Pattern Recognition Letters 45 (2014) 211–216

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Cost-sensitive Bayesian network classifiers ☆

Liangxiao Jiang [a,*], Chaoqun Li [b], Shasha Wang [a]

[a] Department of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] Department of Mathematics, China University of Geosciences, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 17 October 2013
Available online 30 April 2014

Keywords:
Cost-sensitive learning
Bayesian network classifiers
Instance weighting
Classification

ABSTRACT

Cost-sensitive learning has received increased attention in recent years. However, in existing studies, most of the works are devoted to make decision trees cost-sensitive and very few works discuss cost-sensitive Bayesian network classifiers. In this paper, an instance weighting method is incorporated into various Bayesian network classifiers. The probability estimation of Bayesian network classifiers is modified by the instance weighting method, which makes Bayesian network classifiers cost-sensitive. The experimental results on 36 UCI data sets show that when cost ratio is large, the cost-sensitive Bayesian network classifiers perform well in terms of the total misclassification costs and the number of high cost errors. When cost ratio is small, the advantage of cost-sensitive Bayesian network classifiers is not so obvious in terms of the total misclassification costs, but still obvious in terms of the number of high cost errors, compared to the original cost-insensitive Bayesian network classifiers.

1. Introduction

Classification is one of the most important tasks in data mining and machine learning. Traditional data mining and machine learning algorithms [19] are designed to yield classifiers that minimize the number of misclassification errors. In this case, the costs for different misclassification errors are considered to be equal. However, in real-world domains, many practical classification problems have different costs associated with different types of error. For example, in medical diagnosis, the cost of misclassifying a cancer patient to be a healthy person is significantly greater than the opposite type of error. So it is important to create a classifier that minimizes the total misclassification costs rather than the number of misclassification errors. This kind of classification task is called cost-sensitive classification [8,16,31]. In recent years, cost-sensitive classification has received increased attention.

In existing studies, most of the works are devoted to make decision trees cost-sensitive. A detailed survey of cost-sensitive decision tree induction algorithms can be found in Lomax and Vadera's paper [18]. However, a comprehensive study of cost-sensitive Bayesian network classifiers is rare in existing studies. Only a few works make naive Bayesian network classifiers cost-sensitive. For example, Gama [12] presents a cost-sensitive iterative Bayes. For another example, Chai et al. [1] specifically consider test-cost sensitive learning and propose a test-cost sensitive naive Bayes. For the third example, Fang [9] develops a cost-sensitive naive Bayes method which learns and infers the order relation from the training data and classifies the instance based on the inferred order relation.

In this paper, we focus our attention on cost-sensitive Bayesian network classifiers. In existing cost-sensitive studies, some meta-learning methods, such as MetaCost [6], instance weighting [23], thresholding [21], and sampling [13] etc., can be applied to make Bayesian network classifiers cost-sensitive. Among these, instance weighting is a simple, easy to understand and efficient method. Inspired by the success of cost-sensitive C4.5 [23] and weighted random forest [3], in this paper, the instance weighting method is incorporated into various Bayesian network classifiers, such as naive Bayes (NB), tree augmented Bayesian networks (TAN) [10], averaged one dependence estimators (AODE) [25], and hidden naive Bayes (HNB) [14], to make these Bayesian network classifiers cost-sensitive. The resulting classifiers are called cost-sensitive Bayesian network classifiers in this paper. The experimental results on a large number of UCI data sets show that these cost-sensitive Bayesian network classifiers achieve a substantial reduction in the total misclassification costs and the number of high cost errors.

The rest of this paper is organized as follows. In Section 2, some works related to cost-sensitive learning are introduced. In Section 3, we revisit several state-of-the-art Bayesian network classifiers and then incorporate the instance weighting method into them. In Section 4, we conduct a series of experiments on 36 UCI benchmark data sets to validate our proposed cost-sensitive Bayesian network classifiers. In Section 5, we draw conclusions and outline the main directions for our future work.

☆ This paper has been recommended for acceptance by G. Moser.

* Corresponding author. Tel./fax: +86 27 67883716.
E-mail address: ljiang@cug.edu.cn (L. Jiang).

http://dx.doi.org/10.1016/j.patrec.2014.04.017
0167-8655/

2. Related work

Cost-sensitive learning is generally categorized into two categories [16]. One is the direct method, which is to directly introduce and utilize misclassification costs into the learning algorithms. As a result, the learning algorithms are cost-sensitive in themselves. Some direct cost-sensitive learning algorithms include ICET [24], cost-sensitive iterative naive Bayes [12], and cost-sensitive decision trees [7,17].

The other category is called cost-sensitive meta-learning method. Cost-sensitive meta-learning converts existing cost-insensitive learning algorithms into cost-sensitive ones by pre-processing the training data or post-processing the output. Cost-sensitive meta-learning method can be further classified into two main subcategories: threshold adjusting and sampling.

Traditional cost-insensitive classifiers predict the class of a test instance in terms of a default, fixed threshold 0.5. While the threshold adjusting method uses a new threshold based on the misclassification costs to classify a test instance if the cost-insensitive classifiers can produce probability estimations. The threshold adjusting method includes MetaCost ([6]) and ET [21] etc.

Sampling is another kind method of cost-sensitive meta-learning. Because cost-sensitive learning is usually used to classify the imbalanced data [30,22,26,16,13]. In imbalanced data, the inequality between the number of instances in each of classes is severe and thus the class distribution is highly skewed. For example, to binary class data, the number of minority-class (positive class) instances is far less than the number of majority-class (negative class) instances. But in most cases, the minority-class is the class that interest us. Traditional classification algorithms, such as naive Bayes and C4.5, usually perform poorly on imbalanced data. Because these classifiers are designed to minimize the number of misclassification errors, and therefore classification rules that predict the minority-class tend to be far weaker than those that predict the majority-class, which results in that classifiers predict the majority-class much more often than the minority-class. In order to solve this problem, many sampling algorithms [2,26,29,11,13] are usually used to balance the class distribution of the training data and make the minority-class instances are well represented, and as a result, classifiers are allowed to place more importance on the minority-class.

Ling and Sheng [16] views instance weighting [23,28,3,29] as a sampling method. For example, cost-sensitive C4.5 [23] uses an instance weighting method to modify the split criterion of decision tree C4.5, which makes C4.5 cost-sensitive. For another example, Chen et al. [3] also present weighted random forest. Weighted random forest assigns a weight value to each class. In the tree induction procedure, the class weights are used to weight the Gini criterion to find split. In leaf nodes, the class weights are again taken into consideration. In fact, the instance weighting method works whenever the original cost-insensitive classifiers can accept instance weights directly. Bayesian network classifiers are the classifiers that can accept instance weights directly, so in the next section, we try to incorporate the instance weighting method into Bayesian network classifiers to make Bayesian network classifiers cost-sensitive.

3. Cost-sensitive Bayesian network classifiers

3.1. The instance weighting method in cost-sensitive C4.5

The central choice in the decision tree induction is selecting which attribute to split the training data at each non-terminal node in the tree. The information gain measure and its variants are generally used to select attributes. No matter the information gain measure or its variants are based on a measure called entropy [19]. Given a node t of a decision tree, let $N(t)$ be the number of instances in node t and $N_j(t)$ be the number of class j instances in node t, and the entropy of node t is defined as

$$Entropy(t) = -\sum_j P(j|t) log P(j|t), \tag{1}$$

where $P(j|t)$ denotes the probability that an instance is in class j given it falls into the node t. In minimum error tree induction systems, the probability $P(j|t)$ is estimated as the ratio of the total number of class j instances to the total number of instances in this node, which is defined as

$$P(j|t) = \frac{N_j(t)}{N(t)} = \frac{N_j(t)}{\sum_i N_i(t)}. \tag{2}$$

In Eq. (2), each class instance is assigned equal weight 1. In order to induce cost-sensitive trees, the intuition for cost-sensitive C4.5 [23] is to assign different weights for different class instances. The weight is proportional to the cost of misclassifying the class to which the instance belonged to. Let N be the total number of instances from the given training data, N_j be the number of class j instances in training data, and $C(j)$ is the cost of misclassifying a class j instance. The weight $w(j)$ of a class j instance is defined as

$$w(j) = C(j)\frac{N}{\sum_i C(i)N_i}. \tag{3}$$

It can be found from Eq. (3) that the weight $w(j)$ of a class j instance is proportional to the cost of misclassifying a class j instance, and the sum of all training instance weights still equal to N, i.e., $\sum_j w(j)N_j = N$. As a result, $P(j|t)$ is replaced by $P_w(j|t)$ which is estimated as the ratio of the total weight of class j instances to the total weight in node t:

$$P_w(j|t) = \frac{W_j(t)}{\sum_i W_i(t)} = \frac{w(j)N_j(t)}{\sum_i w(i)N_i(t)}, \tag{4}$$

where $W_j(t)$ denotes the sum of all class j instance weights in node t. So the standard greedy divide-and-conquer procedure for inducing minimum error trees can be used without modification, except that $W_j(t)$ is used instead of $N_j(t)$ in the computation of measures for selecting attributes in the tree growing process and the error estimation in the pruning process.

3.2. Incorporating the instance weighting method into Bayesian network classifiers

An instance x is described by an attribute vector $\langle a_1(x), a_2(x), \ldots, a_m(x)\rangle$, and $a_i(x)$ is the value of the ith attribute A_i of the instance. In many classification tasks, Bayesian network classifiers have shown impressive classification performance. Bayesian network classifiers estimate the class membership probabilities $P(c_j|x)$ ($c_j \in C$, C is the set of all class values) and classify the instance x to class c_j if $c_j \equiv argmax_{c_j} P(c_j|x)$. The probability $P(c_j|x)$ is calculated by using Bayes theorem:

$$P(c_j|x) = \frac{P(c_j)P(x|c_j)}{\sum_{c_j} P(c_j)P(x|c_j)}, \tag{5}$$

where $P(c_j)$ denotes the probability that the instance belongs to class c_j. Let N and N_j be the total number of instances and the number of class j instances in training data, and thus $P(c_j)$ is estimated as

$$P(c_j) = \frac{N_j + 1}{N + n_c}, \tag{6}$$

where n_c denotes the number of all class values. However, the full estimation of the conditional probability $P(x|c_j)$ is an NP-hard