Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 4, 787–795
DOI: 10.2478/amcs-2013-0059

LEARNING THE NAIVE BAYES CLASSIFIER WITH OPTIMIZATION MODELS

SONA TAHERI [*], MUSA MAMMADOV [*,**]

[*] Centre for Informatics and Applied Optimization
School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia
e-mail: sonataheri@students.ballarat.edu.au, sona.taheri@unisa.edu.au

[**] Victoria Research Laboratory
National ICT Australia, Victoria 3010, Australia
e-mail: m.mammadov@ballarat.edu.au, musa.mammadov@nicta.com.au

Naive Bayes is among the simplest probabilistic classifiers. It often performs surprisingly well in many real world applications, despite the strong assumption that all features are conditionally independent given the class. In the learning process of this classifier with the known structure, class probabilities and conditional probabilities are calculated using training data, and then values of these probabilities are used to classify new observations. In this paper, we introduce three novel optimization models for the naive Bayes classifier where both class probabilities and conditional probabilities are considered as variables. The values of these variables are found by solving the corresponding optimization problems. Numerical experiments are conducted on several real world binary classification data sets, where continuous features are discretized by applying three different methods. The performances of these models are compared with the naive Bayes classifier, tree augmented naive Bayes, the SVM, C4.5 and the nearest neighbor classifier. The obtained results demonstrate that the proposed models can significantly improve the performance of the naive Bayes classifier, yet at the same time maintain its simple structure.

Keywords: Bayesian networks, naive Bayes classifier, optimization, discretization.

1. Introduction

Bayesian Networks (BNs) introduced by Pearl (1988) are high level representations of probability distributions over a set of variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ that are used for a learning process. The learning of BNs is divided in two steps: structure learning and parameter learning. The former is constructing a directed acyclic graph from the set **X**. In the graph, each node corresponds to the variable and each arc denotes a causal relationship between two variables, while the direction of the arc indicates the direction of the causality. When two nodes are joined by an arc, the causal node is called the parent of the other node, and another one is called the child. We use X_i to denote both the variable (feature) and its corresponding node, and $Pa(X_i)$ to denote the set of parents of the node X_i. Given a structure, finding probability distributions, class probabilities and conditional probabilities, associated with each variable is called parameter learning (Campos *et al.*, 2002; Polanska *et al.*, 2006; Zaidi *et al.*, 2012).

In particular, the joint probability distribution for **X** is given by

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i | Pa(X_i)). \tag{1}$$

However, accurate estimation of $P(X_i|Pa(X_i))$ requires finding the structure which is non-trivial. It has been proved that learning an optimal structure of a BN is an NP-hard problem (Chickering, 1996; Heckerman *et al.*, 2004). In order to avoid the intractable complexity of the structure learning in BNs, the naive Bayes classifier (Langley *et al.*, 1992; Taheri *et al.*, 2011) with the known structure has been used. In Naive Bayes (NB), features are conditionally independent given the class. This means that each feature has the class as an only parent. The efficiency of NB has witnessed its widespread development in real world applications including medical diagnosis, recommender systems, email filtering, web page perfecting and fraud detection (Crawford *et al.*, 2002; Kononenko, 2001; Miyahara and Pazzani, 2000;

Zupan *et al.*, 2001).

In this paper, our aim is to improve the performance of NB by applying optimization techniques, yet at the same time to maintain its simple structure. We consider class probabilities and conditional probabilities as unknown variables, whose optimal values can be computed by applying optimization techniques. We introduce three different optimization models for NB using different definitions of unknown variables.

Most data sets in real world applications involve continuous features. The most well-known attempt at improving the performance of NB with continuous features is the discretization of the features into intervals, instead of using the default option to utilize the normal distribution to calculate probabilities. The main reason is that NB with discretization tends to achieve a lower classification error than the original one (Dougherty *et al.*, 1995). It has been shown that the performance of NB classifier significantly improves when features are discretized using an entropy based method (Dougherty *et al.*, 1995). In this paper, therefore, we use the Fayyad and Irani method (Fayyad and Irani, 1993) based on a minimal entropy heuristic to discretize continuous features. We also apply two other different discretization methods. The first one, which is also the simplest one, transforms the continuous features to discrete ones using the median-based discretization method. The second one is the discretization algorithm recently introduced by Yatsko *et al.* (2011), using the Sub-Optimal Agglomerative Clustering (SOAC) algorithm.

The structure of the paper is as follows. In the next section, we provide a brief description of the NB classifier. In Section 3 with the preliminaries, we briefly describe the globally convergent optimization method, a Combination of the Gradient and Newton (CGN) methods, and discretization algorithms, respectively, which we required for our discussion in the latter part of the paper. In Section 4, we introduce three different optimization models for the NB classifier. The results of numerical experiments are given in Section 5. Section 6 concludes the paper.

2. Naive Bayes classifier

The naive Bayes classifier (Domingos and Pazzani, 1997; Langley *et al.*, 1992; Tóth *et al.*, 2005; Taheri and Mammadov, 2012) assumes that each feature only depends on the class as depicted in Fig. 1. This means that each feature has only the class as a parent. NB is attractive as it has an explicit and sound theoretical basis which guarantees optimal induction given a set of explicit assumptions. There is a drawback in which the independency assumptions of features with respect to the class are violated in some real world problems. However, it has been shown that NB is remarkably robust in the face of such violations (Domingos and Pazzani, 1996; Friedman *et al.*, 1997). NB is fast, easy to implement with the simple structure, and effective. It is also useful for high dimensional data as the probability of each feature is estimated independently. NB is one of the 10 top algorithms in data mining as listed by Wu *et al.* (2008).

Let C denote the class of an observation $\mathbf{X}$. To predict the class of the observation $\mathbf{X}$ by using the Bayes rule, the highest posterior probability of

$$P(C|\mathbf{X}) = \frac{P(C)P(\mathbf{X}|C)}{P(\mathbf{X})} \tag{2}$$

should be found.

In the NB classifier, using the assumption that features $X_1, X_2, \dots, X_n$ are conditionally independent of each other given the class, we get

$$P(C|\mathbf{X}) = \frac{P(C)\prod_{i=1}^{n} P(X_i|C)}{P(\mathbf{X})}. \tag{3}$$

In classification problems, Eqn. (3) is sufficient to predict the most probable class given a test observation.

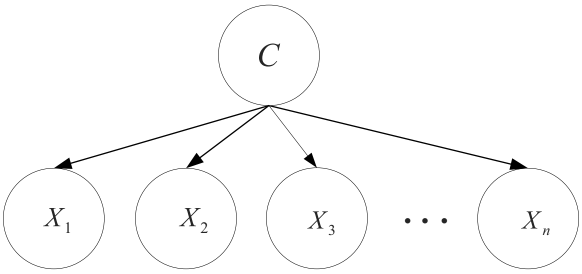

Fig. 1. Naive Bayes.

To estimate class probabilities $P(C)$ and conditional probabilities $P(X_i|C)$, $i = 1, \dots, n$, in the formula (3), in this paper we introduce three different optimization models.

3. Preliminaries

In this section, we briefly review the optimization method, CGN, and discretization algorithms, the Fayyad and Irani method and the SOAC algorithm, which we use in the latter part of the paper.

3.1. Combination of the gradient and Newton methods. In this section, we give a very brief introduction to the optimization algorithm, a combination of the gradient and the Newton methods (Taheri *et al.*, 2012). CGN is a new globally convergent optimization algorithm for solving unconstrained optimization problems. The idea is to combine two directions from different local optimization methods. The first direction is the gradient direction due to its global convergence property. The second one is Newton's direction to speed up the convergence rate. Two different combinations are considered in this algorithm. The first one is a novel combination in which the step length is determined only along the gradient direction. In the second one, the step length is considered along both directions. For more details, see the work of Taheri *et al.* (2012).

3.2. Discretization methods. In order to apply the NB classifier to data sets with continuous features, one should first discretize those features. Discretization is a process which transforms continuous numeric values into discrete ones. In this paper, we apply three different methods to discretize continuous features. The first one, which is also the simplest, transforms the continues features to discrete ones, $\{0{,}1\}$ using the median-based discretization method. We also apply two other methods, which allows us to get more than two values for discretized features. One is the Fayyad and Irani discretization method (Fayyad and Irani, 1993), which is the most often applied discretization method in the literature, and another one is the discretization algorithm using sub-optimal agglomerative clustering, which was recently introduced by Yatsko *et al.* (2011).

In the next section, we introduce three optimization models to improve the performance of the NB classifier. In the proposed models, class probabilities and conditional probabilities are considered to be unknown variables, and the optimal values of these variables are computed by applying optimization techniques.

4. Optimization models

Let $D = \{(\mathbf{X}_1, C_1), (\mathbf{X}_2, C_2), \dots, (\mathbf{X}_N, C_N)\}$ be a data set, where N is the number of observations; $\mathbf{X}_i = \{X_{i1}, X_{i2}, \dots, X_{in}\}$, n is the number of features. We assume binary classification, that is, $C_i \in \{-1, 1\}$, $i = 1, \dots, N$, and we use the notation $\overline{C} = -C$.

From the Bayes rule, we know that

$$P(C_i|\mathbf{X}_i) = \frac{P(\mathbf{X}_i|C_i)P(C_i)}{P(\mathbf{X}_i)}, \tag{4}$$

where $P(\mathbf{X}_i) = P(\mathbf{X}_i|C_i)P(C_i) + P(\mathbf{X}_i|\overline{C_i})P(\overline{C_i})$. Since C_i and $\overline{C_i}$ are complimentary to each other, and $P(C_i), P(\overline{C_i})$ are probabilities, we have

$$P(C_i) + P(\overline{C_i}) = 1, \quad 0 \le P(C_i), \quad P(\overline{C_i}) \le 1. \tag{5}$$

In the NB classifier, it is assumed that all features are independent of each other given the class. This means that

$$P(\mathbf{X}_i|C_i) = \prod_{j=1}^{n} P(X_{ij}|C_i). \tag{6}$$

Therefore the formula (4) for NB can be rewritten as

$$P(C_i|\mathbf{X}_i) = \frac{\prod_{j=1}^{n} P(X_{ij}|C_i)P(C_i)}{\prod_{j=1}^{n} P(X_{ij}|C_i)P(C_i) + \prod_{j=1}^{n} P(X_{ij}|\overline{C_i})P(\overline{C_i})}. \tag{7}$$

Similarly,

$$P(\overline{C}_i|\mathbf{X}_i) = \frac{\prod_{j=1}^{n} P(X_{ij}|\overline{C}_i)P(\overline{C}_i)}{\prod_{j=1}^{n} P(X_{ij}|C_i)P(C_i) + \prod_{j=1}^{n} P(X_{ij}|\overline{C_i})P(\overline{C_i})}. \tag{8}$$

Using the definition of the conditional probability,

$$P(X_{ij}|C_i) = \frac{P(X_{ij}, C_i)}{P(C_i)}, \tag{9}$$

(7) and (8) can be represented as

$$P(C_i|\mathbf{X}_i) = \frac{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(P(C_i)\big)^{n-1}}}{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(P(C_i)\big)^{n-1}} + \dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(P(\overline{C}_i)\big)^{n-1}}} \tag{10}$$

and

$$P(\overline{C}_i|\mathbf{X}_i) = \frac{\dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(P(\overline{C}_i)\big)^{n-1}}}{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(P(C_i)\big)^{n-1}} + \dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(P(\overline{C}_i)\big)^{n-1}}}. \tag{11}$$

Considering that C_i is the class of the observation $\mathbf{X}_i$, the value of $P(C_i|\mathbf{X}_i)$ is expected to be greater than that of $P(\overline{C_i}|\mathbf{X}_i)$ for the majority of observations, $i = 1, \dots, N$.

In the next three subsections, we will present three different optimization models for the NB classifier by considering class probabilities and conditional probabilities as unknown variables.

4.1. Model 1: An optimization model based on class probabilities. In this subsection, we consider class probabilities as variables. We introduce the variable w for the probability $P(1)$, and since $P(1) + P(-1) = 1$, we have $1 - w$ for the probability $P(-1)$. Let us consider

$$\xi(w; C) = \begin{cases} w & \text{if } C = 1, \\ 1 - w & \text{if } C = -1. \end{cases} \tag{12}$$

Then the objective functions for (10) and (11) can be written as

$$f_1(w) = \sum_{i=1}^{N} \frac{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(\xi(w; C_i)\big)^{n-1}}}{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(\xi(w; C_i)\big)^{n-1}} + \dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(\xi(w; \overline{C}_i)\big)^{n-1}}} \tag{13}$$

and

$$f_2(w) = \sum_{i=1}^{N} \frac{\dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(\xi(w; \overline{C}_i)\big)^{n-1}}}{\dfrac{\prod_{j=1}^{n} P(X_{ij}, C_i)}{\big(\xi(w; C_i)\big)^{n-1}} + \dfrac{\prod_{j=1}^{n} P(X_{ij}, \overline{C}_i)}{\big(\xi(w; \overline{C}_i)\big)^{n-1}}}. \tag{14}$$

By considering C_i as the class of $\mathbf{X}_i$, it is quite natural that the value of $f_1(w)$ should be maximized while the value of $f_2(w)$ minimized. Therefore, the NB classifier leads to an optimization problem:

$$\text{maximize: } \psi(w) = \frac{f_1(w)}{f_2(w)} \tag{15}$$

subject to $0 \le w \le 1$.

4.2. Model 2: A simplified version of Model 1.

In this subsection, for simplicity we consider only the variable w for the probability $P(1)$, and $1 - w$ for the probability $P(-1)$ in the formulas (7) and (8). Therefore, the second optimization model for the NB classifier under these assumptions and using (12) can be described by the objective functions (16) and (17). Then we can consider an optimization problem in a similar way to (15):

$$\text{maximize: } \phi(w) = \frac{\widetilde{f}_1(w)}{\widetilde{f}_2(w)} \tag{18}$$

subject to $0 \le w \le 1$.

Since problems in Models 1 and 2 are univariate optimization ones, we partition the constraint $0 \le w \le 1$ into 1000 intervals and we find the maximum value of the objective function in each model.

4.3. Model 3: An optimization model based on class probabilities and conditional probabilities.

In the third optimization model for the NB classifier, we discretize the values of all features to binary values, $\{0, 1\}$, by applying the median-based discretization method. Since we have binary classification $(1$, or $-1)$, we consider not only $P(1)$ and $P(-1)$, but also the conditional probabilities $P(1|1)$, $P(0|1)$, $P(1|-1)$ and $P(0|-1)$ as variables. For each feature j, $j = 1, \dots, n$, we introduce four variables:

v_{1j} for $P(j\text{-th feature is } 1|\text{class is}]\ 1)$,

v_{2j} for $P(j\text{-th feature is } 0|\text{class is } 1)$,

v_{3j} for $P(j\text{-th feature is } 1|\text{class is } -1)$,

v_{4j} for $P(j\text{-th feature is } 0|\text{class is } -1)$.

As a result, we have a matrix of $4n$ variables,

$$V = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ v_{31} & v_{32} & \dots & v_{3n} \\ v_{41} & v_{42} & \dots & v_{4n} \end{pmatrix}. \tag{19}$$

Since we have constraints $v_{1j} + v_{2j} = 1$ and $v_{3j} + v_{4j} = 1,\ j = 1, \dots, n$, the matrix V can be rewritten as

$$W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \end{pmatrix}, \tag{20}$$

where $w_{1j} = v_{1j}$ and $w_{2j} = v_{3j},\ j = 1, \dots, n$. Clearly, $v_{2j} = 1 - w_{1j}$ and $v_{4j} = 1 - w_{2j},\ j = 1, \dots, n$.

Similarly as in (12), we introduce

$$\zeta(\alpha, \beta; X, C) = \begin{cases} \alpha & \text{if } X = 1, C = 1, \\[2ex] 1 - \alpha & \text{if } X = 0, C = 1, \\[2ex] \beta & \text{if } X = 1, C = -1, \\[2ex] 1 - \beta & \text{if } X = 0, C = -1. \end{cases} \tag{21}$$

Then, using (12) and (21), the formulas (16) and (17) are given by (22) and (23) Therefore, the maximization problem for this model is

$$\text{maximize: } \varphi(w, W) = \frac{\widehat{f}_1(w, W)}{\widehat{f}_2(w, W)} \tag{24}$$

subject to $0 \le w, w_{1j}, w_{2j} \le 1,\ \ j = 1, \dots, n$.

The problem (24) is a constrained optimization one. We apply the penalty method with a parameter $\mu = 10^4$ to reduce this problem to an unconstrained one. The unconstrained problem is as follows:

$$\begin{aligned} &\text{maximize} \\ &\varpi(w, W, \mu) = \varphi(w, W) - \mu\big\{[\max(0, -w, w-1)]^2 \\ &\qquad + \sum_{j=1}^{n} \big([\max(0, -w_{1j}, w_{1j} - 1)]^2 \\ &\qquad + [\max(0, -w_{2j}, w_{2j} - 1)]^2\big)\big\}. \end{aligned} \tag{25}$$

$$\widetilde{f}_1(w) = \sum_{i=1}^{N} \frac{\prod_{j=1}^{n} P(X_{ij}|C_i)\xi(w;C_i)}{\prod_{j=1}^{n} P(X_{ij}|C_i)\xi(w;C_i) + \prod_{j=1}^{n} P(X_{ij}|\overline{C}_i)\xi(w;\overline{C}_i)}, \tag{16}$$

$$\widetilde{f}_2(w) = \sum_{i=1}^{N} \frac{\prod_{j=1}^{n} P(X_{ij}|\overline{C}_i)\xi(w;\overline{C}_i)}{\prod_{j=1}^{n} P(X_{ij}|C_i)\xi(w;C_i) + \prod_{j=1}^{n} P(X_{ij}|\overline{C}_i)\xi(w;\overline{C}_i)}. \tag{17}$$

$$\widehat{f}_1(w,W) = \sum_{i=1}^{N} \frac{\prod_{j=1}^{n} \zeta(w_{1j},w_{2j};X_{ij},C_i)\xi(w;C_i)}{\prod_{j=1}^{n} \big(\zeta(w_{1j},w_{2j};X_{ij},C_i)\xi(w;C_i) + \zeta(w_{1j},w_{2j};X_{ij},\overline{C}_i)\xi(w;\overline{C}_i)\big)}, \tag{22}$$

$$\widehat{f}_2(w,W) = \sum_{i=1}^{N} \frac{\prod_{j=1}^{n} \zeta(w_{1j},w_{2j};X_{ij},\overline{C}_i)\xi(w;\overline{C}_i)}{\prod_{j=1}^{n} \big(\zeta(w_{1j},w_{2j};X_{ij},C_i)\xi(w;C_i) + \zeta(w_{1j},w_{2j};X_{ij},\overline{C}_i)\xi(w;\overline{C}_i)\big)}, \tag{23}$$

For solving the nonlinear and nonconvex unconstrained optimization problem (25), we use the new globally convergent optimization method, CGN (Taheri *et al.*, 2012), which was briefly introduced in Section 3. The reason for choosing this method is that it has better performance than some well-known local optimization methods such as the gradient method and the Newton method. We demonstrate this fact in the numerical section.

Model 3 can be generalized to any discrete features. We have not considered such a model in this paper, which could be a topic for a separate research paper.

5. Numerical experiments

To verify the effectiveness of the proposed models, numerical experiments with a number of real world data sets have been carried out on different methods of data mining.

5.1. Data collections. This paper studies 14 binary data sets taken from the literature. A brief description of the data sets is given in Table 1. Their detailed description can be found in the UCI machine learning repository (Asuncion and Newman, 2007), and the tools page of the LIBSVM (Chang and Lin, 2001). These data sets have been analyzed quite frequently by the current data mining approaches. Another reason for selecting these data sets was the fact that conventional approaches have analyzed them with variable success.

5.2. Results and discussion. We conduct empirical comparison with naive Bayes, Tree Augmented Naive (TAN) Bayes, the Support Vector Machine (SVM), a specific algorithm of the decision-tree (C4.5) and the nearest neighbor classifier (1-NN). The reason for choosing these methods for comparison with the proposed models is that they are on the list of top 10 algorithms in data mining (Wu *et al.*, 2008). For each method, we run 50 trials and then the average accuracy over the 50 runs is 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.

We discretize the values of features in data sets using three different methods. In the first one which is the simplest method, we apply the median-based discretization method where the values of features have been transferred to $\{0,1\}$. In the second one, we apply the well-known Fayyad and Irani discretization method (Fayyad and Irani, 1993). The third one is the recently introduced discretization method, the SOAC algorithm (Yatsko *et al.*, 2011). Model 3 is not suitable when applying the Fayyad and Irani and SOAC discretization methods as it needs binary values for features.

The efficiency of the CGN method when applied to Model 3 has been tested on some data sets randomly chosen from Table 1, such as credit approval, diabetes and heart disease.

Table 3. Average test set accuracy over 10 fold cross validation for 14 data sets using the median-based discretization method.

Data	NB	TAN	SVM	C4.5	1-NN	M1	M2	M3
Breas	96.37	95.92	95.32	91.21	96.71	97.40	97.62	**97.99**
Cong	91.43	91.74	96.76	95.71	95.78	96.83	96.89	**96.97**
Credit	84.92	82.90	85.51	87.49	83.21	84.89	88.86	**89.98**
Diabet	75.93	76.51	76.72	76.12	74.94	76.92	77.35	**78.65**
Germa	75.32	74.02	76.15	72.14	72.23	75.65	75.91	**78.36**
Haber	75.22	73.98	73.54	71.79	70.14	75.93	77.98	**78.65**
Heart	81.67	84.72	80.54	81.73	80.03	86.86	87.67	**88.87**
Hepat	83.76	83.44	83.43	82.99	83.58	83.94	84.42	**84.43**
Ionos	82.95	84.57	85.98	86.35	**88.97**	84.61	85.93	88.68
Liver	61.84	61.87	60.05	60.17	62.85	63.01	65.64	**65.72**
Sonar	75.16	75.48	76.99	76.69	74.15	76.41	76.65	**79.97**
Spam	90.31	89.89	90.47	91.96	92.41	90.24	92.56	**93.57**
Svm1	92.72	92.16	93.31	93.72	95.89	93.75	94.83	**96.89**
Svm3	81.34	83.01	80.11	81.23	80.58	83.15	83.72	**86.65**
Ave	82.06	82.15	82.49	82.09	82.24	83.56	84.71	**86.09**

Table 4. Average test set accuracy over 10 fold cross validation for 14 data sets using the FaI discretization method.

Data	NB	TAN	SVM	C4.5	1-NN	M1	M2
Breas	97.18	96.52	96.52	94.11	96.12	97.72	**97.78**
Cong	90.11	93.21	95.04	95.32	95.72	95.81	**96.12**
Credit	86.10	84.78	85.03	84.87	83.04	86.93	**87.71**
Diabet	74.56	75.14	75.51	73.83	73.95	**76.42**	75.84
Germa	74.50	73.13	76.41	71.92	72.03	75.95	**76.81**
Haber	75.09	74.41	73.20	71.24	69.93	**77.14**	76.87
Heart	82.93	81.23	81.67	82.85	78.14	83.56	**85.62**
Hepat	84.56	83.91	85.16	83.87	84.51	85.76	**85.81**
Ionos	88.62	89.77	89.67	89.98	**89.99**	88.96	89.57
Liver	63.26	63.18	62.03	62.15	62.89	64.83	**65.72**
Sonar	76.32	76.47	**77.96**	77.31	72.11	76.40	76.37
Spam	90.41	89.78	90.43	92.97	92.45	**92.84**	92.51
Svm1	92.39	91.61	94.31	95.99	**96.07**	94.88	93.98
Svm3	81.23	82.47	80.37	81.38	80.14	84.90	**86.12**
Ave	82.66	82.54	83.09	82.69	81.93	84.43	**84.77**

Table 5. Average test set accuracy over 10 fold cross validation for 14 data sets using the SOAC discretization algorithm.

Data	NB	TAN	SVM	C4.5	1-NN	M1	M2
Breas	96.12	95.60	95.31	91.16	96.16	97.85	**97.99**
Cong	90.11	91.42	96.75	95.12	95.81	96.81	**96.97**
Credit	85.85	84.98	86.11	87.54	84.46	86.94	**88.51**
Diabet	75.78	75.90	76.68	75.63	74.97	76.17	**78.12**
German	74.61	74.01	**76.35**	72.21	72.22	76.08	76.19
Haber	74.66	**76.08**	73.36	72.15	71.85	75.61	75.32
Heart	78.62	77.37	77.96	79.17	78.67	79.44	**84.11**
Hepat	82.93	81.54	84.24	82.34	84.52	85.83	**86.37**
Ionos	85.92	86.18	86.15	86.71	**88.75**	86.98	88.23
Liver	65.82	65.73	63.69	64.98	63.79	66.51	**66.94**
Sonar	75.09	75.76	77.74	76.41	74.53	76.13	**76.81**
Spam	89.30	89.04	91.56	**93.73**	92.47	92.89	93.43
Svm1	95.81	94.91	95.94	96.91	96.97	97.38	**97.75**
Svm3	77.25	79.99	78.32	78.49	81.05	82.11	**82.27**
Ave	81.99	82.03	82.86	82.32	82.58	84.05	**84.92**

The results are shown in Table 2. This table proves that, in comparison with other commonly used local methods such as the Gradient Method (GM) and the Newton Method (NM), the choice of the optimization method, CGN, is worthwhile and is a more robust and efficient local optimization method to be applied in Model 3. We scaled the values of the objective function obtained by the CGN method to 1 and then the results achieved by other methods were shown in relation to the first method. For example, in this table, a cell containing 1/.7/f would denote that the first method (CGN) finds the optimal solution, the second one (GM) shows that 70 percent of the optimal value was obtained, and the third method (NM) failed to find the solution.

The results of the test set accuracy for different methods using different discretization methods are shown in Tables 3–5.

Table 3 demonstrates the test set accuracy obtained by NB, TAN, the SVM, C4.5, 1-NN, Models 1–3 on 14 data sets using the median-based discretization method. The results table demonstrate that the test set accuracy of Models 2 and 3 in all data sets are better than those obtained by NB. Model 1 has also higher accuracy than NB in the majority of data sets. In 12 cases out of 14, Model 1 shows better accuracy than NB. In data sets Credit Approval and Spam Base, the accuracy of this model almost ties with that obtained by NB. Table 3 also shows that the proposed models have much better accuracy than TAN in all data sets. Observe in this table/that Model 2 has better accuracy than the SVM in 11 cases out of 14, and Model 1 performs better than the SVM in 9 cases out of 14. Model 2 has greater accuracy than C4.5 in 12 data sets, and the accuracy obtained by Model 1 is higher than C4.5 in 10 cases out of 14. The results in this table also show that Model 3 outperforms the SVM and C4.5 in all data sets. Models 1–3 have greater accuracy than 1-NN in 11 data sets, 12 and 13 cases out of 14, respectively.

Table 1. Brief description of data sets.

Data sets	# Observations	# Features
Breast Cancer	699	10
Congres Vote	435	16
Credit Approval	690	15
Diabetes	768	8
German.numer	1000	24
Haberman	306	3
Heart Disease	303	14
Hepatitis	155	19
Ionosphere	351	34
Liver Disorders	345	6
Sonar	208	60
Spambase	4601	57
Svmguide1	7089	4
Svmguide3	1284	21

Table 2. Comparison results to show the efficiency of the chosen optimization method, CGN, for Model 3 of NB.

Initial point	Credit Approval	Diabetes	Heart Disease
$(0.1, 0.9)$	1/.8/1	1/f/1	1/.9/1
$(0.2, 0.8)$	1/.9/1	1/.6/f	1/.9/.9
$(0.3, 0.7)$	1/.9/f	1/.8/f	1/.9/1
$(0.4, 0.6)$	1/1/1	1/.8/1	1/.9/f
$(0.5, 0.5)$	1/1/1	1/.8/1	1/.9/1
$(0.6, 0.4)$	1/1/f	1/.8/f	1/.9/f
$(0.7, 0.3)$	1/.9/f	1/.9/1	1/.9/f
$(0.8, 0.2)$	1/.9/1	1/.6/f	1/.9/1
$(0.9, 0.1)$	1/.8/1	1/1/1	1/.9/1

Table 4 presents the test set accuracy obtained by NB, TAN, the SVM, C4.5, 1-NN Models 1 and 2 on 14 data sets, where continuous features are discretized by applying the Fayyad and Irani (FaI) method. The results presented in this table demonstrate that the accuracy of Models 1 and 2 is significantly better than that of NB in all data sets. In 12 data sets out of 14, these models perform better than TAN, whereas the latter has slightly higher accuracy in the Ionosphere and Sonar data sets. The results from Table 4 also indicate that Models 1 and 2 have the greater accuracy than the SVM in 11 data sets. Compared to the C4.5, Table 4 shows higher accuracy for Models 1 and 2 in 10 data sets. These two models also have greater accuracy than 1-NN in 12 data sets.

The test set accuracy obtained by NB, TAN, the SVM, C4.5, 1-NN, Models 1 and 2 on 14 data sets using the discretization algorithm SOAC is summarized in Table 5. The results from this table show that the accuracy obtained by Models 1 and 2 in all data sets are higher than those obtained by NB. The accuracy of Models 1 and 2 is better than those of TAN on most of data sets. In 13 cases out of 14, Models 1 and 2 have greater accuracy than TAN. The results from Table 5 also demonstrate that Model 2 has greater accuracy than the SVM in 12 data sets out of 14, and the latter method outperforms the Model 1 in the diabetes, german.numer and sonar data sets. Table 5 indicates higher accuracy for the Model 2 in 13 data sets, and 11 cases for Model 1 when compared to C4.5. Models 1 and 2 also perform better than 1-NN in 13 data sets.

The numerical results generally demonstrate that the proposed models can significantly improve the performance of the naive Bayes classifier, yet at the same time maintain its simple structure. The average accuracy of 14 data sets obtained by each classifier shows that Model 3 has a dramatic increase in the test set accuracy, and it is reasonable due to considering more variables replacing class probabilities and conditional probabilities which allows to build more accurate model. However, these models require more training time than the naive

Bayes classifier due to applying optimization techniques.

6. Conclusion

In this paper, we introduced three different optimization models for the naive Bayes classifier by considering class probabilities and conditional probabilities as unknown variables. Then we applied optimization techniques to find the optimal values for these variables. We compared the proposed models with NB, TAN, the SVM, C4.5 and 1-NN on 14 real world binary classification data sets. The values of features in data sets are discretized by using a median-based discretization method and applying two different discretization algorithms, the Fayyad and Irani method and the algorithm SOAC. We presented results of numerical experiments. The results demonstrate that the proposed models perform better than NB, TAN, the SVM, C4.5, and 1-NN in terms of accuracy, yet at the same time they maintain the simple structure of NB. Especially Model 3 increased the test set accuracy of each data sets and this is reasonable due to considering more variables replacing class probabilities and conditional probabilities which allows us to build a more accurate model.

In this work, we mainly focus on binary classification data sets since they are the simplest among the main classification categories. However, the applications of the proposed models for other types of data sets, and also generalizing Model 3 to any discrete features, remain important questions for future work.

References

Asuncion, A. and Newman, D. (2007). UCI machine learning repository, `http://www.ics.uci.edu/mlearn/mlrepository`.

Campos, M., Fernandez-Luna, Gamez, A. and Puerta, M. (2002). Ant colony optimization for learning Bayesian networks, *International Journal of Approximate Reasoning* **31**(3): 291–311.

Chang, C. and Lin, C. (2001). LIBSVM: A library for support vector machines, `http://www.csie.ntu.edu.tw/cjlin/libsvm`.

Chickering, D.M. (1996). Learning Bayesian networks is NP-complete, *in* D. Fisher and H. Lenz (Eds.), *Artificial Intelligence and Statistics*, Springer-Verlag, Berlin/Heidelberg, pp. 121–130.

Crawford, E., Kay, J. and Eric, M. (2002). The intelligent email sorter, *Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia*, pp. 83–90.

Domingos, P. and Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier, *Proceedings of the 13th International Conference on Machine Learning, Bari, Italy*, pp. 105–112.

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* (29): 103–130.

Dougherty, J., Kohavi, R. and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features, *Proceedings of the 12th International Conference on Machine Learning, San Francisco, CA, USA*, pp. 194–202.

Fayyad, U.M. and Irani, K. (1993). On the handling of continuous-valued attributes in decision tree generation, *Machine Learning* **8**: 87–102.

Friedman, N., Geiger, D. and Goldszmidti, M. (1997). Bayesian network classifiers, *Machine Learning* **29**(2): 131–163.

Heckerman, D., Chickering, D. and Meek, C. (2004). Large sample learning of Bayesian networks is NP-hard, *Journal of Machine Learning Research* **5**: 1287–1330.

Kononenko, I. (2001). Machine learning for medical diagnosis: History, state of the art and perspective, *Artificial Intelligence in Medicine* **23**: 89–109.

Langley, P., Iba, W. and Thompson, K. (1992). An analysis of Bayesian classifiers, *10th International Conference on Artificial Intelligence, San Jose, CA, USA*, pp. 223–228.

Miyahara, K. and Pazzani, M.J. (2000). Collaborative filtering with the simple Bayesian classifier, *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, Melbourne, Australia*, pp. 679–689.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Fransisco, CA.

Polanska, J., Borys, D. and Polanski, A. (2006). Node assignment problem in Bayesian networks, *International Journal of Applied Mathematics and Computer Science* **16**(2): 233–240.

Taheri, S. and Mammadov, M. (2012). Structure learning of Bayesian networks using a new unrestricted dependency algorithm, *IMMM 2012: The 2nd International Conference on Advances in Information on Mining and Management, Venice, Italy*, pp. 54–59.

Taheri, S., Mammadov, M. and Bagirov, A. (2011). Improving naive Bayes classifier using conditional probabilities, *9th Australasian Data Mining Conference, Ballarat, Australia*, pp. 63–68.

Taheri, S., Mammadov, M. and Seifollahi, S. (2012). Globally convergent algorithms for solving unconstrained optimization problems, *Optimization*: 1–15.

Tóth, L., Kocsor, A. and Csirik, J. (2005). On naive Bayes in speech recognition, *International Journal of Applied Mathematics and Computer Science* **15**(2): 287–294.

Wu, X., Vipin Kumar, J., Quinlan, R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, J., Ng, A., Liu, B., Yu, P. S., Zhou, Z., Steinbach, M., Hand, D. J. and Steinberg, D. (2008). Top 10 algorithms in data mining, *Knowledge and Information Systems* **14**: 1–37.

Yatsko, A., Bagirov, A.M. and Stranieri, A. (2011). On the discretization of continuous features for classification, *Proceedings of the 9th Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia*, Vol. 125.

Zaidi, A., Ould Bouamama, B. and Tagina, M. (2012). Bayesian reliability models of Weibull systems: State of the art, *International Journal of Applied Mathematics and Computer Science* **22**(3): 585–600, DOI: 10.2478/v10006-012-0045-2.

Zupan, B., Demsar, J., Kattan, M.W., Ohori, M., Graefen, M., Bohanec, M. and Beck, J.R. (2001). Orange and decisions-at-hand: Bridging predictive data mining and decision support, *Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning, Freiburg, Germany*, pp. 151–162.

Sona Taheri completed her Ph.D. in 2012 in Australia. The main areas of her interest are Bayesian networks, optimization and their applications. She has published 9 research papers during her Ph.D.research.

Musa Mammadov is a senior research fellow at the School of Information Technology and Mathematical Sciences of the University of Ballarat, Australia. He is a member of the Control and Signal Processing Group at National ICT Australia (NICTA). The main areas of his interest include optimal control theory, global optimization and their applications. He has published more than 100 research papers.

Received: 27 August 2012
Revised: 12 March 2013
Re-revised: 15 July 2013