Neural Comput & Applic (2014) 24:995–1002
DOI 10.1007/s00521-012-1329-z

ORIGINAL ARTICLE

Attribute weighted Naive Bayes classifier using a local optimization

Sona Taheri · John Yearwood · Musa Mammadov · Sattar Seifollahi

Received: 6 February 2012 / Accepted: 19 December 2012 / Published online: 5 January 2013

Abstract The Naive Bayes classifier is a popular classification technique for data mining and machine learning. It has been shown to be very effective on a variety of data classification problems. However, the strong assumption that all attributes are conditionally independent given the class is often violated in real-world applications. Numerous methods have been proposed in order to improve the performance of the Naive Bayes classifier by alleviating the attribute independence assumption. However, violation of the independence assumption can increase the expected error. Another alternative is assigning the weights for attributes. In this paper, we propose a novel attribute weighted Naive Bayes classifier by considering weights to the conditional probabilities. An objective function is modeled and taken into account, which is based on the structure of the Naive Bayes classifier and the attribute weights. The optimal weights are determined by a local optimization method using the quasisecant method. In the proposed approach, the Naive Bayes classifier is taken as a starting point. We report the results of numerical experiments on several real-world data sets in binary classification, which show the efficiency of the proposed method.

Keywords Classification · Naive Bayes · Attribute weighting · Local optimization

1 Introduction

Classification is the task of identifying the class labels for instances based on a set of attributes. Learning accurate classifiers from pre-classified data is a very active research topic in machine learning and data mining. Classification learning is the process of predicting a discrete class label $C \in \{C_1, \ldots, C_m\}$ for a test instance $\mathcal{X} = (X_1, \ldots, X_n)$.

One of the most effective classifiers [27] is the Bayesian Network (BN) introduced by Pearl [20]. A BN is composed of a network structure and its conditional probabilities. The structure is a directed acyclic graph where the nodes correspond to domain variables and the arcs between nodes represent direct dependencies between the variables. The classifier represented by the BN can be expressed as:

$$\arg \max_{1 \le k \le m} P(C_k|\mathcal{X}) = \arg \max_{1 \le k \le m} \frac{P(C_k)P(\mathcal{X}|C_k)}{P(\mathcal{X})}; \tag{1}$$

this rule is called Bayes rule. We can see that for each class, the denominator of Eq. (1) is the same and it will not interfere in classification. So, the BN classifier can be rewritten as:

$$\arg \max_{1 \le k \le m} P(C_k|\mathcal{X}) \propto \arg \max_{1 \le k \le m} P(C_k)P(\mathcal{X}|C_k). \tag{2}$$

However, accurate estimation of $P(\mathcal{X}|C_k)$ is non-trivial. It has been proved that learning an optimal BN is NP-hard [4, 10]. In order to avoid the intractable complexity for learning the BN, the Naive Bayes (NB) classifier has been used. In the NB [15, 22], attributes are conditionally independent given the class. Compared to other supervised

S. Taheri · J. Yearwood · M. Mammadov
School of Science, Information Technology and Engineering,
University of Ballarat, Ballarat, VIC 3353, Australia
e-mail: sonataheri@students.ballarat.edu.au

M. Mammadov
National ICT Australia, VRL, Melbourne, Australia

S. Seifollahi (✉)
School of Civil, Environmental and Mining Engineering,
University of Adelaide, Adelaide, SA 5005, Australia
e-mail: sattarseif@gmail.com

machine learning methods, the NB classifier is perhaps one of the simplest, yet surprisingly powerful, techniques to construct predictive models from labeled training sets. The NB classifier is important for several reasons. It is easy to construct and implement because the structure is given a priori (no structure learning procedure is required) and it needs only to compile a table of class probabilities and conditional probabilities from the training instances. Therefore, it may be readily applied to huge data sets. It is easy to interpret, and even unskilled users in classifier technology can understand why it is making the classification it makes. Finally, it may not be the best possible classifier in any particular application, but it can usually be relied on to be robust and to do quite well [5].

A sample of the NB classifier with n attributes is depicted in Fig. 1. The NB classifies an instance $\mathcal{X} = (X_1, \ldots, X_n)$ by selecting

$$\arg \max_{1 \leq k \leq m} P(C_k|\mathcal{X}) \propto \arg \max_{1 \leq k \leq m} P(C_k) \prod_{i=1}^{n} P(X_i|C_k). \quad (3)$$

However, the attribute independence assumption made by the NB classifier harms its classification performance when it is violated in reality [12]. In order to relax the attribute independence assumption of the NB classifier while at the same time retaining its simplicity and efficiency, researchers have proposed many effective methods. These methods have been proposed in order to improve the performance of the Naive Bayes classifier by alleviating the attribute independence assumption. Among the variety of works, semi-NB classifiers [14, 16, 19] show significant improvements in the NB classifier by using the selected subset of attributes. The Tree Augmented Naive Bayes (TAN) [8] utilizes the tree structure to find relations between attributes. The Super Parent [13] uses the same representation as the TAN, but utilizes leave-one-out cross validation error as a criterion to add a link. The Improved Naive Bayes (INB), proposed by Taheri et al. [22], uses conditional probabilities for finding the dependencies between attributes.

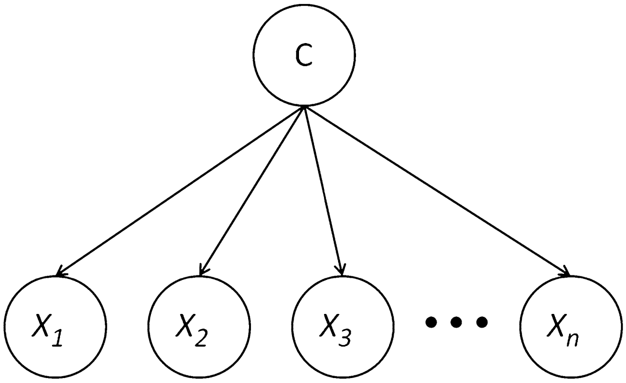

Fig. 1 Naive Bayes

Another way to mitigate its attributes independence assumption is assigning weights to important attributes in the classification. Since attributes do not play the same role in many real-world applications, some of them are more important than others. A natural way to extend the NB classifier is to assign a weight to each attribute. This is the main idea of the algorithm called attribute weighted NB. Much work to evaluate the importance of attributes has been done in recent years [9, 11, 18, 26, 31, 32]. Jiang and Zhang [11] developed the improved NB called weightily averaged one-dependence estimators based on the idea of a model introduced by Webb et al. [25]. Hall [9] presented a simple filter method for setting attribute weights to use in the NB classifier. The assumption made is that the weight assigned to a predictive attribute should be inversely related to the degree of dependency it has on other attributes. More recently, Wu and Cai [26] used differential evolution algorithms to determine the weights of attributes in the model introduced by Hall [9], and then, they used these weights in the developed weighted NB classifier. The paper [31] investigates how to learn a weighted NB classifier with accurate ranking from data, or more precisely, how to learn the weights of a weighted NB classifier to produce accurate ranking.

In this paper, we propose a new attribute weighted NB classifier, called AWNB, which assigns more than one weight for each attribute. The number of weights for each attribute is considered as the number of class labels. These weights are written in the form of powers to the conditional attribute-class probabilities. An objective function is constructed based on the NB structure and the attribute weights. The weights, then, are determined by using a local optimization method, which here is the quasisecant method [2]. The initial weights for the quasisecant method are set to unity; this means that the NB classifier is taken as an initial point. More precisely, our aim is improving the NB classifier by modeling a proper objective function and optimizing the attribute weights. To find a global solution, one can also apply a global optimization; however, the complexity of the problem will increase.

Most of data sets in real-world applications often involve continuous attributes. The most well-known attempt for improving the performance of the NB with continuous attributes is the discretization of the attributes into intervals, instead of using the default option to utilize the normal distribution to calculate probabilities. Numerous discretization methods have been examined for the NB learning [17, 24, 28, 29, 30]. The performance of the NB classifier significantly improves when attributes are discretized using an entropy-based method [6]. In this paper, we use Fayyad and Irani's discretization method [7]; a method based on a minimal entropy heuristic. We also apply the discretization algorithm using sub-optimal agglomerative clustering algorithm which is an efficient discretization method, recently introduced in [28].

The rest of the paper is organized as follows. In the next section, we present a brief review of the quasisecant method. Section 3 reviews briefly two different discretization methods, Fayyad and Irani's method and sub-optimal agglomerative clustering-based method, respectively. The leaning of the proposed method is illustrated in Sect. 4, which follows by the experiments and discussion on the experiments in Sect. 5. Section 6 concludes the paper followed by a few directions for future work.

2 A brief review of the quasisecant method

The quasisecant method [2] is a local method for solving nonsmooth, nonconvex optimization problems. In general, this method is applicable for solving the following unconstrained minimization problem:

$$\textit{minimize} \quad f(x) \tag{4}$$

where $x \in \mathbb{R}^d$, and the objective function f is assumed to be locally Lipschitz.

Formally, quasisecants are defined as follows. Let $S = \{x \in R^d : \|x\| = 1\}$ be the unit sphere. A vector $v \in R^d$ is called a quasisecant of the function f at the point x in the direction $g \in S$ with the length $h > 0$ iff

$$f(x+hg) - f(x) \leq h\langle v, g\rangle.$$

Here, $\langle v, g\rangle$ is the inner product of vectors $v, g \in R^d$. The above inequality is called a quasisecant inequality. Quasisecants provide overestimation to the function f in some neighborhood of a point x. There are many vectors v satisfying the quasisecant inequality. We consider only those which provide approximation to the function. Subgradient-related quasisecants introduced in [2] provide such approximations and they converge to tangents of the graph of the function f.

Any quasisecant is defined with respect to a given direction $g \in S$ and with given length $h > 0$. The choice of h allows one to compute descent directions with different lengths. Therefore, one can compute descent directions even from some shallow local minimizers using quasisecants. This observation makes the quasisecant method applicable to nonconvex problems and computes a "deep" local minimizers.

On the other hand, the quasisecant method uses a bundle of quasisecants at a given point to compute descent directions which makes it similar to the well-known bundle methods in nonsmooth optimization. Therefore, it is applicable to solve nonsmooth optimization problems. Results presented in [2] demonstrate that the quasisecant method is efficient and robust method for solving nonsmooth, nonconvex optimization problems.

3 Discretization methods

In order to apply the NB classifier to data sets with continuous attributes, one should first discretize the attributes. Discretization is a process which transforms continuous numeric values into discrete ones. In this paper, we apply two different methods in our experiments to discretize the attributes. The first one is the Fayyad and Irani's discretization method, and the second one is discretization algorithm using sub-optimal agglomerative clustering proposed by Yatsko et al. [28].

3.1 Fayyad and Irani's method

The Fayyad and Irani's Discretization method is based on a minimal entropy heuristic, and it uses the class information entropy of candidate partitions to select bin boundaries for discretization. In this subsection, we give a brief review to this method, and details can be found in [7].

Let us consider a given set of instances $\mathbf{X}$, an attribute X, and a partition boundary T, the class information entropy of the partition induced by T, denoted $E(X, T; \mathbf{X})$ is given by

$$E(X, T; \mathbf{X}) = \frac{|\mathbf{X}^{(1)}|}{|\mathbf{X}|} Ent(\mathbf{X}^{(1)}) + \frac{|\mathbf{X}^{(2)}|}{|\mathbf{X}|} Ent(\mathbf{X}^{(2)}),$$

where $\mathbf{X}^{(1)} \subset \mathbf{X}$ be the subset of instances in $\mathbf{X}$ with X-values not exceeding T and $\mathbf{X}^{(1)} = \mathbf{X} - \mathbf{X}^{(1)}$. Let there be m classes $C_1, \ldots, C_m$. Let $P(C_i, \mathbf{X})$ be the proportion of instances in $\mathbf{X}$ that have the class C_i. The class entropy of a subset $\mathbf{X}$ is defined as:

$$Ent(\mathbf{X}) = -\sum_{i=1}^{m} P(C_i, \mathbf{X}) \lg(P(C_i, \mathbf{X})),$$

where the logarithm may be to any convenient base. When the base is 2, $Ent(\mathbf{X})$ measures the amount of information needed, in bits, to specify the classes in $\mathbf{X}$.

For a given attribute X, the boundary $T_{\min}$ which minimizes the entropy function over all possible partition boundaries is selected as a binary discretization boundary. This method can then be applied recursively to both of the partitions induced by $T_{\min}$ until some stopping condition is achieved, thus creating multiple intervals on the attribute X.

Fayyad and Irani make use of the minimal description length principle to determine a stopping criteria for their recursive discretization strategy. Recursive partitioning within a set of values $\mathbf{X}$ stops if

$$Gain(X, T; \mathbf{X}) < \frac{\lg_2(N-1)}{N} + \frac{\Delta(X, T; \mathbf{X})}{N},$$

where N is the number of instances in the set $\mathbf{X}$, and

$$Gain(X, T; \mathbf{X}) = Ent(\mathbf{X}) - E(X, T; \mathbf{X}),$$

$$\Delta(X, T; \mathbf{X}) = \lg_2(3^m - 2) - [m.Ent(\mathbf{X}) - m_1.Ent(\mathbf{X}^{(1)}) - m_2 Ent(\mathbf{X}^{(2)})],$$

and m_i is the number of class labels represented in the set $\mathbf{X}^{(i)}$. Since the partitions along each branch of the recursive discretization are evaluated independently using this criteria, some areas in the continuous spaces will be partitioned very finely whereas others (which have relatively low entropy) will be partitioned coarsely.

3.2 Sub-optimal agglomerative clustering-based method (SOAC)

In this section, we give a brief description of the discretization algorithm SOAC. Details of this algorithm can be found in [28]. Consider a finite set of points $\mathbf{X}$ in the n-dimensional space R^n, that is $\mathbf{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$, where $\mathcal{X}_i \in R^n, i = 1, \ldots, N$. Assume the sets $A_j, j = 1, \ldots, k$ be clusters, and each cluster A_j can be identified by its centroid $\mathcal{X}_j \in R^n, j = 1, \ldots, k$. The discretization algorithm SOAC proceeds as follows.

4 Learning the proposed attribute weighted Naive Bayes using optimization

Good attribute weighting can eliminate the effects of noisy or irrelevant attributes. In this section, we propose a weighting procedure, in which each conditional attribute-class probability has its own power as a weight. The number of weights for each attribute is equal to the number of class labels. The idea of our weighting method is similar to the works in [26, 32], however constructing a proper objective function and utilizing the new weighting procedure are different from the existing methods.

```
Step 1. Set $k = N$, and a small value of parameter $\theta$, $0 < \theta < 1$. Sort values of the current feature in the ascending order. Each continuous feature requiring discretization is treated in turn.
Step 2. Calculate the center of each cluster, $\mathcal{X}_j = \sum_{\mathcal{X} \in A_j} \frac{\mathcal{X}}{|A_j|}$, $j = 1, ..., k$ and the error $E_k$ of the cluster system approximating set $\mathbf{X}$, $E_k = \sum_{j=1}^{k} \sum_{\mathcal{X} \in A_j} \|\mathcal{X}_j - \mathcal{X}\|^2$.
Step 3. Merge in turn each cluster with the next tentatively. Calculate the error increase $E_{k-1} - E_k$ after each merge and choose the pair of clusters giving the least increase. Merge these two clusters permanently. Set $k = k - 1$.
Step 4. If $E_k \geq \theta E_1$, then stop, otherwise go to Step 2.
```

Algorithm 1: Discretization Algorithm SOAC

Let us consider $D = \{\mathcal{X}_i, C_i\}, 1 \leq i \leq N$, where N is the number of instances and $C_i \in \{C_1, \ldots, C_m\}$. $\mathcal{X}_i$ is an n-dimensional vector, $\mathcal{X}_i = (X_{i1}, X_{i2}, \ldots, X_{in}), n$ is the number of attributes, and C_i is the class label. In this paper, we consider the binary classification and assume that the two classes are 1 and -1. Then, for each attribute, we define two weights, one corresponding to the class $C_1 = 1$ and another to the class $C_2 = -1$. By considering two weights for each attribute, the attribute weighted NB classifies an instance $\mathcal{X}_i$ by selecting:

$$\arg \max_{1 \leq k \leq 2} P(C_k) \prod_{j=1}^{n} P(X_{ij}|C_k)^{w_{jk}}. \tag{5}$$

In Eq. (5), there are two alternatives for k in w_{jk}. We denote these cases by w_j and $\overline{w}_j$ if $\mathcal{X}_i$ is allocated to the real class and its counterpart, respectively. Considering that C_k is the real class of $\mathcal{X}_i$, the value of $P(C_k|\mathcal{X}_i)$ is expected to be greater than the value of $P(\overline{C}_k|\mathcal{X}_i)$ for the majority of instances, $i = 1, \ldots, N$, where $\overline{C}_k = -C_k$. Then, it is quite natural that the value of

$$P(C_k) \prod_{j=1}^{n} P(X_{ij}|C_k)^{w_j} \tag{6}$$

should be maximized, while the value of

$$P(\overline{C}_k) \prod_{j=1}^{n} P(X_{ij}|\overline{C}_k)^{\overline{w}_j} \tag{7}$$

to be minimized. Therefore, one possible objective function for the NB classifier, by considering the weights for attributes, can be written as follows:

$$\text{maximize}\ f(w) = \sum_{i=1}^{N} \frac{P(C_k) \prod_{j=1}^{n} P(X_{ij}|C_k)^{w_j} - P(\overline{C}_k) \prod_{j=1}^{n} P(X_{ij}|\overline{C}_k)^{\overline{w}_j}}{P(C_k) \prod_{j=1}^{n} P(X_{ij}|C_k)^{w_j} + P(\overline{C}_k) \prod_{j=1}^{n} P(X_{ij}|\overline{C}_k)^{\overline{w}_j}}, \tag{8}$$

where $w = (w_1, \overline{w}_1, w_2, \overline{w}_2, \ldots, w_n, \overline{w}_n)$ is a set of unknown variables (attribute weights). The objective function (8) is similar to the objective function presented in [23]. The weights in (8) are considered as positive numbers. Also, we put an upper limit for these weights to prevent large numbers. So, we maximize the above objective function over a hyper box $[a, b]$. Therefore, the problem (8) can be formulated as a constrained optimization problem:

$$\text{minimize} \quad -f(w) \tag{9}$$

$$\text{subject to } w_i, \overline{w}_i \in [a, b], 1 \leq i \leq n.$$

Different methods can be applied to transfer the problem (9) to an unconstrained optimization [21]. One of the well-known methods is the penalty method, which is used here. To find the weights in (9), a local optimization method is applied, which here is the quasisecant method presented in Sect. 2. The NB classifier is taken as a starting point for the quasisecant method. More precisely, we initialize all the weights to unity, and then, we use the quasisecant method to find the attribute weights for further improvement.

In other words, we search for an optimal classifier starting with the NB classifier. It is noted that a global optimization is also applicable to find the global solution of the problem (9), but the complexity of the problem will increase.

Table 1 A brief description of data sets

Data sets	# Instances	# Attributes
Breast cancer	699	10
Congressional voting records	435	16
Credit approval	690	15
Diabetes	768	8
Haberman's survival	306	3
Heart disease	303	14
Ionosphere	351	34
Liver disorders	345	6
Phoneme CR	5,404	5
Sonar	208	60
Spambase	4,601	57
Fourclass	862	2
German.numer	1,000	24
Splice	3,175	60
Svmguide1	7,089	4
Svmguide3	1,284	21

5 Experiments

5.1 Data collections

This paper studies 16 benchmark data sets taken from the literature. A brief description of the data sets is given in Table 1. The detailed description of the first eleven data sets used in this experiments can be found in the UCI repository of machine learning databases [1], and the last five data sets are downloadable on the tools page of the LIBSVM [3]. These data sets have been analyzed quite frequently by the current data mining approaches. Another reason for selecting these data sets was that conventional approaches have analyzed them with variable success.

5.2 Results and discussion

We conduct empirical comparison for the Naive Bayes (NB), the Tree Augmented Naive Bayes (TAN), the improved Naive Bayes (INB) proposed by Taheri et al. [22], and the attribute weighted Naive Bayes (AWNB) in terms of accuracy. The structure of the TAN and the INB is originated from the structure of the NB, in which each attribute has at most one augmenting edge pointing to it. The relations between attributes in the TAN are found by using the tree procedure [8], while the INB uses conditional probabilities for finding the correlations [22].

We discretize the values of continuous attributes in data sets using two different methods. In the first one, we apply

Table 2 Test set accuracy averaged over 50 runs for data sets using Fayyad and Irani's discretization method

Data sets	NB	TAN	INB	AWNB
Breast cancer	97.18	96.52	97.63	**97.74**
Congressional voting records	90.11	93.21	93.47	**94.24**
Credit approval	86.10	84.78	86.72	**86.91**
Diabetes	74.56	75.14	**76.06**	75.98
Haberman's survival	75.09	74.41	**77.03**	76.83
Heart disease	82.93	81.23	83.36	**85.57**
Ionosphere	88.62	**89.77**	88.98	89.61
Liver disorders	63.26	63.18	64.89	**65.79**
Phoneme CR	77.56	**78.31**	77.71	78.22
Sonar	76.32	**76.47**	76.41	76.39
Spambase	90.41	89.78	**92.87**	92.43
Fourclass	77.46	77.61	78.61	**78.91**
German.numer	74.50	73.13	75.91	**76.79**
Splice	95.43	94.87	**95.91**	95.88
Svmguide1	92.39	91.61	**94.04**	93.97
Svmguide3	81.23	82.47	84.98	**87.44**

NB, Naive Bayes; TAN, Tree Augmented Naive Bayes; INB, Improved Naive Bayes; AWNB, Attribute weighted Naive Bayes

Table 3 Test set accuracy averaged over 50 runs for data sets using discretization algorithm SOAC

Data sets	NB	TAN	INB	AWNB
Breast cancer	96.12	95.60	96.45	**96.56**
Congressional voting records	90.11	91.42	91.47	**94.52**
Credit approval	85.85	84.98	**86.85**	86.79
Diabetes	75.78	75.90	**77.68**	77.53
Haberman's survival	74.66	**76.08**	75.33	75.91
Heart disease	78.62	77.37	79.31	**82.41**
Ionosphere	85.92	**86.18**	85.97	86.11
Liver disorders	65.82	65.73	66.51	**66.85**
Phoneme CR	77.01	78.53	79.36	**79.65**
Sonar	75.09	75.76	**75.83**	75.69
Spambase	89.30	89.04	92.30	**92.44**
Fourclass	78.58	79.52	79.70	**79.76**
German.numer	74.61	74.01	75.23	**75.81**
Splice	92.12	**93.04**	92.39	92.87
Svmguide1	95.61	94.91	**97.54**	97.43
Svmguide3	77.25	79.99	80.85	**81.23**

NB, Naive Bayes; TAN, Tree Augmented Naive Bayes; INB, Improved Naive Bayes; AWNB, Attribute Weighted Naive Bayes

Fayyad and Irani's discretization method [7]. The second one is the discretization algorithm SOAC [28].

For each method, we run 50 trials and then the average accuracy over the 50 runs are calculated. The accuracy of the methods in each run is calculated using 10-fold cross-validation with random orders of data records in partitioning training and test data sets to have more reliable results. More precisely, each fold contained 10% of the data set randomly selected (without replacement). For consistent comparison, the same folds, including the same training and test data sets, are used in implementing the methods.

The penalty parameter is chosen as $\mu = 10^6$. We set the lower and upper limits in (9) as $a = 0.1$, $b = 10$.

Table 2 presents the average accuracy obtained by the NB, the TAN, the INB, and the AWNB on 16 data sets, where continuous attributes are discretized by applying Fayyad and Irani's method [7]. The results presented in this table demonstrate that the accuracy of the proposed method (AWNB) is much better than that of the NB in all data sets; the bold font shows the highest accuracy in each data. It is also shown a higher accuracy of the AWNB, in general, compared to the results obtained by the TAN and the INB. The proposed method outperforms both methods (the TAN and the INB) in most of the data sets, and the accuracy of this method is slightly less or almost ties with the TAN and the INB in a few cases.

The results of the average accuracy obtained by the methods on 16 data sets using discretization algorithm SOAC are reported in Table 3. The results show that the accuracy obtained by the proposed method (AWNB) in all data sets is higher than those of the NB. The accuracy of the AWNB is also higher than those of the TAN and the INB in most of data sets, and the accuracy of the AWNB

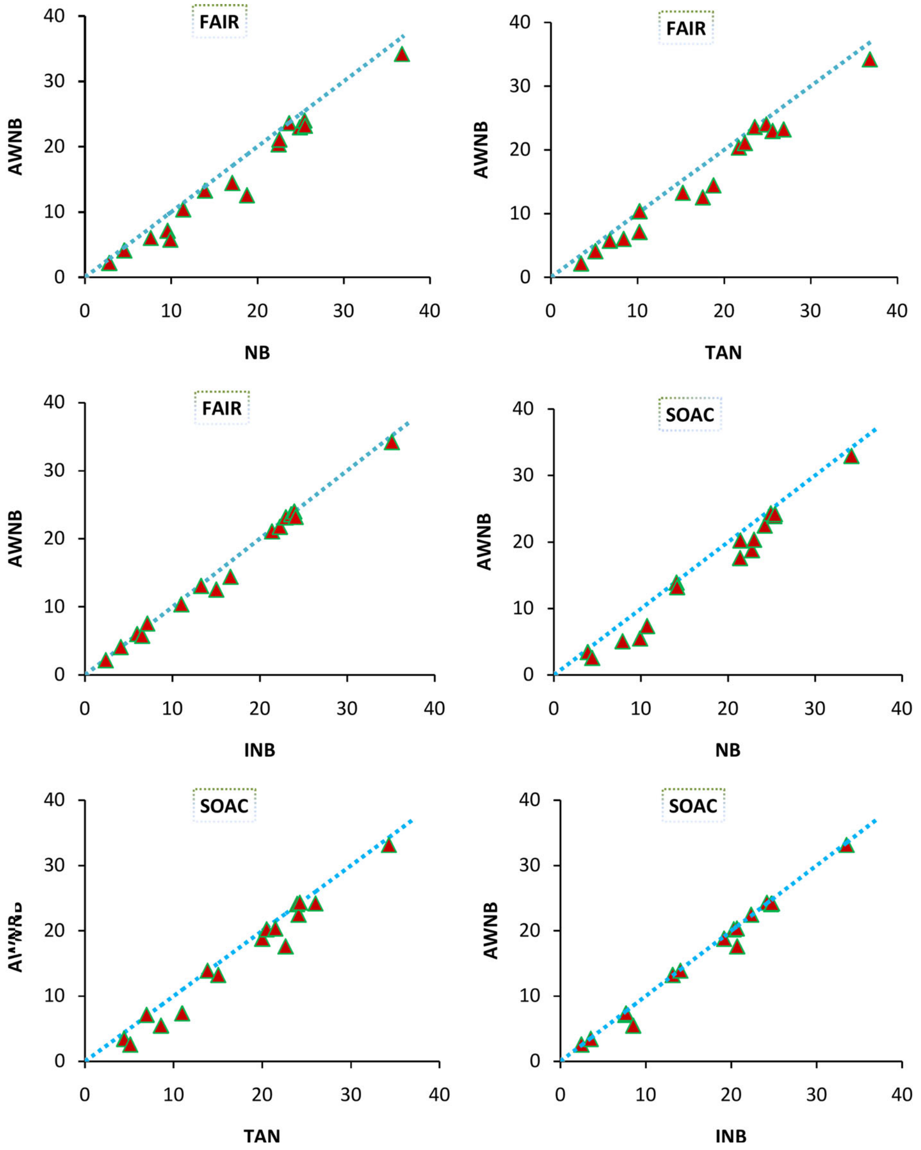

Fig. 2 Scatter plot comparing the average miss-classifications of the proposed method (AWNB) with Naive Bayes (NB), Tree Augmented Naive Bayes (TAN), Improved Naive Bayes (INB) using Fayyad and Irani's discretization method (FAIR) and Sub-Optimal Agglomerative Clustering-based method (SOAC)

almost ties with those of the TAN and the INB in a few cases.

Figure 2 shows the scatter plots comparing the average miss-classifications of the proposed attribute weighted Naive Bayes, AWNB, with those of the NB, the TAN, and the INB using two different discretization methods. In these plots, each point represents a data set, where the horizontal axis shows the percentage of miss-classifications according to the NB, the TAN and the INB, and the vertical axis is the percentage of miss-classification according to the proposed method, AWNB. Therefore, points below the diagonal line correspond to data sets where the AWNB performs better, and points above the diagonal line correspond to data sets where the other mentioned methods perform better.

According to the results explained above, the proposed attribute weighted Naive Bayes, AWNB, works well in that it improves the results of the NB classifier. Moreover, in general, it outperforms the TAN and the INB so that its accuracy in most of the data sets is higher than those of the the TAN and the INB. In a few cases, the TAN and the INB perform slightly better than the proposed method, and the results are acceptable as the two methods are also developments on the NB classifier.

The complexities of the methods are not compared in this work, since different softwares are used to implement the methods. The proposed method is coded in Matlab, while others are coded in Fortran. It is clear that the complexity of the proposed method is higher than the others due to the complexity of the optimization procedure. A global optimization is also applicable to determine the weights for the attributes. Although it may cause a better accuracy, a higher level of computational effort is required.

6 Conclusions

In this paper, we proposed a classifier based on attribute weighted Naive Bayes, AWNB. A novel weighting method for attribute weighted NB classifier was introduced, in which for each attribute we used more than one weight depending on the number of class labels. An objective function consisting of the attribute weights based on the structure of the NB classifier was then modeled to optimize the attribute weights. This objective function was optimized by a local optimization using the quasisecant method. The initial values in the quasisecant method were chosen as one, meaning that the NB classifier was taken as a starting point.

We carried out a number of experiments on some data sets obtained from the UCI machine learning repository and LIBSVM. The numerical results demonstrated that the proposed method has positive impact on the NB accuracy as expected. How this attribute weighting for the NB classifier performs in multiclass data sets remains an important question for future work.

References

1. Asuncion A, Newman D (2007) UCI machine learning repository. School of Information and Computer Science, University of California http://www.ics.uci.edu/mlearn/MLRepository.html
2. Bagirov AM, Nazari Ganjehlou A (2010) A quasisecant method for minimizing nonsmooth functions. Optim Methods Soft 25(1):3–18
3. Chang C, Lin C (2001) LIBSVM: A library for support vector machines Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
4. Chickering DM (1996) Learning Bayesian networks is NP-complete. In: Fisher D, Lenz H (eds) Learning from data: artificial intelligence and statistics V. Springer, Berlin, pp 121–130
5. Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29:103–130
6. Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Proceedings of the 12th international conference on machine learning, pp 194–202
7. Fayyad UM, Irani KB (1993) On the handling of continuous-valued attributes in decision tree generation. Mach Learn 8:87–102
8. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifierss. Mach Learn 29:131–163
9. Hall M (2007) A decision tree-based attribute weighting filter for Naive Bayes. Knowl Based Syst 20:120–126
10. Heckerman D, Chickering DM, Meek C (2004) Large-sample learning of Bayesian networks is NP-Hard. J Mach Learn Res 5:1287–1330
11. Jiang L, Zhang H (2006) Weightily averaged one-dependence estimators. In: Proceedings of the 9th biennial pacific rim international conference on artificial intelligence, Guilin, China, pp 970–974
12. Jiang L, Wang D, Cai Z, Yan X (2007) Survey of improving Naive Bayes for classification. In: Proceedings of the 3rd international conference on advanced data mining and applications, 4632, Springer, Berlin, pp 134–145
13. Keogh EJ, Pazzani MJ (1999) Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proceedings of international workshop on artificial intelligence and statistics, pp 225–230
14. Kohavi R (1996) Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of 2nd ACM SIGKDD International, conference on knowledge discovery and data mining, pp 202–207
15. Langley P, Iba W, Thompson K (1992) An analysis of Bayesian classifiers. In: 10th international conference artificial intelligence, AAAI Press, pp 223–228
16. Langley P, Saga S (1994) Induction of selective Bayesian classifiers. In: Proceedings of tenth conference, uncertainty in artificial intelligence, Morgan Kaufmann, pp 399–406
17. Lu J, Yang Y, Webb GI (2006) Incremental discretization for naive-Bayes classifier. Springer, Heidelberg, 4093, pp 223–238
18. Ozsen S, Gunecs S (2009) Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems. Expert Syst Appl 36:386–392

19. Pazzani MJ (1996) Constructive induction of cartesian product attributes, ISIS: In-formation, Stat Induction Sci 66–77
20. Pearl J (1988) Probabilistic reasoning in Intelligent systems: networks of plausible inference. Morgan Kaufmann
21. Sun W, Yuan YX (2006) Optimization theory and methods: nonlinear programming. Springer, Berlin
22. Taheri S, Mammadov M, Bagirov AM (2011) Improving Naive Bayes classifier using conditional probabilities, In the proceedings of ninth Australasian data mining conference (AusDM 2011), vol 125. Ballarat, Australia
23. Taheri S, Mammadov M (2011) Tree augmented Naive Bayes based on optimization. In: Proceedings of 42nd annual Iranian mathematics conference, Vali-e-Asr University of Rafsanjan, Iran
24. Wang S, Min F, Wang Z, Cao T (2009) OFFD: Optimal Flexible Frequency Discretization for Naive Bayes Classification. Springer, Heidelberg, pp 704–712
25. Webb GI, Boughton J, Wang Z (2005) Not so Naive Bayes: aggregating one dependence estimators. Mach Learn 58:5–24
26. Wu J, Cai Z (2011) Attribute weighting via differential evolution algorithm for attribute weighted Naive Bayes (WNB). J Comput Inf Syst 7(5):1672–1679
27. Xindong W et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14:1–37
28. Yatsko A, Bagirov AM, Stranieri A (2010) On the discretization of continuous features for classification, School of Information Technology and Mathematical Sciences, University of Ballarat Conference. (http://researchonline.ballarat.edu.au:8080/vital/access/manager/Repository)
29. Ying Y, Geoffrey I (2009) Discretization for naive-Bayes learning: managing discretization bias and variance. Mach Learn 74(1):39–74
30. Ying Y (2003) Discretization for naive-Bayes learning, PhD thesis, School of Computer Science and Software Engineering of Monash University
31. Zhang H, Sheng S (2005) Learning weighted Naive Bayes with accurate ranking. In: Proceedings of the 4th IEEE international conference on data mining 567–570
32. Zhou Y, Huang TS (2006) Weighted Bayesian network for visual tracking. In: Proceedings of the 18th international conference on pattern recognition (ICPR'O6), 0-7695-2521-0106