Interval Estimation Naïve Bayes

V. Robles[1], P. Larrañaga[2], J.M. Peña[1], E. Menasalvas[1], and M.S. Pérez[1]

[1] Department of Computer Architecture and Technology,
Technical University of Madrid, Madrid, Spain,
{vrobles, jmpena, emenasalvas, mperez}@fi.upm.es

[2] Department of Computer Science and Artificial Intelligence,
University of the Basque Country, San Sebastián, Spain,
ccplamup@si.ehu.es

Abstract. Recent work in supervised learning has shown that a surprisingly simple Bayesian classifier called naïve Bayes is competitive with state of the art classifiers. This simple approach stands from assumptions of conditional independence among features given the class. In this paper a new naïve Bayes classifier called *Interval Estimation naïve Bayes* is proposed. Interval Estimation naïve Bayes is performed in two phases. First, an interval estimation of each probability necessary to specify the naïve Bayes is calculated. On the second phase the best combination of values inside these intervals is calculated using a heuristic search that is guided by the accuracy of the classifiers. The founded values in the search are the new parameters for the naïve Bayes classifier. Our new approach has shown to be quite competitive related to simple naïve Bayes. Experimental tests have been done with 21 data sets from the UCI repository.

1 Introduction

The naïve Bayes classifier [3,9] is a probabilistic method for classification. It can be used to determine the probability that an example belongs to a class given the values of the predictor variables. The naïve Bayes classifier guarantees optimal induction given a set of explicit assumptions [1]. However, it is known that some of these assumptions are not compliant in many induction scenarios, for instance, the condition of variable independence respecting to the class variable. Improvements of accuracy have been demonstrated by a number of approaches, collectively named semi naïve Bayes classifiers, which try to adjust the naïve Bayes to deal with a-priori unattended assumptions.

Previous semi naïve Bayes classifiers can be divided into three groups, depending on different pre/post-processing issues: (i) to manipulate the variables to be employed prior to application of naïve Bayes induction [13,15,19], (ii) to select subsets of the training examples prior to the application of naïve Bayes classification [11,14] and (iii) to correct the probabilities produced by the standard naïve Bayes [5,22].

We propose a new semi naïve Bayes approach named *Interval Estimation naïve Bayes (IENB)*, that tries to relieve the assumption of independence of the

M.R. Berthold et al. (Eds.): IDA 2003, LNCS 2810, pp. 143–154, 2003.

variables given the class by searching for the best combination of naïve Bayes probabilities inside theirs confidence intervals. There is some related work with this approach. In [7] it is described an algorithm named *Iterative Bayes*, that tries to improve the conditional probabilities of naïve Bayes in an iterative way with a hill-climbing strategy. In [20] is presented an algorithm named *Robust Bayes Classifier (RBC)*, a naïve Bayes classifier designed for the case in which learning instances are incomplete. RBC takes into account all the possibilities for the missing values, calculating an interval for each naïve Bayes probability. In [23] a naïve Bayes extension named *naïve Credal Classifier (NCC)* is presented. Credal sets represent probability distributions as points that belong to a close and fenced geometrical regions. In [24] the authors propose to find the parameters that maximize the conditional likehood in a Bayesian network in spite of the sample joint likehood.

Interval Estimation naïve Bayes has been implemented in Visual C++ and the experimental evaluation has been done with 21 problems from the UCI database [18].

The outline of this paper is as follows: Section 2 presents the naïve Bayes classifier. Section 3 is a brief introduction to statistical inference. Section 4 presents the new algorithm Interval Estimation naïve Bayes. Section 5 illustrates the results with the UCI experiments. Section 6 gives the conclusions and suggests further future work.

2 Naïve Bayes

The naïve Bayes classifier [3,9] is a probabilistic method for classification. It performs an approximate calculation of the probability that an example belongs to a class given the values of predictor variables. The simple naïve Bayes classifier is one of the most successful algorithms on many classification domains. In spite of its simplicity, it is shown to be competitive with other more complex approaches in several specific domains.

This classifier learns from training data the conditional probability of each variable X_k given the class label c. Classification is then done by applying Bayes rule to compute the probability of C given the particular instance of $X_1, \ldots, X_n$,

$$P(C = c|X_1 = x_1, \ldots, X_n = x_n)$$

Naïve Bayes is founded on the assumption that variables are conditionally independent given the class. Therefore the posterior probability of the class variable is formulated as follows,

$$P(C = c|X_1 = x_1, \ldots, X_n = x_n) \propto P(C = c) \prod_{k=1}^{n} P(X_k = x_k|C = c) \quad (1)$$

This equation is highly appropriate for learning from data, since the probabilities $p_i = P(C = c_i)$ and $p^i_{k,r} = P(X_k = x^r_k|C = c_i)$ may be estimated from

training data. The result of the classification is the class with highest posterior probability.

In naïve Bayes these parameters are estimated using a point estimation (see section 3.1). The first step in the new algorithm we propose is based on an interval estimation for the parameters (see section 3.2).

3 Parameter Estimation

Statistical inference studies a collection of data based on a sample of these ones. This sample represents the part of population considered in the analysis. Amongst other things, statistical inference studies the problem known as "estimation problem".

There are two ways of accomplishing this task:

- *Point Estimation:* Point estimation uses a sample with the aim of assigning a single value to a parameter. The maximum likelihood method is used in this context.
- *Interval Estimation:* This technique calculates for each sample an interval that probably contains the parameter. This interval is called confidence interval. The probability of a parameter to be included in an interval is known as confidence level.

3.1 Point Estimation of Parameters in Naïve Bayes

Considering the instances of the database $\mathcal{D} = \{(\mathbf{x}^{(1)}, c^{(1)}), \ldots, (\mathbf{x}^{(N)}, c^{(N)})\}$ as a random sample on size N where the predictor variables $X_1, \ldots, X_n$ follow Bernoulli distributions and using the maximum likelihood method, the next intuitive results are reached:

$$\hat{p}_i = \frac{N_i}{N} \tag{2}$$

where,
N is the size of the database
N_i is the number of instances where $C = c_i$

$$\hat{p}^i_{k,r} = \frac{N_{kri}}{N_i} \tag{3}$$

where,
N_{kri} denotes the number of instances where $X_k = x^r_k$ and $C = c_i$
N_i denotes the number of instances where $C = c_i$

3.2 Interval Estimation of Parameters in IENB

In the case of IENB the calculation of confidence intervals of the parameters is required. This estimation is achieved by the calculation of the sum of the variables in the sample, which generate a binomial distribution that can be approximated by a normal distribution, given the next result, obtaining that the interval that contains the parameter value, p, with a confidence level of $1-\alpha$ is given by:

$$\left(\hat{p} - z_\alpha \sqrt{\frac{\hat{p}(1-\hat{p})}{M}}; \hat{p} + z_\alpha \sqrt{\frac{\hat{p}(1-\hat{p})}{M}}\right) \tag{4}$$

where,
$\hat{p}$ is the point estimation of the probability
z_α is the $(1-\frac{\alpha}{2})$ percentil in the $\mathcal{N}(0,1)$ distribution
M is the corresponding sample size (N_i or N)

4 Interval Estimation Naïve Bayes – IENB

We propose a new semi naïve Bayes approach named *Interval Estimation naïve Bayes (IENB)*. In this approach, instead of calculating the point estimation of the conditional probabilities from data, as simple naïve Bayes makes, confidence intervals are calculated. According to this, by searching for the best combination of values into these intervals, we aim to relieve the assumption of independence among variables the simple naïve Bayes makes –because the effect of the combination of probabilities is evaluated with the overall classifier accuracy–. This search is carry out by a heuristic search algorithm and is guided by the accuracy of the classifiers.

As it is represented in figure 3 at the end of the paper, there are three main important aspects in IENB algorithm:

– **Calculation of confidence intervals**
Given the dataset, the first step is to calculate the confidence intervals for each conditional probability and for each class probability. For the calculation of the intervals first the point estimations of these parameters (see section 3.1) must be computed.
In this way, each conditional probability $p^i_{k,r} = P(X_k = x^r_k | C = c_i)$, that has to be estimated from the dataset must be computed with the next confidence interval, as it is introduced in section 3.1. The contribution of this paper is addressed towards this approach.

For $k = 1, \ldots, n; i = 1, \ldots, r_0; r = 1, \ldots, r_k$

$$\left(\hat{p}^i_{k,r} - z_\alpha \sqrt{\frac{\hat{p}^i_{k,r}(1-\hat{p}^i_{k,r})}{N_i}}; \hat{p}^i_{k,r} + z_\alpha \sqrt{\frac{\hat{p}^i_{k,r}(1-\hat{p}^i_{k,r})}{N_i}}\right) \tag{5}$$

denotes the interval estimation for the conditional probabilities $p^i_{k,r}$, where,
r_k is the possible values of variable X_k
r_0 is the possible values of the class
$\hat{p}^i_{k,r}$ is the point estimation of the conditional probability $P(X_k = x^r_k | C = c_i)$
z_α is the $(1 - \frac{\alpha}{2})$ percentil in the $\mathcal{N}(0,1)$ distribution.
Also, in a similar way, the probabilities for the class values $p_i = P(C = c_i)$ are estimated with the next confidence interval,

$$\left(\hat{p}_i - z_\alpha \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}}; \hat{p}_i + z_\alpha \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}}\right) \tag{6}$$

where,
$\hat{p}^i_i$ is the point estimation of the probability $P(C = c_i)$
z_α is the $(1 - \frac{\alpha}{2})$ percentil in the $\mathcal{N}(0,1)$ distribution
N is the number of cases in dataset

– **Search space definition**
Once the confidence intervals are estimated from the dataset, it is possible to generate as many naïve Bayes classifiers as we want. The parameters of these naïve Bayes classifiers must only be taken inside theirs corresponding confidence intervals.
In this way, each naïve Bayes classifier is going to be represented with the next tupla of dimension $r_0(1 + \sum_{i=1}^{n} r_i)$

$$(p^*_1, \ldots, p^*_{r_0}, p^{*1}_{1,1}, \ldots, p^{*r_0}_{1,1}, \ldots, p^{*r_0}_{1,r_1}, \ldots, p^{*r_0}_{n,r_n}) \tag{7}$$

where each component in the tupla p^* is the selected value inside its corresponding confidence interval.
Thus, the search space for the heuristic optimization algorithm is composed of all the valid tuplas. A tupla is valid when it represents a valid naïve Bayes classifier. Formally,

$$\sum_{i=1}^{r_0} p^*_i = 1; \forall k \forall i \sum_{r=1}^{r_k} p^{*i}_{k,r} = 1 \tag{8}$$

Finally, each generated individual must be evaluated with a fitness function. This fitness function is based on the percentage of successful predictions on each dataset, which means that we are carrying out one wrapper approach.

– **Heuristic search for the best individual**
Once the individuals and the search space are defined, one heuristic optimization algorithm is ran in order to find the best individual.
To deal with the heuristic search, the continuous variant of EDAs –estimation of distribution algorithms– algorithms have been selected. EDAs [16,17] are

non-deterministic, stochastic and heuristic search strategies that belong to the evolutionary computation approaches. In EDAs, a number of solutions or individuals is created every generation, evolving once and again until a satisfactory solution is achieved. In brief, the characteristic that most differentiates EDAs from other evolutionary search strategies, such as GAs, is that the evolution from a generation to the next one is done by estimating the probability distribution of the fittest individuals, and afterwards by sampling the induced model. This avoids the use of crossing or mutation operators, and, therefore, the number of parameters that EDAs require is reduced considerably.

EDA

$D_0 \leftarrow$ Generate M individuals (the initial population) randomly

Repeat for $l = 1, 2, \ldots$ until a stopping criterion is met

$D_{l-1}^S \leftarrow$ Select $S \leq M$ individuals from D_{l-1} according to a selection method

$\rho_l(\boldsymbol{x}) = \rho(\boldsymbol{x}|D_{l-1}^S) \leftarrow$ Estimate the probability distribution of an individual being among the selected individuals

$D_l \leftarrow$ Sample M individuals (the new population) from $\rho_l(\boldsymbol{x})$

Fig. 1. Pseudocode for the EDA approach

In EDAs, the variables belonging to an individual are analyzed looking for dependencies. Also, while in other heuristics from evolutionary computation the interrelations among the different variables representing the individuals are kept in mind implicitly (e.g. building block hypothesis), in EDAs the interrelations are expressed explicitly through the joint probability distribution associated with the individuals selected at each iteration. The task of estimating the joint probability distribution associated with the database of the selected individuals from the previous generation constitutes the hardest work to perform, as this requires the adaptation of methods to learn models from data developed in the domain of probabilistic graphical models.

Figure 1 shows the pseudocode of EDA, in which the main steps of this approach are sketched:

1. At the beginning, the first population D_0 of M individuals is generated, usually by assuming a uniform distribution (either discrete or continuous) on each variable, and evaluating each of the individuals.
2. Secondly, a number S ($S \leq M$) of individuals are selected, usually the fittest.
3. Thirdly, the n–dimensional probabilistic model expressed better the interdependencies between the n variables is induced.

4. Next, the new population of M new individuals is obtained by simulating the probability distribution learnt in the previous step.

Steps 2, 3 and 4 are repeated until a stopping condition is verified. The most important step of this new paradigm is to find the interdependencies between the variables (step 3). This task will be done using techniques from the field of probabilistic graphical models.

In the particular case where every variable in the individuals are continuous and follows a gaussian distribution, the probabilistic graphical model is called *Gaussian network*.

Figure 2 shows the result of the execution of the IENB algorithm in the *pima* dataset with a z_α value of 1.96. In the figure can be observed all the confidence intervals calculated for this dataset and the final values selected by the algorithm.

In this execution IENB gets a result of 79.84% of successful classified while naïve Bayes gets 77.73%.

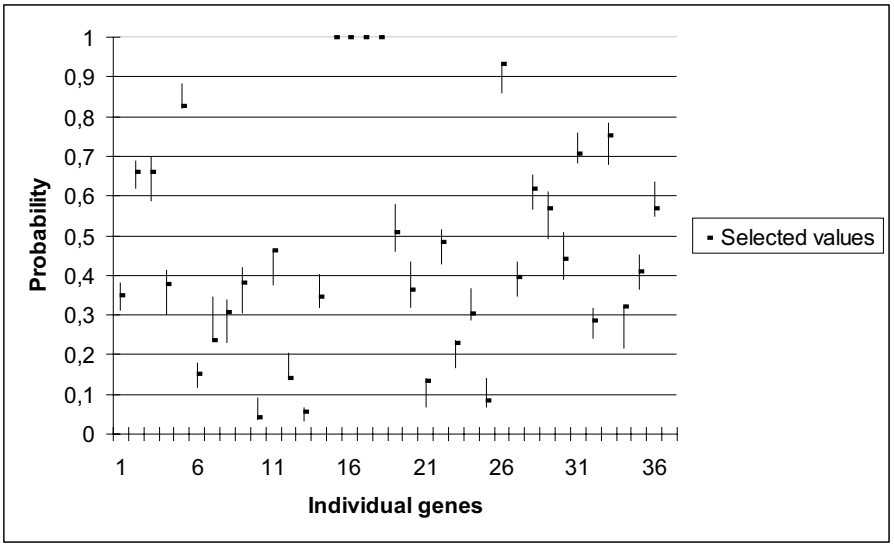

Fig. 2. Confidence intervals and final values obtained by IENB for the *pima* dataset

5 Experimentation

5.1 Datasets

Results are compared for 21 classical datasets –see table 1–, also used by other authors [6]. All the datasets belong to the UCI repository [18], with the exception of *m-of-n-3-7-10* and *corral*. These two artificial datasets, with irrelevant

and correlated attributes, were designed to evaluate methods for feature subset selection [10].

Table 1. Description of the data sets used in the experiments

	Attributes				Instances	
Name	**Total**	**Continuous**	**Nominal**	**Classes**	**Learning**	**Validation**
breast	10	10	-	2	699	-
chess	36	-	36	2	3196	-
cleve	13	6	7	2	303	-
corral	6	-	6	2	128	-
crx	15	6	9	2	692	-
flare	10	2	8	2	1066	-
german	20	7	13	2	1000	-
glass	9	9	-	7	214	-
glass2	9	9	-	2	163	-
hepatitis	19	6	13	2	155	-
iris	4	4	-	3	150	-
lymphography	18	3	15	4	148	-
m-of-n-3-7-10	10	-	10	2	300	1024
pima	8	8	-	2	768	-
satimage	36	36	-	6	6435	-
segment	19	19	-	7	2310	-
shuttle-small	9	9	-	7	5800	-
soybean-large	35	-	35	19	683	-
vehicle	18	18	-	4	846	-
vote	16	-	16	2	435	-
waveform-21	21	21	-	3	300	4700

5.2 Experimental Methodology

To estimate the prediction accuracy for each classifier our own implementation of a naïve Bayes classifier has been programmed. This implementation uses the Laplace correction for the point estimation of the conditional probabilities [8,10] and deals with missing values as recommended by [1].

However, our new algorithm does not handle continuous attributes. Thus, a discretization step with the method suggested by [2] has been performed usign MLC++ tools [12]. This discretization method is described by Ting in [21] that is a global variant of the method of Fayyad and Irani [4].

Nineteen of the datasets has no division between training and testing sets. On these datasets the results are obtained by a *leave-one-out* method inside of the heuristic optimization loop.

On the other hand, two out of these twenty one datasets include separated training and testing sets. For these cases, the heuristic optimization algorithm

uses only the training set to tune the classifier. A *leave-one-out* validation is perfomed internally inside of the optimization loop, in order to find the best classifier. Once the best candidate is selected, it is validated using the testing set.

5.3 Results

Experiments were ran in a Athlon 1700+ with 256MB of RAM memory. The parameters used to run EDAs were: population size 500 individuals, selected individuals for learning 500, new individuals on each generation 1000, learning type UMDA (*Univariate Marginal Distribution Algorithm*) [17] and elitism. Experiments were ran 10 times with the percentile 0.95 ($z_\alpha = 1.96$).

Table 2. Experiment results for Interval Estimation naïve Bayes

Dataset	*Naïve Bayes*	*IENB*	*Time(min)*	*Improv.*	*Iter. Bayes impr*
breast	97.14	97.71 ± 0.00 †	1	0.57	-0.16
chess	87.92	93.31 ± 0.10 †	221	5.39	
cleve	83.82	86.29 ± 0.17 †	2	2.47	0.1
corral	84.37	93.70 ± 0.00 †	1	9.33	
crx	86.23	89.64 ± 0.10 †	14	3.41	
flare	80.86	82.69 ± 0.13 †	3	1.83	
german	75.40	81.57 ± 0.10 †	81	6.17	-0.05
glass	74.77	83.00 ± 0.20 †	2	8.23	-1.19
glass2	82.21	88.27 ± 0.00 †	0	6.06	
hepatitis	85.16	92.60 ± 0.34 †	1	7.44	1.41
iris	94.67	95.97 ± 0.00 †	0	1.30	1.4 †
lymphography	85.14	94.56 ± 0.00 †	5	9.42	
monf-3-7-10	86.33	95.31 ± 0.00 †	64	8.98	
pima	77.73	79.84 ± 0.09 †	2	2.11	
satimage	82.46	83.88 ± 0.32 †	496	1.42	3.59 †
segment	91.95	96.38 ± 0.08 †	692	4.43	1.5 †
shuttle-small	99.36	99.90 ± 0.00 †	32	0.54	
soybean-large	92.83	95.67 ± 0.10 †	585	2.84	
vehicle	61.47	71.16 ± 0.25 †	64	9.69	4.39 †
vote	90.11	95.07 ± 0.19 †	4	4.96	1.45 †
waveform-21	78.85	79.81 ± 0.10 †	4	0.96	

Results for the UCI problems are shown in the table 2. Columns in the table are: first, the value obtained by the naïve Bayes algorithm, second, the mean value ± standard deviation from IENB, third, the average time –in minutes– for the executions, fourth, the improvement (*IENB-Naïve Bayes*) and finally the improvement respect to naïve Bayes obtained by *Iterative Bayes*, the most similar semi naïve Bayes approach.

Results are really interesting. Respect to naïve Bayes, using the 0.95 percentile, we obtained an average improvement of 4.30%. Besides, although this

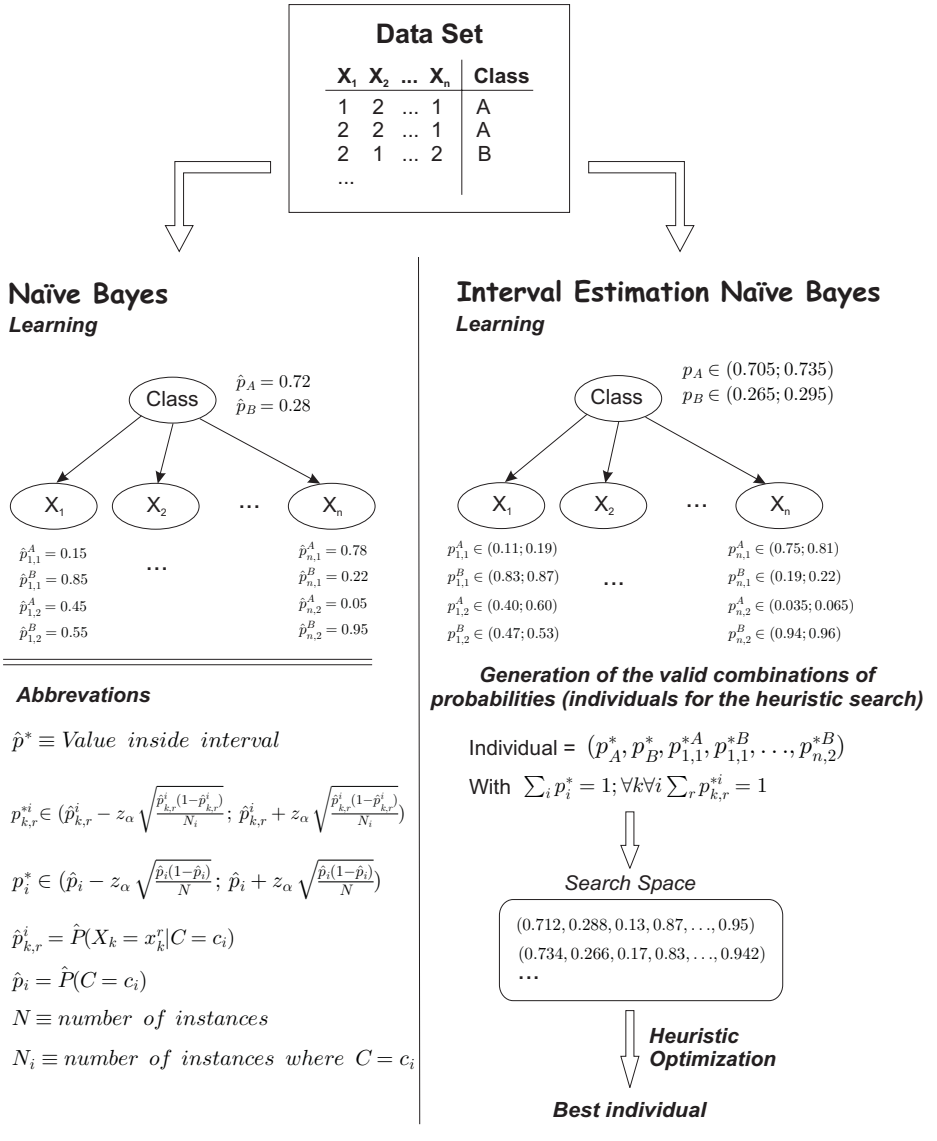

Fig. 3. Illustration of the differences between naïve Bayes and Interval Estimation naïve Bayes

is not always true, better improvements are obtained in the datasets with less number of cases, as the complexity of the problem is lower.

The non-parametric tests of Kruskal-Wallis and Mann-Whitney were used to test the null hypothesis of the same distribution densities for all of them. This

task was done with the statistical package S.P.S.S. release 11.50. The results for the tests applied to all the algorithms are shown below.

- Between naïve Bayes algorithm and IENB:
  - Naïve Bayes vs. IENB. Fitness value: $p < 0.001$

These results show that the differences between naïve Bayes and Interval Estimation naïve Bayes are significant in all the datasets, meaning that the behavior of selecting naïve Bayes or IENB is very different.

6 Conclusion and Further Work

In this work a new semi naïve Bayes approach has been presented. In our experiments this approach has an average improvement of 4.30% respect to the simple naïve Bayes.

As this is the first time we use this approach, many issues remain for future research. For instance, Interval Estimation naïve Bayes can also be used with continuous variables. It is possible to estimate intervals for the parameters μ and σ of the Gaussian distribution.

IENB can also be executed with different percentiles, in order to enlarge the confidence intervals. This also means an increment in the search space of the heuristic algorithm and therefore the execution time also increases.

The accuracy increment for the cases with correlated and redundant attributes (*corral* and *m-of-n-3-7-10*) is much better than the other cases. This can be interpreted as an improvement in terms of the assumptions problems of the original naïve Bayes.

Also, it is possible to change the objective of the heuristic search. We can try to maximize the area under the ROC curve instead of the percentage of successful predictions. Another factible idea is to combine Interval Estimation naïve Bayes with a feature subset selection. On a first phase it is possible to make a subset selection, and on a second phase to apply interval estimation to the previous results.

References

1. P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, pages 105–112, 1996.
2. J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202, 1995.
3. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
4. U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Conference on Artificial Intelligence*, pages 1022–1027, 1993.

5. J.T.A.S. Ferreira, D.G.T. Denison, and D.J. Hand. Weighted naïve Bayes modelling for data mining. Technical report, Deparment of Mathematics, Imperial College, May 2001.
6. N. Friedman, D. Geiger, and D.M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.
7. J. Gama. Iterative Bayes. *Intelligent Data Analysis*, 4:475–488, 2000.
8. I.J. Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. MIT Press, 1965.
9. D.J. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.
10. J. Kohavi, B. Becker, and D. Sommerfield. Improving simple Bayes. Technical report, Data Mining and Visualization Group, Silicon Graphics, 1997.
11. R. Kohavi. Scaling up the accuracy of naïve-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207, 1996.
12. R. Kohavi, G. John, R. Long, D. Manley, and K.Pfleger. MLC++: A machine learning library in C++. *Tools with Artificial Intelligence. IEEE Computer Society Press*, pages 740–743, 1994.
13. I. Kononenko. Semi-naïve Bayesian classifier. In *Sixth European Working Session on Learning*, pages 206–219, 1991.
14. P. Langley. Induction of recursive Bayesian classifiers. In *European Conference on Machine Learning. Berlin: Springer-Verlag*, pages 153–164, 1993.
15. P. Langley and S. Sage. Induction of selective Bayesian classifiers. In Morgan Kaufmann, editor, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406, Seattle, WA, 1994.
16. P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.
17. H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.
18. P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/, 1995.
19. M. Pazzani. Searching for dependencies in Bayesian classifiers. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 239–248, 1996.
20. M. Ramoni and P. Sebastiani. Robust learning with missing data. *Machine Learning*, 45(2):147–170, 2001.
21. K.M. Ting. Discretization of continuous-valued attributes and instance-based learning. Technical Report 491, University of Sydney, 1994.
22. G.I. Webb and M.J. Pazzani. Adjusted probability naïve Bayesian induction. In *Australian Joint Conference on Artificial Intelligence*, pages 285–295, 1998.
23. M. Zaffalon, K. Wesnes, and O. Petrini. Credal classification for dementia screening. *Artificial Intelligence in Medicine, LNAI 2101*, pages 67–76, 2001.
24. W. Zhou and R. Greiner. Learning accurate belief nets. Technical report, University of Alberta, 1999.