Applied Soft Computing 69 (2018) 516–527

Contents lists available at ScienceDirect

Applied Soft Computing

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

Mixture of latent multinomial naive Bayes classifier

Nima Shiri Harzevili, Sasan H. Alizadeh *

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

ARTICLE INFO

Article history:
Received 3 September 2017
Received in revised form 14 February 2018
Accepted 13 April 2018
Available online 26 April 2018

MSC:
00-01
99-00

Keywords:
Naive Bayes
Mixture model
Latent variable
Multinomial distribution

ABSTRACT

Naive Bayes classifier has been extensively applied in various domains in the past few decades due to its simple structure and remarkable predictive performance. However, it is based on a strong assumption which confines its usage for many real-world applications; conditional independence of attributes given class information. In this paper, we propose *mixture of latent multinomial naive Bayes (MLMNB)* classifier as an extension of naive Bayes to relax the independence assumption. MLMNB incorporates a latent variable in a predefined Bayesian network structure to model the dependencies among attributes, yet avoids burden complexities of structural learning approaches. We theoretically prove that MLMNB automatically shrinks to naive Bayes classifier whenever conditional independence assumption holds. Expectation-maximization (EM) algorithm is modified for the parameter estimation. The experimental results on 36 datasets from the University of California, Irvine (UCI) machine learning repository show that MLMNB achieves a substantial predictive performance as compared with the state-of-the-art modifications of naive Bayes classifier, in terms of classification accuracy (ACC), conditional log-likelihood (CLL), and area under the ROC curve (AUC).

1. Introduction

Naive Bayes classifier (NB) [1] has received much attention in the past few decades on account of its simple structure. Naive Bayes introduces significant benefits, since it requires a few parameters for estimation in comparison with its competitors for classification, and does not require structural learning [2–4]. This simplicity fundamentally emerges according to a strong assumption in which each pair of attributes is conditionally independent given the information of the class variable. However, this assumption may not hold to be true in many real-world applications and introduces a bias in estimated probabilities subsequently which leads to a significant deterioration in performance of the classifier.

There has been much research works recently conducted to tackle this assumption. These efforts can be arranged into four major categories: (1) attribute selection [5–8]; (2) attribute weighting [9,10]; (3) local learning [11,12]; (4) structure extension [13,14]. First two categories mainly revolve around dealing with the attributes and their values while preserving the structure of the classifier. The third category deals with a subset of instances in the dataset to learn the classifier. The core motivation here is to find and organize a set of attributes and instances which optimizes the performance of the classifier. In contrast, the last category tries to model the dependencies among attributes through modifying the structure of the classifier. In this approach, in fact, simplicity is somehow compromised with performance. Augmenting the networks with new links or nodes are two possible alternatives to model the dependencies among attributes [14–16]. Having no constraints, this approach investigates the whole space of Bayesian networks to find out the best one, which is, of course, itself an NP-Hard problem [17]. Accordingly, many researchers have adopted to conjecture some restrictions on the search space to facilitate it and hold the balance between complexity and performance contemporary [18,16]. To our knowledge, most of the efforts in this area have been focused on incorporating some new links in the structure of the network, and except for few recent articles coping with latent variables as new nodes is largely untouched [13,19]. It is worth noting that likelihood equivalence [20] and over-fitting problems besides computational complexity still constitute major challenges of structure extension approach.

Bearing in mind the aforementioned issues, in this paper, we propose *mixture of latent multinomial naive Bayes (MLMNB)* classifier which does not cope with burden complexities of structural learning toward extending the structure of naive Bayes classifier for modeling conditional mutual dependency among attributes. A latent variable is incorporated into a predefined Bayesian network structure to full-fill this motivation. This variable is just standing

* Corresponding author.
E-mail addresses: nimashiri@qiau.ac.ir (N. Shiri Harzevili), sasan.h.alizadeh@qiau.ac.ir (S.H. Alizadeh).

https://doi.org/10.1016/j.asoc.2018.04.020
1568-4946/

beside the class variable as the parent of all other attributes. In fact, MLMNB is a compromise between general Bayesian networks in which the structure is learned without limitation, and the naive Bayes, in which the structure is already determined to be naive. In order to estimate the parameters of our proposed model, EM algorithm [21] is slightly modified to consider small possibilities for unobserved events in training data. The contributions of MLMNB with respect to the literature on this context can be summarized as follows:

- Predefined network structure saves us from structural learning challenges specially burden computational complexities.
- The location of the latent variable is justified in accordance with conventions of Bayesian networks literature so as to model the mutual dependency among attributes.
- Establishing parameter estimation based on EM algorithm guarantees us with a robust estimation and convergence to the local optimum.

We establish extensive experiments to evaluate MLMNB and compare it with state-of-the-art extensions of naive Bayes classifier. The results demonstrate that MLMNB obtained remarkable performance in terms of classification accuracy (ACC), conditional log-likelihood (CLL), and area under the ROC curve (AUC) compared to state-of-the-art extensions of naive Bayes classifier.

Organization of this paper is as follows. Section 2 is devoted to literature review for relaxing naive Bayes independence assumption. Section 3 reviews the naive Bayes classifier basic concepts. In Section 4, we present our proposed method followed by the parameter estimation. Section 5 provides a wide range of experiments and corresponding results. Finally, Section 6 concludes the paper and defines some future research directions.

2. Related works

The naive Bayes classifier is a well-known probabilistic classifier which has applied in many real-world problems [2–4,22]. However, its independence assumption may not hold in real-world data generation mechanisms because attributes may have a high dependency. In order to model dependencies among attributes, one may use Bayesian networks [16,23] which are promising tools in this context. Nevertheless, optimal structural learning in Bayesian networks has proved to be an NP-hard problem [17]. Burden complexities of structural learning have motivated many researchers to enhance naive Bayes performance. The approaches in this context can be classified into four main categories: (1) attribute selection; (2) attribute weighting; (3) local learning; (4) structure extension.

2.1. Attribute selection

The Attribute selection approach selects most important attributes to enhance the performance of naive Bayes classifier. The efficient selection of attributes that have the most contribution to the prediction can be considered as the main challenge in this approach. Hence, Numerous researchers have conducted studies for this purpose. Jiang et al. [5] presented an algorithm, named *evolutional naive Bayes (ENB)*. By performing an evolutional search process, their algorithm selects a subset of attributes through whole attributes space. The naive Bayes classification accuracy employed as the fitness function to assess the selected attributes subsets.

Langley et al. proposed an algorithm named *selective Bayesian classifier (SBC)* [6]. SBC considers classification accuracy as its attribute selection criteria and always selects attributes that have the most contribution to classification. The problem with SBC is that it may fall into a local optimum because its search strategy is based on a greedy search strategy. Hence, Jiang et al. [24] proposed an algorithm based on SBC approach named *randomly selected naive Bayes (RSNB)*, in which it improves naive Bayes classifier performance by exploiting random search through attributes space.

Apart from the approaches proposed to enhance naive Bayes classifier performance, there are some methods used in general attribute selection problems. For Instance, Kohavi and John [7] explored the optimality and relevance of attribute selection techniques and presented a wrapper method for attribute subset selection. Ratanamahatana and Gunopulos [8] proposed decision trees based method for attribute selection. However, the major limitation of these methods is the amount of time needed for the search operations.

2.2. Attribute weighting

In contrast to attribute selection methods which eliminate least important attributes, weighting methods use weights to prioritize attributes according to their contribution to the prediction. For example, Taheri et al. [9] used weighted naive Bayes to relax conditional independence of naive Bayes classifier. Their proposed method named *attribute weighted naive Bayes (AWNB)*. They assigned a weight for each attribute in the dataset which depends on the number of labels of the class variable. Also, an objective function modeled on naive Bayes classifier network structure to optimize attributes weights.

Hall [10] proposed a method for relaxing independence assumption based on adjusting weights for each attribute. This filtering method is based on decision-tree approach.

A simple weighted naive Bayes which relax conditional independence assumption with learning local models proposed by Frank et al. [25]. They tried to compare locally weighted naive Bayes with locally weighted regression methods.

Zaidi et al. [26] investigated that weighting of attributes tackles naive Bayes independence assumption. Therefore, they proposed a weighting algorithm named *WANIBIA*, in which weights minimum the negative log-likelihood and objective function of squared error.

Zhang and Sheng [27] conducted a set of experiments to indicate that how to learn weights of a weighted naive Bayes classifier in order to increase its ranking. They used many methods like hill-climbing, Markov chain Monte-Carlo (MCMC), gain-ratio, and the combination of Markov chain with the gain-ratio method.

Jiang et al [28] proposed an improved version of naive Bayes named *deep feature weighting (DFW)*. The idea behind DFW is that it incorporates attribute weights into the conditional probability distribution. Given a training set, it deeply computes the attribute weights.

2.3. Local learning

In local learning approach, given a training set, instead of building naive Bayes model on the whole training set, naive Bayes is built on a subset of the training set. The basic idea behind local learning approach is that dependencies among attributes through the subset of the training set are weaker compared to the whole training set. As a result, trained naive Bayes based on the subset of training set obtains better classification accuracy. According to the studies which have conducted by Kohavi [11], datasets with large sample sizes do not affect the classification accuracy of naive Bayes classifier.

An example of local tree-based approach is an algorithm named *naive Bayes tree (NBTree)* presented by Kohavi [11]. It follows a hybrid approach in which it synthesizes decision tree with naive Bayes classifier. The learning procedure of NBtree is similar to C4.5. However, evaluation criteria to split attributes is different. Naive Bayes is built for leaf nodes and the classification task is done by