SUCCESSIVE NAIVE BAYESIAN CLASSIFIER

Igor Kononenko
University of Ljubljana, Faculty of Electrical & Computer Engineering,
Tržaška 25, 61001 Ljubljana, Slovenia
e-mail: igor.kononenko@ninurta.fer.uni-lj.si

Keywords: naive Bayesian classifier, successive learning, non-linear problems, empirical learning, empirical evaluation

Edited by: Igor Mozetič
Received: April 21, 1993 **Revised:** May 27, 1993 **Accepted:** July 21, 1993

The naive Bayesian classifier is fast and incremental, can deal with discrete and continuous attributes, has excellent performance in real-life problems and can explain its decisions as the sum of information gains. However, its naivety may result in poor performance in domains with strong dependencies among attributes. In this paper, the algorithm of the naive Bayesian classifier is applied successively enabling it to solve also non-linear problems while retaining all the advantages of naive Bayes. The comparison of performance in various domains confirms the advantages of successive learning and suggests its application to other learning algorithms.

1 Introduction

Let $A_i, i = 1 \dots n$ be a set of attributes, each having values $V_{i,j}, j = 1 \dots NV_i$. Let C_j be one out of k possible classes. An object is described with vector $X = (X_1, ..., X_n)$ where X_i may have one of values $V_{i,j}, j = 1 \dots NV_i$. Let an object with unknown class be described with $X^l = (X_1^l, ..., X_n^l)$. If the conditional independence of attributes with respect to all classes is assumed, the naive Bayesian formula can be used to classify such an object:

$$\hat{P}(C_j|X = X^l) = P(C_j) \prod_{i=1}^{n} \frac{P(C_j|X_i = X_i^l)}{P(C_j)} \quad (1)$$

where prior and conditional probabilities on the right-hand side can be approximated from a set of training examples with known classes. An object is classified by class with maximal probability calculated with (1).

If a limited number of training data are available, the approximation of probabilities with relative frequency becomes unreliable. Cestnik (1990) has shown that instead of using relative frequencies, it is more appropriate to use the *m-estimate* of conditional probabilities:

$$\hat{P}(C_j|X_i = V_{i,J_i}) = \frac{N_{C_j,V_{i,J_i}} + m \times \hat{P}(C_j)}{N_{V_{i,J_i}} + m}, \quad j = 1..k \quad (2)$$

and Laplace's law of succession (Good, 1950) for prior probabilities of k classes:

$$\hat{P}(C_j) = \frac{N_{C_j} + 1}{N + k}, \quad j = 1..k \quad (3)$$

In the above formulas $N_{C_j,V_{i,J_i}}$ represents the number of training instances with value V_{i,J_i} of the i-th attribute and belonging to the j-th class, N_{C_j} and $N_{V_{i,J_i}}$ are interpreted in a similar manner and N is the number of all training instances. The same formula was also used by Smyth and Goodman (1990). Parameter m trades off the relative frequency and the prior probability. A lower setting for m suggests stronger belief in training data, whereas a higher setting implies greater reliance on prior probabilities. In our experiments described in section 4, parameter m was set to 2, which is an empirically verified appropriate choice for a typical learning problem (Cestnik, 1990).

The naive Bayesian formula applies to discrete attributes, while continuous attributes have to be discretized in advance. It was shown that fuzzy discretization for modeling continuous attributes achieves a better performance and that the results are less sensitive with respect to factual discretization (Kononenko, 1991). In experiments described in this paper, fuzzy discretization was not applied.

Many authors have experimentally verified that the naive Bayesian formula achieves better or at least as good classification accuracy as inductive learning algorithms in many real-world problems (Kononenko et al., 1984; Cestnik, 1990; Smyth and Goodman, 1990) and, surprisingly, the explanation ability of naive Bayes, at least in inexact domains such as medical diagnosis, is better (as estimated by physicians) than that of a decision tree (Kononenko, 1990). The kind of explanation by naive Bayes is the *sum of information gains* by each attribute for/against each class, which is obtained with logarithm of eq. (1):

$$-\log_2 \hat{P}(C_j|X=X^l) = -\log_2 P(C_j)$$
$$-\sum_{i=1}^{n}\left(\log_2 P(C_j|X_i=X_i^l) - \log_2 P(C_j)\right) \quad (4)$$

The explanation can be presented to human experts as a list of attribute values with corresponding information gains for each class that appear in the sum on the right-hand side of equation (4). Human experts appeared to prefer explanations of this type to a single if-then rule for a classified object.

However, the naivety of formula (1) can be too drastic in certain domains with strong dependencies among attributes. A classical non-linear problem which cannot be solved by naive Bayes is exclusive "or" (XOR):

$$Class(X) = \begin{cases} C_1, & X_1 \neq X_2 \\ C_2, & X_1 = X_2 \end{cases} \quad (5)$$

This problem is also hard for other machine-learning algorithms. One class of hard machine-learning problems contains parity problems of higher degrees. This paper describes a method for successive application of naive Bayes which under certain conditions can solve parity problems. In the next section, the theoretical limitations of naive Bayes are briefly discussed and results on well-known problems are compared to those of other learning algorithms. In Section 3, the successive naive Bayesian classifier is described and section 4 gives empirical results on various problem domains. In the discussion, a generalization of the approach to other learning algorithms is proposed.

2 Performance of naive Bayes

Despite its limitations, the performance of the naive Bayesian classifier in many real-world problems is excellent compared to that of other learning algorithms. In table 1 the performance on three well-known medical diagnostic problems - primary tumor, breast cancer, and lymphography - is compared to that of other propositional-logic learning algorithms. We also tested naive Bayes on the problem of finite element mesh design (Dolšak & Muggleton, 1991), which has been a focus of the inductive logic programming (ILP) community (see section 4 for descriptions of learning data). Although it cannot use information about geometric properties of objects in this domain (this holds for all propositional logic algorithms) it outperformed all sophisticated ILP algorithms. The comparison is given in table 2.

Let us now examine more closely the limitations of the naive Bayesian classifier. With appropriate recoding of objects, the naive Bayesian classifier can also be interpreted as a linear function which discriminates between two classes C_i and C_j:

$$Class(Y) = \begin{cases} C_i, & P^{i,j}Y > 0 \\ C_j, & P^{i,j}Y < 0 \end{cases} \quad (6)$$

where
vector $Y = (1, Y_{1,1}, .., Y_{1,NV_1}, ..., Y_{n,1}, .., Y_{n,NV_n})$ is recoded vector X so that each attribute's value corresponds to one vector's component:

$$Y_{i,j} = \begin{cases} 1, & X_i = V_{i,j} \\ 0, & X_i \neq V_{i,j} \end{cases} \quad (7)$$

and $P^{i,j}Y$ is the inner product of vector $P^{i,j}$, which is used to discriminate between classes C_i and C_j, and vector Y. The term $P^{i,j}$ is obtained by subtracting two instances of eq. (4):

$$P^{i,j} = (p_0^{i,j}, p_{1,1}^{i,j}, .., p_{1,NV_1}^{i,j}, ..., p_{n,1}^{i,j}, .., p_{n,NV_n}^{i,j}) \quad (8)$$

where

$$p_0^{i,j} = \log_2 P(C_i) - \log_2 P(C_j)$$

algorithm	reference	prim.tumor	brea.cancer	lymphogr.
Assistant	(Kononenko et al., 1984)	44 %	73 %	77 %
AQ15	(Michalski et al., 1986)	41 %	66 %	80 %
Assistant 86	(Cestnik et al., 1987)	44 %	77 %	76 %
LogArt	(Cestnik & Bratko, 1988)	44 %	78 %	84 %
CN2	(Clark & Boswell, 1991)	46 %	73 %	82 %
naive Bayes		51 %	79 %	84 %

Table 1: Performance of different algorithms in three medical domains.

algorithm	reference	accuracy
FOIL	(Quinlan, 1990)	12 %
mFOIL	(Džeroski, 1991)	22 %
GOLEM	(Dolšak & Muggleton, 1991)	29 %
LINUS	(Lavrač & Džeroski, 1991)	29 %
naive Bayes		33 %

Table 2: Performance of different algorithms in finite element mesh problem.

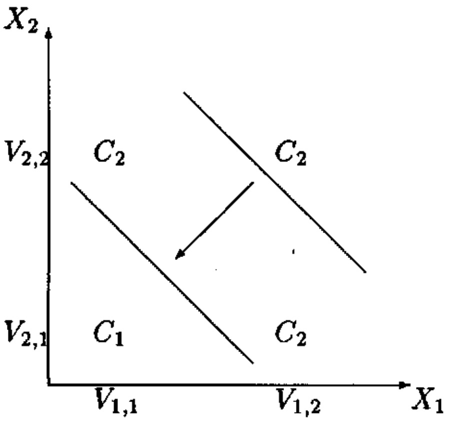


Figure 1 Applying delta learning rule to naive Bayes

and

$$p_{l,m}^{i,j} = \log_2 \frac{P(C_i|X_l = V_{l,m})}{P(C_i)} - \log_2 \frac{P(C_j|X_l = V_{l,m})}{P(C_j)}$$

For each pair of classes we have one linear discrimination function. This derivation confirms that the naive Bayesian classifier is limited to linear decision functions, which cannot solve non-linear problems.

This raises the question of whether it is worth using a delta learning rule, in the sense of perceptrons (Minsky & Papert, 1969), to adapt the discrimination function to discriminate more reliably between classes on training data. This can be achieved by iteratively duplicating training instances not correctly classified by naive Bayes. This alters the probability distribution so that the discrimination function moves in an appropriate direction. This is illustrated in Fig. 1 where, by duplicating instances $(V_{1,1}, V_{2,2})$ and $(V_{1,2}, V_{2,1})$, the discrimination function is changed into a perfect discriminator.

However, such changes in probability distribution are inappropriate for predicting cases unseen during learning. The decision function in fact overfits the training data which affects the performance on unseen cases. We tried the above (delta) learning rule on several medical diagnostic problems. The performance on training data increased (in lymphography it even reached 100% classification accuracy), but the performance on test data of the classifier drastically decreased. This suggests that the decision function given by the original probability distribution is optimal among linear discrimination functions.

One should be careful when changing the representation space. In fact, if Y representation is used instead of X, the space is much sparser, as many points are illegal (an instance cannot have more than one attribute's value). On the other hand, in the original space (X coding) in general the discrimination function of naive Bayes is not linear. For clarity of illustration, in the next section we will assume approximately linear discrimination functions in the original space.

3 Successive learning with naive Bayesian classifier

If one tries to solve the XOR problem with the naive Bayesian classifier, the result may be one of the decision curves (a or b) from Figure 2.1. The direction of the curve depends on the distribution of training instances. However, if all instances are equally likely, no decision curve appears, as all components of $P^{1,2}$ are equal to 0. In such a case, it is desirable to modify slightly the distribution (e.g., by duplicating one of the instances) to get one of the decision curves (i.e. breaking the symmetry).

To enable the naive Bayesian classifier to solve the XOR problem, the same algorithm may be repeatedly applied, each time on a redefined problem. In each iteration, training instances that are correctly classified by the current discrimination function are assigned to an additional special class C_0 and the other training instances retain their original classes. The resulting learning tasks of the XOR problem (depending on the current discrimination function) and their solutions are depicted in Fig. 2.2a, and 2.2b. There is one discrimination function for each pair of classes (labeled with their indices). In both cases, the discrimination is perfect after two successive learning iterations. However, for parity problems of higher order, more iterations may be needed. In general, it is not always possible to obtain perfect discrimination with such successive learning. On the other hand, perfect discrimination of training instances usually implies overfitting. This is avoided here by keeping all training instances for each new learning problem, thus enabling reliable probability estimates. Overfitting the training data may be interpreted as reliance on unreliable probability estimates from a small number of training instances. Fig. 3 illustrates successive learning in a parity problem involving two three-valued attributes and three classes. in Fig. 3 (1), two discrimination lines (between classes 2 and 3 (2-3) and 2 and 1 (2-1)) overlap.

The above discussion leads to the following learning algorithm:

```
repeat
    Train naive Bayes;
    Change the class label of correctly classified
        training instances to $C_0$;
until all training instances are classified to $C_0$
```

Note that the terminating condition does not require perfect classification (i.e. not all training instances need be correctly classified). This algorithm may enter an infinite loop if perfect discrimination is impossible. However, more iterations do not cause overfitting of the training data, as all training instances are used in all iterations. For practical reasons, it is necessary to limit the number of iterations. In our experiments described in the next section, the number of iterations was limited to ten.

When classifying new objects, the discrimination function learned last should be tried first. If the result is class C_0 the next latest function must be tried, while if the result is one of the original classes, it is accepted as an answer. The reverse order for classification follows from the training algorithm, because the classification into a class other than C_0 is more reliable with the latest discrimination function. Eventually, by repeted application of discrimination functions in reverse order, a class $C_i, i > 0$ is obtained as an answer.

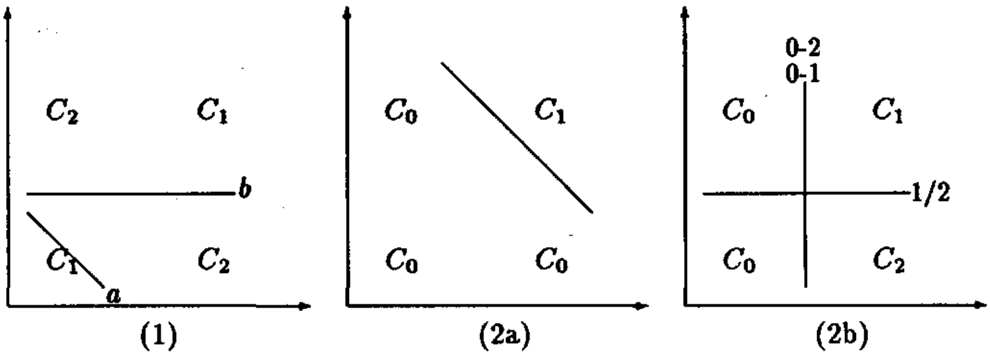
4 Experimental results

We applied successive naive Bayesian learning to several data sets from medical diagnostics, chess endgame, criminology, engineering, and one artificial data set. Basic data characteristics are given in table 3. A brief description of each problem follows:

Primary tumor: Locate the position of the primary tumor in the body of a patient with metastases.

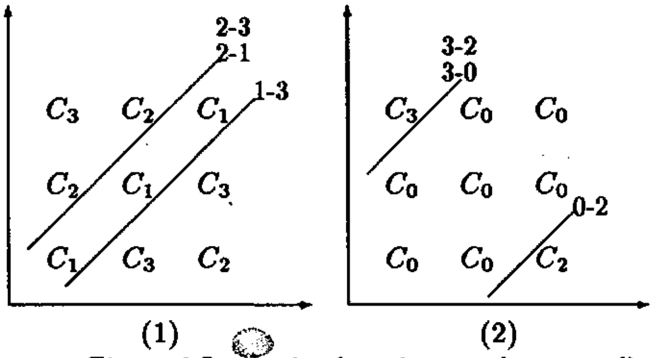
Breast cancer: Predict the recurrence of the disease in five years after the operation.

Lymphography: Classify the type of tumor of a patient with metastases.



(1) (2a) (2b)

Figure 2
(1) Original XOR problem
(2a) New problem obtained from discrimination function *a*
(2b) New problem obtained from discrimination function *b*



(1) (2)

Figure 3 Successive learning on the generalized parity problem

domain	#class	#atts.	#val/att.	# instances
primary tumor	22	17	2.2	339
breast cancer	2	10	2.7	288
lymphography	4	18	3.3	148
rheumatology	6	32	9.1	355
criminology	4	11	4.5	723
chess 1	2	6	8.0	1000
chess 2	2	18	2.0	1000
mesh 1	13	3	7.0	278
mesh 2	13	15	7.3	278
artificial	2	12	2.0	200

Table 3: Basic description of data sets

Rheumatology: Determine the type of rheumatologic disease.

Criminology: Determine the education of the violator.

Chess 1: Detect illegal positions in King-Rook-King chess endgame given only the coordinates of pieces.

Chess 2: Detect illegal positions in King-Rook-King chess endgame given the relations such as "same rank", "neighbour file", etc.

Mesh 1: Determine the number of elements of an edge in a finite element mesh design, given the three basic attributes but no geometric relations.

Mesh 2: Determine the number of elements of an edge in a finite element mesh design, given the three basic attributes and additional attributes such as the number and the type of neighbour edges.

Artificial: A data set was generated with two attributes defining parity relation with class, 5 additional random binary attributes and 5 additional independent and slightly informative binary attributes. In addition, class labels were corrupted with 5% noise (5% of cases had wrong class labels).

Except in the "mesh" problems, the experiments were performed with 10 random splits on 70% of the instances for training and 30 % for testing. The results were averaged. In "mesh" problems, experiments were done in the same way as with ILP systems (see Džeroski, 1991, for details). The measured parameters were:

- accuracy: the percentage of correctly classified instances
- average information score (Kononenko & Bratko, 1991): a measure that eliminates the influence of prior probabilities. It is defined as follows:

$$Inf = \frac{\sum_{i=1}^{\#testing\ instances} Inf_i}{\#testing\ instances} \qquad (9)$$

where information score of classification of the i-th testing instance is defined by (10):

problem	naive Bayes		successive Bayes	
	%	bit	%	bit
primary tumor	51.0	1.57	51.7	1.61
breast cancer	79.2	0.18	78.4	0.16
lymphography	84.2	0.83	83.9	0.82
rheumatology	67.2	0.51	68.3	0.53
criminology	61.2	0.27	61.5	0.27
chess 1	66.2	0.18	66.5	0.18
chess 2	91.7	0.73	92.3	0.75
mesh 1	33.5	0.61	32.4	0.60
mesh 2	34.5	0.62	36.0	0.66
artificial	61.8	0.24	78.3	0.57

Table 4: Results of naive and successive naive learning.

where Cl_i is the class of i-th testing instance, $P(Cl)$ is the prior probability of class Cl and $P'(Cl)$ the probability returned by a classifier.

Results are summarized in table 4.

The results indicate that the performance of successive learning is the same as that of naive Bayes in most real-world domains. The only significant difference according to accuracy and information score appears in "primary tumor" and "mesh 2" problems. Both problems are very difficult (see tables 1 and 2), involving many classes. The result with the artificial data set indicates that successive learning may be much better in domains with strong dependencies among attributes.

5 Discussion

The results of experiments suggest that successive naive Bayesian learning may improve the performance of naive Bayes while preserving its advantages: simplicity, efficiency and transparency. The successive learning approach keeps all training instances together in all learning iterations and thus avoids the overfitting problem. However, the algorithm may reach a non-solvable learning problem. Covering algorithms (e.g. AQ, Assistant, CN2 and FOIL) discard correctly covered training instances in each iteration and are able to discriminate cases in any learning problem, but with great danger of overfitting the training data.

$$Inf_i = \begin{cases} -\log_2 P(Cl_i) + \log_2 P'(Cl_i), & P'(Cl_i) \geq P(Cl_i) \\ -(-\log_2(1 - P(Cl_i)) + \log_2(1 - P'(Cl_i))), & P'(Cl_i) < P(Cl_i) \end{cases} \tag{10}$$

The principle of successive learning can be used with any learning algorithm and, probably more efficiently, different learning algorithms may be successively applied. Further investigations should empirically verify this hypothesis, as well as the idea of combining the successive and covering approaches. More theoretical work is needed to determine the limitations of successive learning and answer questions such as: which problems cannot be solved with successive learning and which problems lead the approach into infinite cycling.

6 Acknowledgements

Medical data sets were obtained from the University Medical Center in Ljubljana, and were provided by Dr. Matjaž Zwitter (primary tumor and breast cancer), Dr. Milan Soklič (lymphography), and Dr. Vladimir Pirnat (rheumatology). The KRK endgame data set is the first of five KRK data sets used by Sašo Džeroski for testing rules generated by mFOIL (Džeroski, 1991); the data sets were originally provided by Michael Bain. The Mesh data were provided by Bojan Dolšak. Boštjan Vovk performed experiments with delta learning rule described in section 2. This research was supported by the Slovenian Ministry of Science and Technology.

References

[1] Cestnik, B. (1990) Estimating probabilities: A crucial task in machine learning, *Proc. European Conference on Artificial Intelligence*, Stockholm, August 1990, pp.147-149.

[2] Cestnik, B. & Bratko, I. (1988) Learning redundant rules in noisy domains, *Proc. European Conf. on Artificial Intelligence ECAI-88*, München, August 1988, pp.348-350.

[3] Cestnik, B., Kononenko, I., Bratko, I. (1987) ASSISTANT 86 : A knowledge elicitation tool for sophisticated users. In: I. Bratko, N. Lavrač (eds.), *Progress in Machine learning*. Wilmslow, England: Sigma Press.

[4] Clark P. & Boswell R. (1991) Rule induction with CN2: Recent improvements, *Proc. EWSL 91*, Porto, March 1991, pp.151-163.

[5] Dolšak, B. & Muggleton, S. (1991) The application of inductive logic programming to finite element mesh design. *Proc. 8th Inter. Workshop on Machine Learning*, Evanstone: Morgan Kaufmann.

[6] Džeroski S. (1991) Handling noise in inductive logic programming, M.Sc. Thesis, University of Ljubljana, Faculty of Electrical & Computer Engineering, Ljubljana.

[7] Good I.J. (1950) *Probability and the weighing of evidence*. London: Charles Griffin.

[8] Kononenko, I. (1990) Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In: B. Wielinga et al. (eds.) *Current Trends in Knowledge Acquisition*, Amsterdam: IOS Press.

[9] Kononenko, I. (1991) Feedforward Bayesian neural networks and continuous attributes, *Proc. Int. Joint Conf. on Neural Networks IJCNN-91*, Singapore, Nov. 1991, pp. 146-151.

[10] Kononenko, I. & Bratko, I. (1991) Information based evaluation criterion for classifier's performance. *Machine Learning*, Vol.6, pp.67-80.

[11] Kononenko, I., Bratko, I., Roškar, E. (1984) Experiments in automatic learning of medical diagnostic rules, Technical report, Faculty of Electrical & Computer Eng., Ljubljana, Slovenia (presented at ISSEK Workshop, Bled, August, 1984).

[12] Lavrač, N. & Džeroski, S. (1991) Inductive learning of relations from noisy examples, In: Muggleton, S. (ed.) *Inductive logic programming*, Glasgow: Academic Press.

[13] Michalski, R.S., Mozetič, I., Hong, J., Lavrač, N. (1986) The multi purpose incremental learning system AQ15 and its testing

application to three medical domains. *Proc. of the National Conf. on Artificial Intelligence AAAI 86*. Philadelphia, August, 1986, pp.1041-1047.

[14] Minsky, M. & Papert, S. (1969) *Perceptrons*, Cambridge, MA : MIT Press.

[15] Quinlan (1990) Learning logical definitions from relations, *Machine Learning*, Vol. 5, No. 3, pp.239-266.

[16] Smyth P. & Goodman R.M. (1990) Rule induction using information theory. In: G.Piarersky & W.Frawley (eds.) *Knowledge Discovery in Databases*, MIT Press.