costs. Ling and Sheng [16] use different terms such as *direct method*, and *indirect methods*.

A *direct method,* introduces misclassification costs into algorithms such as cost sensitive decision trees [4, 8,17,23]. On the other hand, *indirect methods* use techniques such as Sampling [31,34], Relabelling [7,33], Weighting [32], and Thresholding [9,30]. These methods can be applied before or after applying an existing accuracy based classifier. The following sections describe the application of direct and indirect methods to produce cost-sensitive decision tree learning algorithms. There are numerous methods that could be described and we focus on just a couple to illustrate the main ideas that we use later in Sections 4 and 5. Readers interested in other methods are referred to a comprehensive survey carried out by Lomax and Vadera [18].

3.1. Cost-sensitive direct learning methods

A key step in decision tree learning is to select the criteria used for the next node of the decision tree and to split the data. Early decision tree induction algorithms that focused on accuracy used a measure based on information theory to select the splitting criteria. For example, ID3, and C4.5 [26] are based on calculating the gain in information achieved by each of the attributes if these were chosen for the split and choosing the attribute which maximizes this gain:

$$\text{Info}_A = E(D) - E(A) \tag{5}$$

Where:

$$E(D) = \sum_{c \in C} -\frac{N_c}{N} * \log_2 \frac{N_c}{N}$$

$$E(A) = \sum_{a \in A} P(a) * \sum_{c \in C} -P(a|c) * \log_2 P(a|c)$$

Where, $a \in A$ are the values of attribute A, and $c \in C$ are the class values.

Thus, an obvious way of adapting these algorithms is to amend this measure to take account of costs. For example, Breiman et al. [4] modify the class probabilities that are used in the information gain measure, and exchange that probability with the altered probability as shown in Eq. (6), where the probability for a class i is weighted by the relative cost of misclassifying an example of class i (Cost ratio$_i$).

$$\text{Altered Probability } i = \text{Cost ratio}_i * \left(\frac{Ni}{N}\right) \tag{6}$$

Where, for k classes:

$$\text{Cost ratio}_i = \frac{\text{cost } i}{\Sigma_j^k \text{cost } j}$$

For example, for the cost matrix in Table 1, the cost ratio for the positive class is 50/51, while, the cost ratio for the negative class is 1/51. Also, Ni is the number of examples in class i, while N is the total number of examples.

3.2. Cost-sensitive indirect learning method

Indirect methods do not change the learning process of a classifier and instead use it as a black box. As an example, consider one the earliest indirect methods, called MetaCost [7]. In this method, an accuracy based learner is used on several samples of the data, each resulting in a decision tree. The resulting trees are used to predict the class of each example, and then used to predict the class that minimises the cost, which in turn is used to relabel the examples. The accuracy based learner is then applied on the relabelled data to produce a cost-sensitive decision tree. Another interesting indirect method is Costing [34] which makes use of a result due to Elkan [9] that states a Folk Theorem that the data distribution can be changed to reflect the costs. As Zadrozny et al. [35] state:

> "*If the new examples are drawn from the old distribution, then optimal error rate classifiers for the new distributions are optimal cost minimizers for data drawn from the original distribution.*"

This is presented as the following equation [35]:

$$D'(x, y, c) = \frac{C}{E_{x,y,c \sim D[c]}} D(x, y, c) \tag{7}$$

Where, the new distribution D' = factor * Old distribution D; x is instance; y is the class label; and C is the cost according to misclassified instance x. This theorem can be used to create a new distribution from the old distribution by multiplying the old distribution with a factor proportional to the relative cost of each example.

For example, consider the hepatitis dataset,which has 32 instances in the class "die" (class distribution

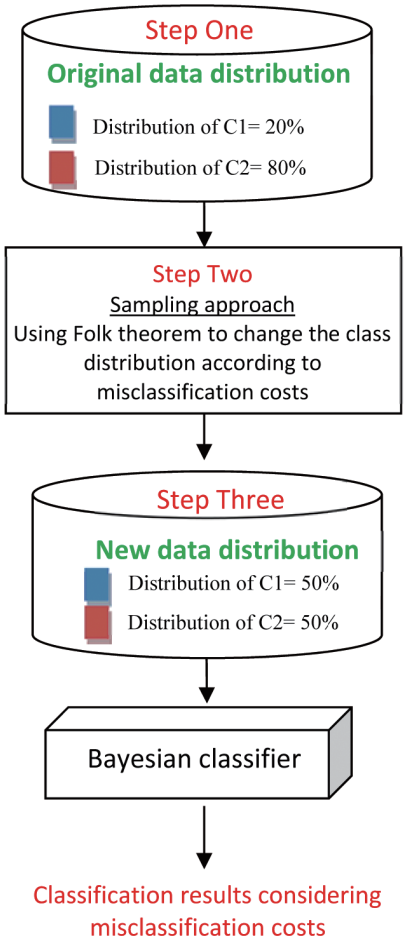

Fig. 2. Changing the data distribution of hepatitis dataset.

0%), and 123 instances in the class "live" (class distribution 80%). Given the imbalance in examples for the two classes, an accuracy based classifier will always be biased to the most common class, that is live, though misclassifying examples of class "die" is more serious. The folk theorem can be used to address this kind of situation. Suppose the misclassification costs are 4:1 for Live: Die respectively. Then the new distribution of class die = 4*32 = 128 die (class distribution 50%); and the new distribution of class live = 1*123 = 123 live (class distribution 50%), as summarised by the steps given in Fig. 2.

4. Development of cost-sensitive Bayesian networks

The direct and indirect methods described above, have been used mainly when developing cost-sensitive decision tree learners. Section 4.1 describes the use of a direct approach for learning cost-sensitive Bayesian networks and Section 4.2 describes an indirect approach for learning cost-sensitive Bayesian networks.

4.1. Cost-sensitive Bayesian network induction via a direct approach

As described in Section 2, a key step of existing algorithms for learning the structure of a Bayesian network is to compute the Minimum Description Length (MDL). Hence, by analogy with the approach taken for decision trees, where the information theoretic measure was modified, the modification made to develop our new algorithm is to change the original MDL measure [27], which was described in Section 2 in Eqs (3) and (4).

As with the modification made for decision trees, we make two amendments when learning the structure of a Bayesian network.

First, the Log-likelihood factor that is used in the MDL measure Eq. (4) is amended to take account of costs. The modification made is to multiply each part of the information measurement with the cost proportion of a class, resulting in the new LL(B|D) given in Eq. (8).

$$\text{LL(B|D)} = \sum_{j=1}^{k}\sum_{i=1}^{N} p(x_i, \pi_{xi}) \log_2 \left(\frac{p(x_i, \pi_{xi})}{p(\pi_{xi})}\right) * \text{Cost ratio}_j \tag{8}$$

Where, M is the number of class labels, N is the number of parent attributes to node x_i, π_{xi} represents the parents of the attribute x_i and Cost ratio$_j$ is the ratio of the cost of misclassifying class j over the total costs, as described in Section 3.1. While $p(x_i, \pi_{xi})$ represents the probabilities of events in D.

Secondly, the parameters are modified to reflect costs by modifying the conditional probability of each node given its parent. That is, instead of using the Laplace estimator of $P(i)$, we weight it by the cost ratio:

$$P_{\text{class}_j}(x_i|\pi_{xi}) = \text{Cost ratio}_j * \frac{p(x_i, \pi_{xi}) + 1}{p(\pi_{xi}) + n_{xi}} \tag{9}$$

Where, x_i is the node that is connected with π_{xi} (class label, and another parent); n_{xi} is the number of possible values of node x_i.

These amendments lead to the algorithm presented in Fig. 3, which is an amended version of the algorithm by Friedman et al. [11]. This algorithm was implemented in Java in the WEKA system [33] and an empirical comparison with existing algorithms is presented in Section 5.

CS-BN via Direct approach:

1. Compute new conditional mutual information between each pair of attributes (nodes) based on class label, and include cost proportions for each class in the calculation, based on MDL score:

$$\sum_{k}^{N_{class}} \sum_{y}^{y_{value}} \sum_{x}^{x_{value}} p(x, y, Class_k) \log_2 \frac{p(x, y, Class_k)}{p(y, Class_k)} * \text{Cost ratio}_k$$

2. Build a complete undirected graph between each pair of attributes (nodes) without class node.
3. Use the Maximum Weight Spanning Tree algorithm, to maximize the information gained about the classification weighted by the cost of misclassification to obtain a tree.
4. Convert the tree to a directed tree.
5. Add the class label as root for all attributes (nodes).
6. Learn the parameters for each node with its parent by using the new probability estimation that includes misclassification costs.

$$p_{Class\ K}(X|\ Y, Class_K) = \text{Cost ratio}_k * \frac{P(x, y, Class_k)}{P(y, Class_k)}$$

Fig. 3. Cost-Sensitive Bayesian Network Algorithm by direct amendment.

4.2. *Cost-sensitive Bayesian networks induction via an indirect approach*

This section presents an indirect approach to develop cost-sensitive Bayesian networks (CS-BNs) that uses sampling to take account of misclassification costs. The approach used is based on that introduced by Zadrozny et al. [35] and Elkan's Folk Theorem [9] that was described in Section 3.2. This theorem draws a new distribution from the old distribution, according to cost proportions to change the data distribution and obtain optimal cost-minimization from the original distribution. Figure 4 gives the algorithm we adopt using sampling.

The main steps of this algorithm are:

Step 1: The data are split into a training set and testing set. The training set uses 75% of the original data, while the testing set uses 25% of the original data.[1]

Step 2: The distribution of the data is altered to take account of costs. For instance, if the cost of wrongly classifying a sick patient as healthy is £20 and the cost of misclassifying a healthy patient as sick is £2, then the cost ratio of the sick class will be 20/22 = 0.90.

[1] Other ways of splitting the data could, of course be adopted without affecting the principles of the approach.

CS-BN via Indirect approach (Sampling)

1. Divide dataset into 75% of instances for training, and 25% for testing. With the same class distributions.
2. Change the data distribution according to the cost ratio of each class:

$$\text{Cost ratio}_i = \frac{\text{costi}}{\sum_j \text{cost j}}$$

3. Learn the TAN structure and its parameters
4. Evaluate the TAN on the original test set distribution.

Fig. 4. Cost-sensitive Bayesian network algorithm by indirect approach using sampling.

The cost ratios are then used to change the data distributions. For example, if a dataset has a class distributions of 50% for each class, when the costs are 1:4, the new proportions for each class will be 20% and 80% respectively. There are different methods that can be used to sample the data to redistribute the data. During our research, we used two methods, under-sampling and over-sampling. Where the new proportion was less than the original proportion, we used under-sampling (without replacement) to delete some of the examples in the frequent class. On the other hand, if the new proportion was greater than the original proportion, we used over-sampling (with replacement) to randomly select new instances which belonged to the rare class, and hence increase the number of examples.

Step 3: Uses Friedman et al.'s [11] cost-insensitive algorithm on the new distribution from step 2.

Step 4: Evaluates the model on the original distribution.

5. Empirical evaluation

This section presents an empirical evaluation of the amended cost-sensitive BN algorithm, and CS-BN using a sampling approach. The evaluation is carried out using 33 data sets from the UCI repository [3] and adopting the 75% training and 25% testing methodology. The cost matrix adopts 16 cost ratios [4:1,4:2,4:3,4:4, 3:1,.., 1:4]. The evaluation is carried out with respect to the two methods presented in this paper as well as the following algorithms:

(i) The original TAN learning algorithm [11], to assess the extent to which the amendments make a difference.

(ii) The MetaCost [7] algorithm with J48 as the base classifier to compare against a cost-sensitive decision tree learner that is known to perform well.

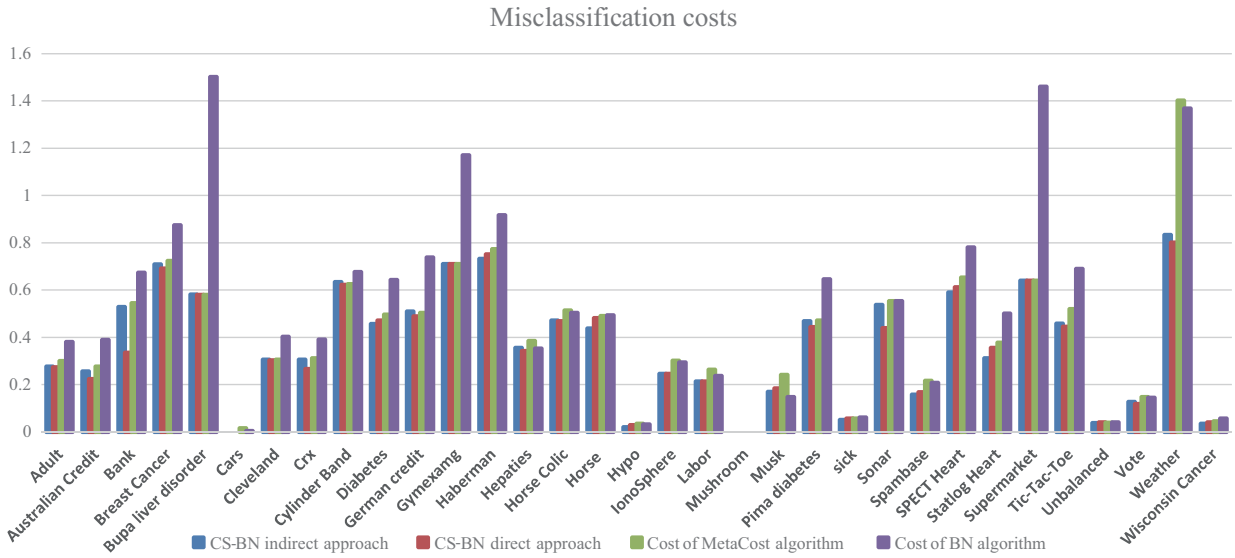

Fig. 5. Expected cost per instance of CS-BN via direct, indirect methods and existing algorithms.

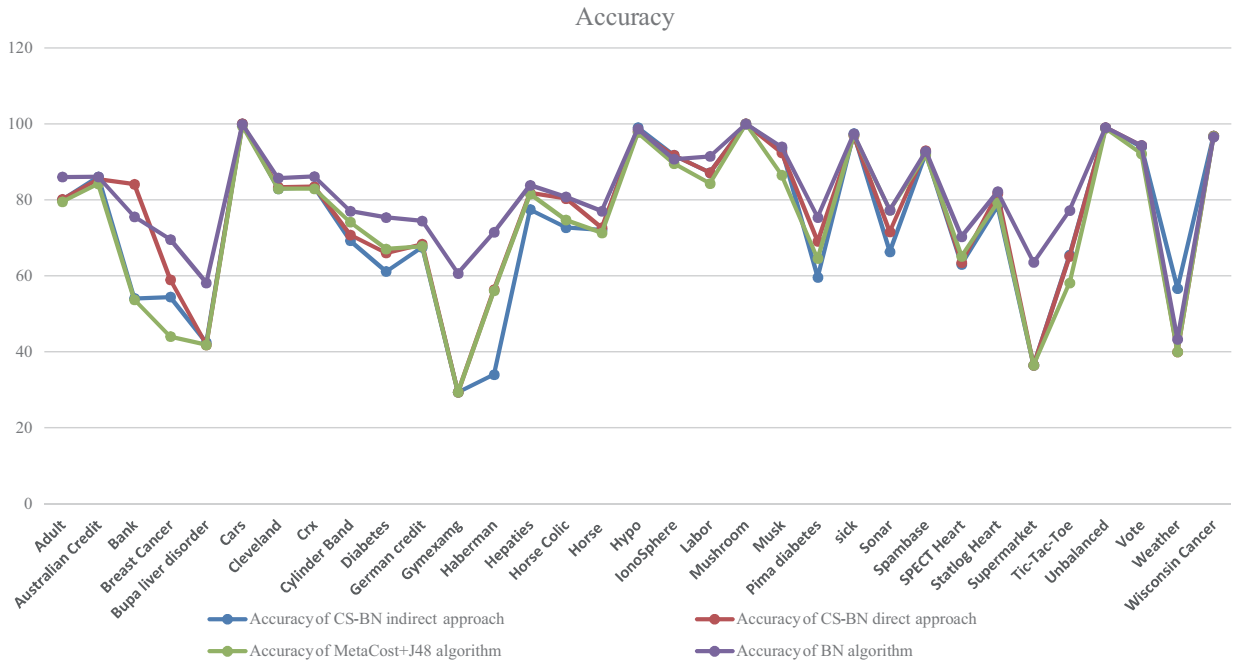

Fig. 6. Accuracy of CS-BN via direct, indirect methods and an existing algorithm.

Table 3 presents the results for each of the 33 data sets and highlights the result with the lowest cost for each data set. Figure 5 presents the results of expected costs for each data set in the form of bar charts, and Fig. 6 presents the accuracy across different data sets.

These experiments show that:

(i) The numbers of misclassifications of the rare class (i.e., more expensive) are always less than the number of misclassifications of the frequent class in all datasets.

(ii) The use of the direct approach gives good results on most of types of data, whether numeric, nominal, balanced, or unbalanced data. The direct approach performed better in terms of min-

Table 3
Comparison between CS-BN via direct and indirect methods

Dataset	CS-BN direct approach		CS-BN Indirect approach		MetaCost +J48		Original BN	
	Cost	Accuracy	Cost	Accuracy	Cost	Accuracy	Cost	Accuracy
Adult	**3344.2 ± 22.81**	80.13 ± 0.09	3375.0 ± 39.26	80.08 ± 0.17	3651.5 ± 39.48	79.51 ± 0.15	4640.1 ± 34.25	86.02 ± 0.14
Australian Credit	**37.8 ± 1.31**	85.44 ± 0.57	43.4 ± 3.8	86.04 ± 0.98	46.5 ± 1.57	84.2 ± 0.65	65.9 ± 3.45	86.04 ± 0.68
Bank	**49.6 ± 2.92**	84.12 ± 0.96	78.2 ± 2.98	54.05 ± 1.61	80.8 ± 2.13	53.72 ± 1.45	99.8 ± 5.05	75.54 ± 1.34
Breast Cancer	**48.5 ± 3.0**	59.0 ± 0.93	49.6 ± 2.93	54.43 ± 1.01	50.6 ± 2.5	44.0 ± 1.41	61.2 ± 3.37	69.57 ± 1.68
Bupa liver disorder	**50.0 ± 0.0**	41.86 ± 0.0	50.1 ± 0.1	42.44 ± 0.4	50.0 ± 0.0	41.86 ± 0.0	138.0 ± 6.0	58.14 ± 0.0
Cars	0.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	100.0 ± 0.0	1.4 ± 0.65	99.43 ± 0.26	0.4 ± 0.4	99.89 ± 0.11
Cleveland disease	**22.7 ± 2.01**	83.33 ± 1.15	23.0 ± 2.06	82.93 ± 1.25	23.0 ± 2.06	82.93 ± 1.25	30.2 ± 3.16	85.73 ± 1.23
Crx	45.0 ± 4.67	83.49 ± 1.19	51.9 ± 2.59	83.31 ± 0.63	52.6 ± 2.07	82.9 ± 0.5	66.0 ± 3.07	86.15 ± 0.55
Cylinder Band	83.6 ± 1.85	70.75 ± 0.79	85.0 ± 3.81	69.25 ± 0.87	83.9 ± 3.14	74.1 ± 0.82	90.8 ± 5.65	77.01 ± 0.89
Diabetes	**89.8 ± 4.26**	66.02 ± 0.8	86.8 ± 2.65	61.15 ± 1.14	95.0 ± 3.52	67.07 ± 1.06	122.7 ± 3.68	75.34 ± 0.59
German credit	**122.7 ± 6.32**	68.32 ± 1.08	127.1 ± 6.37	67.52 ± 0.97	125.9 ± 4.9	67.76 ± 1.25	184.2 ± 5.44	74.44 ± 0.65
Gymexamg	**438.0 ± 0.0**	29.35 ± 0.0	438.0 ± 0.0	29.35 ± 0.0	438.0 ± 0.0	29.35 ± 0.0	728.0 ± 0.0	60.65 ± 0.0
Haberman	56 ± 3.2	56.4 ± 1.77	**54.9 ± 1.72**	34.0 ± 2.12	58.1 ± 2.21	56.13 ± 1.63	68.8 ± 1.08	71.47 ± 0.8
Hepaties	**13.4 ± 2.06**	81.79 ± 2.04	13.9 ± 1.39	77.44 ± 2.25	15.0 ± 1.91	81.54 ± 2.22	13.8 ± 2.12	83.85 ± 1.62
Horse Colic	**43.0 ± 2.86**	80.33 ± 1.23	43.4 ± 2.56	72.72 ± 1.13	47.3 ± 2.68	74.67 ± 1.34	46.2 ± 2.61	80.76 ± 1.32
Horse	43.8 ± 2.67	72.64 ± 1.59	**43.4 ± 4.51**	72.09 ± 1.56	44.4 ± 3.82	71.32 ± 1.16	44.9 ± 3.44	77.03 ± 1.45
Hypo	22.5 ± 2.4	98.08 ± 0.23	**16.7 ± 2.7**	98.98 ± 0.18	26.7 ± 3.51	97.7 ± 0.22	25.2 ± 2.99	98.66 ± 0.13
IonoSphere	**21.3 ± 3.13**	91.72 ± 0.89	**21.3 ± 3.13**	91.72 ± 0.89	26.2 ± 2.35	89.54 ± 0.53	28.0 ± 2.76	87.47 ± 0.95
Labor	**3.0 ± 1.06**	87.14 ± 2.33	**3.0 ± 1.06**	87.14 ± 2.33	3.7 ± 0.96	84.29 ± 2.56	3.3 ± 1.19	91.43 ± 2.33
Mushroom	**0.0 ± 0.0**	100.0 ± 0.0	0.0 ± 0.0	100.0 ± 0.0	2.4 ± 0.88	99.97 ± 0.01	2.0 ± 0.67	99.98 ± 0.01
Musk	21.5 ± 2.7	92.39 ± 0.77	**19.7 ± 8.31**	93.42 ± 2.35	28.1 ± 2.94	86.5 ± 1.09	17.3 ± 1.94	93.93 ± 0.63
Pima diabetes	**84.9 ± 3.24**	69.06 ± 1.2	89.4 ± 2.2	59.63 ± 1.16	89.9 ± 3.55	64.55 ± 1.03	123.6 ± 4.09	75.34 ± 1.1
Sick	40.2 ± 1.7	96.86 ± 0.14	**34.3 ± 2.08**	97.4 ± 0.16	40.7 ± 2.79	97.04 ± 0.14	43.5 ± 2.66	97.33 ± 0.12
Sonar	**22.9 ± 2.28**	71.54 ± 1.51	28.0 ± 2.13	66.35 ± 1.29	28.9 ± 3.03	77.31 ± 2.25	28.9 ± 3.03	77.31 ± 2.25
Spambase	190.0 ± 4.91	92.96 ± 0.18	**177.7 ± 8.14**	92.35 ± 0.26	244.3 ± 6.53	91.87 ± 0.17	234.5 ± 9.05	92.77 ± 0.18
SPECT Heart	40.3 ± 2.26	63.48 ± 1.47	**38.8 ± 2.46**	63.03 ± 1.58	43.1 ± 1.63	65.15 ± 1.69	51.4 ± 1.19	70.3 ± 0.79
Statlog Heart	23.5 ± 2.25	81.67 ± 1.38	**20.6 ± 1.42**	78.33 ± 0.72	25.1 ± 2.25	79.24 ± 1.11	33.1 ± 3.12	82.12 ± 1.29
Supermarket	727.0 ± 0.0	36.45 ± 0.0	727.0 ± 0.0	36.45 ± 0.0	727.0 ± 0.0	36.45 ± 0.0	1668.0 ± 0.0	63.55 ± 0.0
Tic-Tac-Toe	**105.1 ± 3.26**	65.13 ± 1.0	107.9 ± 4.52	65.47 ± 1.1	122.5 ± 3.54	58.14 ± 0.96	162.7 ± 5.04	77.2 ± 0.52
Unbalanced	8.0 ± 0.0	99.05 ± 0.0	**7.9 ± 0.35**	98.95 ± 0.06	8.2 ± 0.47	98.81 ± 0.15	8.0 ± 0.0	99.05 ± 0.0
Vote	**12.7 ± 1.38**	94.3 ± 0.51	13.7 ± 1.27	93.93 ± 0.56	15.6 ± 1.67	92.15 ± 0.59	15.4 ± 1.67	94.3 ± 0.45
Weather	**2.4 ± 0.64**	40.0 ± 9.69	2.5 ± 0.64	56.67 ± 7.11	4.2 ± 0.57	40.0 ± 8.31	4.1 ± 0.57	43.33 ± 8.68
Wisconsin Cancer	7.0 ± 0.8	96.8 ± 0.35	**5.8 ± 0.88**	96.8 ± 0.4-7	7.8 ± 0.89	96.69 ± 0.25	9.9 ± 1.08	96.51 ± 0.31

imizing costs than the indirect approach in 19 out of the 33 data sets evaluated

(iii) The use of the indirect approach, involving changing the data distributions yields good results on most data; especially if the data are not very highly skewed towards one class.

(iv) The indirect approach performed better in terms of minimizing costs than the direct approach in 14 out of the 33 data sets evaluated.

(v) Overall, both the direct and indirect versions outperform MetaCost+J48, and the original accuracy only version in terms of minimizing cost. Both approaches performed better in terms of minimizing costs than MetaCost+J48, and original BN algorithm in 27 out of the 33 data sets evaluated.

(vi) The accuracy of the cost-sensitive version is similar but slightly less than the original accuracy based version of TAN, though the level of sacrifice is not as significant as reported in studies that use similar approaches for learning cost-sensitive decision trees [18].

6. Conclusion

Cost-sensitive learning algorithms have received increasing attention in most real world applications, though most of the existing studies are devoted to making decision trees cost-sensitive. Existing Bayesian network algorithms that are designed to minimize misclassification errors do not take misclassification costs into consideration. Hence, this study has explored whether it is possible to develop cost-sensitive Bayesian networks. Two algorithms were developed by analogy with the strategies used for producing cost-sensitive decision trees: (i) a direct approach that involved amending the MDL measure used in constructing a network and (ii) an indirect approach, based on sampling to change the distribution of examples to reflect the costs of misclassification.

The main findings from the empirical evaluation relative to other algorithms are:

- As one would expect, the new algorithms outperformed the cost-insensitive version of the algorithm that learns Bayesian networks.
- The direct approach gives good results on most of the data sets and is also better than the indirect approach on some data sets, while, the indirect approach works very well when the data are not very highly skewed towards one class.
- In our evaluations, both the direct and indirect approaches performed better than MetaCost+J48 in terms of minimizing costs.

In conclusion, application of strategies to induce cost-sensitive decision trees to learn cost-sensitive Bayesian networks have proved to be effective and, in general, lead to more cost-effective classification than the use of decision trees.

Acknowledgements

This paper is an extended and refined version of the paper presented at the First International Conference on Soft Computing and Data Mining (SCDM-2014) reference [21]. The authors are grateful to the reviewers and editor for their constructive comments that have helped improve the presentation of this paper.

References

[1] M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integrated Computer-Aided Engineering* **17**(3) (2010), 197–210.

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

[3] A. Asuncion and D.H. Newman, UCI machine learning repository, https://archive.ics.uci.edu/ml/datasets.html, (2007).

[4] L. Breiman, J. Friedman, C.J. Stone and R.A. Olshen, Classification and regression trees. CRC press, 1984.

[5] C.K. Chow and C.N. Liu, Approximating discrete probability distributions with dependence trees, *IEEE Transactions on Information Theory* **4**(3) (1968), 462–467.

[6] S. Dasgupta, Learning polytrees, In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (1999), pp. 134–141.

[7] P. Domingos, Metacost: A general method for making classifiers cost-sensitive, In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, (1999), pp. 155–164.

[8] C. Drummond and R.C. Holte, Exploiting the cost (in) sensitivity of decision tree splitting criteria, In *ICML*, (2000), pp. 239–246.

[9] C. Elkan, The foundations of cost-sensitive learning, *International Joint Conference on Artificial Intelligence* **17**(1) (2001), 973–978.

[10] J.H. Friedman, Data Mining and Statistics: What's the connection? *Computing Science and Statistics* **29**(1) (1998), 3–9.

[11] N. Friedman, D. Geiger and M. Goldszmidt, Bayesian Network Classifiers, *Machine Learning* **29**(2-3) (1997), 131–163.

[12] N. Friedman and M. Goldszmidt, Learning Bayesian networks with local structure, In *Learning in Graphical Models, Springer Netherlands*, (1998), pp. 421–459.

[13] S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integrated Computer-Aided Engineering* **14**(3) (2007), 187–212.

[14] D. Heckerman, A. Mamdani and M.P. Wellman, Real-world applications of Bayesian networks, *Communications of the ACM 38*(3) (1995), 24–26.

[15] P. Langley, W. Iba and K. Thompson, An analysis of Bayesian classifiers, In *Proceedings, Tenth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press, **90**, (1992), 223–228.

[16] C.X. Ling and V.S. Sheng, Cost-sensitive learning, In *Encyclopedia of Machine Learning* (2010), pp. 231–235.

[17] C.X. Ling, Q. Yang, J. Wang and S. Zhang, Decision Trees with Minimal Costs, *ACM International Conference Proceeding Series 21st international conference on Machine learning*, Banff, Alberta, Canada, Article No. 69, ACM Press New York, NY, USA, (2004).

[18] S. Lomax and S. Vadera, A survey of cost-sensitive decision tree induction algorithms, *ACM Computing Surveys (CSUR)* **45**(2) (2013), 16.

[19] M. Meila and M.I. Jordan, Learning with Mixtures of Trees, *Journal of Machine Learning Research* (2000), 1–48.

[20] T.M. Mitchell, Does machine learning really work? *AI magazine* **18**(3) (1997), 11.

[21] E. Nashnush and S. Vadera, Cost-Sensitive Bayesian Network Learning Using Sampling, In *Recent Advances on Soft Computing and Data Mining, Springer International Publishing*, (2014), pp. 467–476.

[22] R.E. Neapolitan, Learning Bayesian networks, Upper Saddle River: Prentice Hall, (2004).

[23] M.J. Pazzani, C.J. Merz, P.M. Murphy, K. Ali, T. Hume and C. Brunk, Reducing Misclassification Costs, In *ICML*, (1994), pp. 217–225.

[24] J. Pearl, Embracing Causality in Formal Reasoning, In *AAAI*, (1988), pp. 369–373.

[25] C. Phua, V. Lee, K. Smith and R. Gayler, A comprehensive survey of data mining-based fraud detection research, *arXiv preprint arXiv: 1009.6119*, (2010).

[26] J.R. Quinlan, Induction of decision trees, *Machine Learning* **1**(1) (1986), 81–106.

[27] J. Rissanen, Modeling by shortest data description, *Automatica* **14**(5), (1978), 465–471.

[28] R. Santos-Rodríguez, D. García-García and J. Cid-Sueiro, Cost-sensitive classification based on Bregman divergences for medical diagnosis, In *Machine Learning and Applications, ICML*, (2009), pp. 551–556.

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

[30] V.S. Sheng and C.X. Ling, Thresholding for making classifiers cost-sensitive, In *Proceedings of the national conference on artificial intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press. **21**(1) (2006), pp. 476.

[31] V.S. Sheng and C.X. Ling, Roulette sampling for cost-sensitive learning, In *Machine Learning: ECML*, Springer, (2007), pp. 724–731.

[32] K.M. Thing, An Instance-Weighting Method to Induce Cost-Sensitive Decision Trees, *IEEE Transactions on Knowledge and Data Engineering* **14**(3) (2002), 659–665.

[33] I.H. Witten and E. Frank, *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, (2005).

[34] B. Zadrozny, J. Langford and N. Abe, A simple method for cost-sensitive learning, *IBM Technical Report RC22666*, (2003).

[35] B. Zadrozny, J. Langford and N. Abe, Cost-sensitive learning by cost-proportionate example weighting, In *Data Mining, ICDM, Third IEEE International Conference*, (2003), pp. 435–442.

[36] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang and A. Cichocki, Aggregation of sparse linear discriminant analysis for event-related potential classification in brain-computer interface, *International Journal of Neural Systems* **24**(1) (2014), 1450003.

[37] Y. Zhang, G. Zhou, Q. Zhao, J. Jin, X. Wang and A. Cichocki, Spatial-temporal discriminant analysis for ERP-based brain-computer interface, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **21**(2) (2013), 233–243.