
Production Systems and Information Engineering
Volume 5 (2009), pp. 41-50.

ESTIMATION OF MISCLASSIFICATION ERROR USING BAYESIAN CLASSIFIERS

PÉTER BARABÁS
University of Miskolc, Hungary
Department of Information Technology
barabas@iit.uni-miskolc.hu

LÁSZLÓ KOVÁCS
University of Miskolc, Hungary
Department of Information Technology
kovacs@iit.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. Bayesian classifiers provide relatively good performance compared with other more complex algorithms. Misclassification ratio is very low for trained samples, but in the case of outliers the misclassification error may increase significantly. The usage of 'summation hack' method in Bayesian classification algorithm can reduce the misclassifications rate for untrained samples. The goal of this paper is to analyze the applicability of summation hack in Bayesian classifiers in general.

Keywords: Bayesian classifier, summation hack, polynomial distribution, misclassification error

1. Introduction

The Bayesian classification method is a generative statistical classifier. Studies comparing classification algorithms have found that the simple or Naive Bayesian classifier provides relatively good performance compared with other more complex algorithms. Accuracy of classification is a very important property of a classifier, a measure of which can be separated into two parts: a measure of accuracy in case of trained samples and a measure of accuracy in case of untrained samples. Naive Bayesian classification is generally very accurate in the first case since all testing samples are trained before and have no outliers; in the second case the efficiency is worse due to outliers. In [1], the role of outliers is examined in classification methods, the Naive Bayesian classification is reactive to outliers, and they can cause misclassification. Usage of summation hack can reduce the effect of outliers. The goal of our research is to analyze the generalization capability of Bayesian classification using summation hack. In the second part a short summary about Naive Bayesian classification is given. In the third part the concept of summation hack is introduced and examined. In the fourth part the classification methods are

analyzed considering the misclassification error. Finally, the test results and conclusions have been summarized in the last section.

It is assumed that the objects to be classified are described by n-dimensional pattern vectors $\boldsymbol{x} = (x_1,\ldots,x_n) \in \boldsymbol{R}^n$. The dimensions correspond to the attributes of the objects. Every pattern vector is associated with a class label c, where the total number of classes is m. The class label c_i denotes that the object belongs to the i-th class. Thus, a classifier can be regarded as a function

$$g(\mathbf{x}): R^n \rightarrow \{c_1,\ldots,c_m\}. \tag{1.1}$$

The optimal classification function is aimed at minimizing the misclassification risk [2]. The risk value R depends on the probability of the different classes and on the misclassification cost of the classes:

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \sum_{c_j} b(g(\mathbf{x}) \rightarrow c_j) \cdot P(c_j \mid \mathbf{x}), \tag{1.2}$$

where $P(c_j \mid \boldsymbol{x})$ denotes the conditional probability of c_j for the pattern vector $\boldsymbol{x}$ and $b(c_i \rightarrow c_j)$ denotes the cost value of deciding in favor of c_i instead of the correct class c_j. The cost function b has usually the following simplified form:

$$b(c_i \rightarrow c_j) = \begin{cases} 0, if & c_i = c_j \\ 1, if & c_i \neq c_j \end{cases}. \tag{1.3}$$

Using this kind of function b, the misclassification error value can be given by

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \sum_{g(x) \neq c_j} P(c_j \mid \mathbf{x}). \tag{1.4}$$

The optimal classification function minimizes the value $R(g(\boldsymbol{x}) \mid \boldsymbol{x})$. As

$$\sum_{c_j} P(c_j \mid \mathbf{x}) = 1, \tag{1.5}$$

thus if

$$P(g(\mathbf{x}) \mid \mathbf{x}) \rightarrow \max, \tag{1.6}$$

then the $R(g(\boldsymbol{x})|\boldsymbol{x})$ has a minimal value. The decision rule which minimizes the average risk is the Bayes' rule which assigns the $\boldsymbol{x}$ pattern vector to the class that has the greatest probability for $\boldsymbol{x}$[3].

2. Bayes classification

A Bayesian classifier is based on Bayes' theorem which relates to the conditional and marginal probabilities of two random events. Let A and B denote events. Conditional probability $P(A|B)$ is the probability of event A, given the occurrence of event B. Marginal probability is the unconditional probability $P(A)$ of event A, regardless of whether event B does or does not occur.

The simplified version of Bayesian theorem can be written for event A and B as follows:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{2.1}$$

If $\bar{A}$ is the complementary event of A, called „not A". Let $A_1, A_2, A_3, \ldots$ be a partition of the event space. The general form of the theorem is given as:

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_j P(B \mid A_j)P(A_j)}. \tag{2.2}$$

Let $C = \{c_k\}$ denote the set of classes. The observable properties of the objects are described by vector $\boldsymbol{x}$. An object with properties $\boldsymbol{x}$ has to be classified into the class for which the $P(c_k|\boldsymbol{x})$ probability is maximal. On the basis of Bayes' theorem:

$$P(c_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_k)P(c_k)}{P(\mathbf{x})}. \tag{2.3}$$

Since $P(\boldsymbol{x})$ is the same for all k we have to maximize only the expression $P(\boldsymbol{x}|c_k)P(c_k)$. The value $P(c_k)$ is given a priori or can be appreciated with relative frequencies from the samples. According to the assumption of Naive Bayes classification the attributes in a given class are conditionally independent of every other attribute. So the joint probability model can be expressed as

$$P(c_k, x_1, \ldots, x_n) = P(c_k)\prod_{i=1}^{n} P(x_i \mid c_k). \tag{2.4}$$

Using the above equation the probability of class c_k for an object featured by vector $\mathbf{x}$ is equal to

$$P(c_k \mid \mathbf{x}) = \frac{P(c_k)\prod_{i=1}^{n} P(x_i \mid c_k)}{P(\mathbf{x})}. \tag{2.5}$$

For the case where $P(c^*|\boldsymbol{x})$ is maximal the corresponding class label [5] is:

$$c^* = \operatorname{argmax}_{c \in C}\{P(c \mid \mathbf{x})\} = \operatorname{argmax}_{c \in C}\left\{P(c)\prod_{i=1}^{n} P(x_i \mid c)\right\}. \tag{2.6}$$

If a given class and feature never occur together in the training set, then the relative frequency will be zero. Thus, the total probability is also set to zero. One of the simplest solutions of this problem is to add 1 to all occurrences of the given attribute. In case of a large number of samples the distortion of probabilities is marginal and the information loss through the zero tag can be eliminated successfully. This technique is called Laplace estimation [4]. A more refined solution is to add p_k instead of 1 to the relative frequencies, where p_k is the relative frequency of k^{th} attribute value in the global teaching set, not only in the set belonging to class c_i.

3. Summation hack

Outliers in the classification can indicate faulty data which cause misclassification. The use of summation hack is an optional method to reduce the misclassification error. Summation hack is an ad-hoc replacement of a product by a sum in a probabilistic expression [1]. This hack is usually explained as a device to cope with outliers, with no formal derivation. This note shows that the hack does make sense probabilistically, and can be best thought of as replacing an outlier-sensitive likelihood with an outlier-tolerant one.

Let us define a vector $\boldsymbol{x}$ with components $x_1,x_2,\ldots,x_n$ and a class c. In Bayes classification where the vector values are conditionally independent:

$$P(\mathbf{x} \mid c) = \prod_{i=1}^{n} P(x_i \mid c). \tag{3.1}$$

In this case the probability is sensitive to outliers in individual dimensions so if any $P(x_i|c)$ value is equal to 0, the product will be zero. Using summation hack we get the following:

$$P(\mathbf{x} \mid c) \approx \sum_{i=1}^{n} P(x_i \mid c). \tag{3.2}$$

In this case the result will be zero if and only if all $p(x_i|c)$ values are equal to 0. Using (2.9) and (3.2) the computing of winner class is based upon the following formula:

$$c^* = \operatorname{argmax}_{c \in C} \{P(c \mid \mathbf{x})\} \approx \operatorname{argmax}_{c \in C} \left\{ P(c) \sum_{i=1}^{n} P(x_i \mid c) \right\}, \tag{3.3}$$

Applying summation hack the error of classification can be reduced. In every equation above the frequency probabilities are replaced with their approximated values, where

$$P(e) = \lim_{n \to \infty} \frac{n_e}{n_t}, \tag{3.4}$$

and n_t is the total number of trials and n_e is the number of trials where event e occurred. If the number of test events approaches infinity, the relative frequency value will converge to the probability value. In many classification tasks; a small number of samples is given [6], the number of tests is low, so a larger approximation error will arise in the calculations. We can write the probability as follows:

$$\frac{n_{x_i = v | c_j}}{n_{x_i}} = P(x_i = v \mid c_j) + \Delta_i, \tag{3.5}$$

where Δ_i means the error of approximation. The cumulated classification error in case of summation hack can be computed by the summation of the error elements. This error value differs from the classification error for the product of probabilities as it is calculated by the following form:

$$\prod_{i=1}^{n} (P(x_i = v \mid c_j) + \Delta_i) - \prod_{i=1}^{n} (P(x_i = v \mid c_j)). \tag{3.6}$$

4. Analysis of approximation error

The main cause of misclassification is the error of the approximated probability values shown in formula (3.4). To calculate the error value, the following model is applied. Let $\{c\}$ be the set of classes, and $\{a'\}$ the set of attributes where an attribute may be of vector value. A test case is described by a (a',c) pair where c denotes the class related to the a' attribute. The unknown probability that a_i belongs to c_j is denoted by p_{ij}. The relative frequency of the event that a_i belongs to c_j is denoted by g_{ij}. In the calculations p_{ij} are approximated by g_{ij}. The classification of the attribute can be regarded as a stochastic event, where $P(p_{ij}, g_{ij})$ denotes the probability that g_{ij} will be used in the calculations instead of g_{ij}.

Let $X(x_i)$ be a k-dimensional stochastic variable, where x_i denotes the number of attributes classified as c_i. X has a polynomial distribution:

$$P(x_1 = n_1, x_2 = n_2, ..., x_k = n_k) = \frac{N!}{n_1! n_2! ... n_k!} P_1^{n_1} P_2^{n_2} ... P_k^{n_k}, \tag{4.1}$$

where

$$N = \sum_{i=1}^{k} n_i, \sum_{i=1}^{k} P_i^{n_i} = 1. \tag{4.2}$$

A given $\boldsymbol{g}(n_1,n_2,...,n_r)$ frequency value has different P probabilities for the different $\boldsymbol{p}(p_1,p_2,...,p_r)$ probability tuples. The $\boldsymbol{p}(p_1,p_2,...,p_r)$ with maximal P value is assumed to be the real probability value tuple. As the maximum likelihood approximation of the probability is the frequency value, the relative frequencies are the best approximations of real probabilities:

$$P_i^{n_i} \approx \frac{n_i}{N}. \tag{4.3}$$

The probability of other $\boldsymbol{p}$ vectors can also be calculated with this formula. For the case $n = 2$ the resulting P distribution function is shown in Fig. 1.

In the next step, the approximation error of product $\prod P_i$ is calculated. It is clear that the larger the difference between $\boldsymbol{p}$ and $\boldsymbol{g}$, the higher the error value is. On the other hand, the lower the difference between p and g, the higher probability of this pair is. In the investigation, the average error value is calculated in the following way:

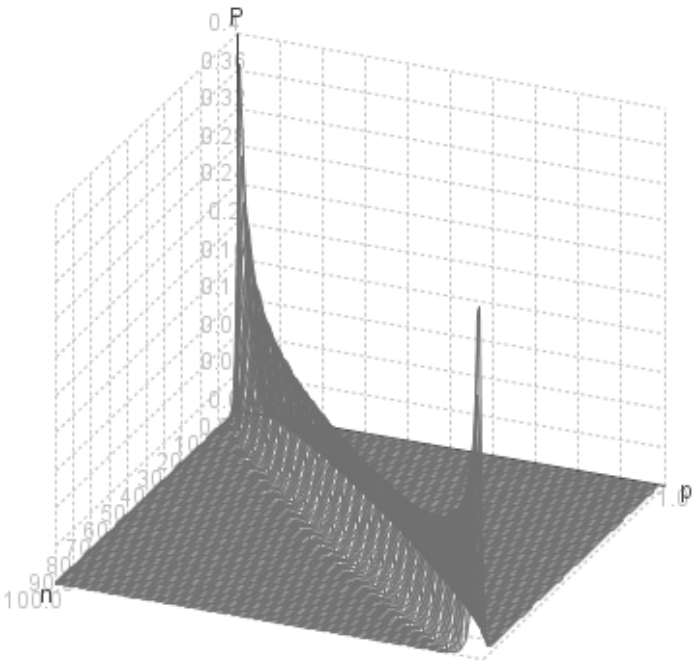
$$\varepsilon(\mathbf{g}) = \sum_{\mathbf{p}} P(\mathbf{p},\mathbf{g}) \cdot \varepsilon(\mathbf{p},\mathbf{g}), \tag{4.4}$$

where ε denotes the error value, where

$\varepsilon(\boldsymbol{p},\boldsymbol{g})$: the error value of matching $\boldsymbol{p}$ with $\boldsymbol{g}$,

$P(\boldsymbol{p},\boldsymbol{g})$: the probability of matching $\boldsymbol{p}$ with $\boldsymbol{g}$ and

$\varepsilon(\boldsymbol{g})$: the average error related to frequency vector $\boldsymbol{g}$.



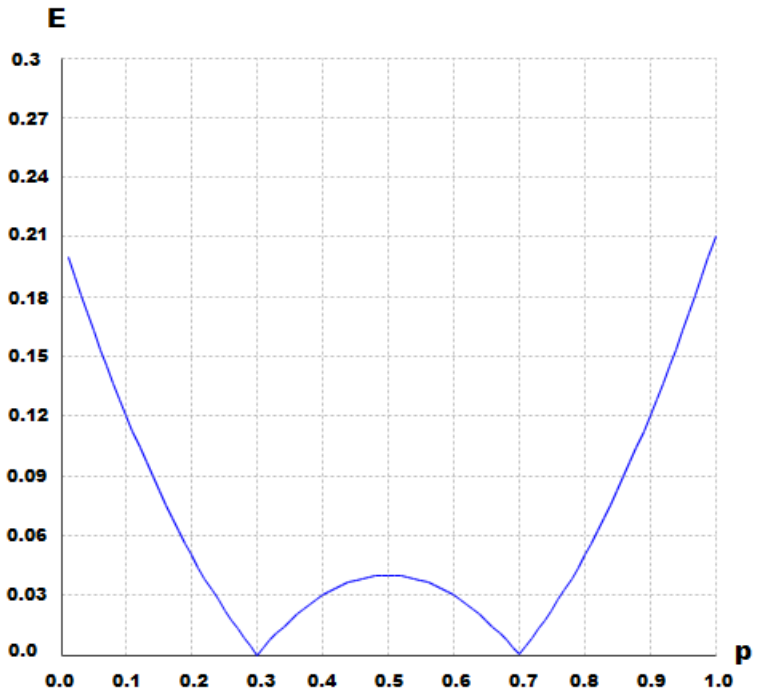
,

Figure 1. Probability function for the case n=2

In the test case, the error formula for p_i and the mean value of error can be computed as follows:

$$\varepsilon(\mathbf{p},\mathbf{g}) = p_1(1-p_1) - \frac{n_1}{N}(1-\frac{n_1}{N}), \tag{4.5}$$

Fig. 2 shows the error function for the test binomial case. The number of attempts is 100 where the number of attributes belonging to class c_1 is 30. In the Figure, can be seen the minimum error is in case of p=0.3. Since the function is symmetric, another minimum point can be found at p=0.7.



,

Figure 2. Error function for the case p=0.3

In Naive Bayesian classifier the accuracy depends strongly on the number of attempts. The larger the test pool, the better the accuracy is. In Fig. 3 the mean value error function can be seen for different N values. The results show that for a small number of N values the use of summation hack can improve the accuracy but for a larger test pool the Naive Bayesian classifier is the dominant one.

5. Test results

In first tests [7] the reference points were generated with uniform distribution in space. The winner was the Naive Bayesian classifier in teaching and testing phase equally. The teaching accuracy had values from 80% to 100% depending on environment parameters. Using summation hack this accuracy decreased by about 10%. The testing accuracy is far lower, it is between 40 and 70 percent in case of Naive Bayesian classifier and lower using summation hack. The relatively large range of result values can be explained by the overtraining of the model which can be controlled by the correct choice of environment parameters.

In later tests the reference points were generated sparsely, so the space has a small region with a relatively large number of reference points and outside this region there are only a few reference points.

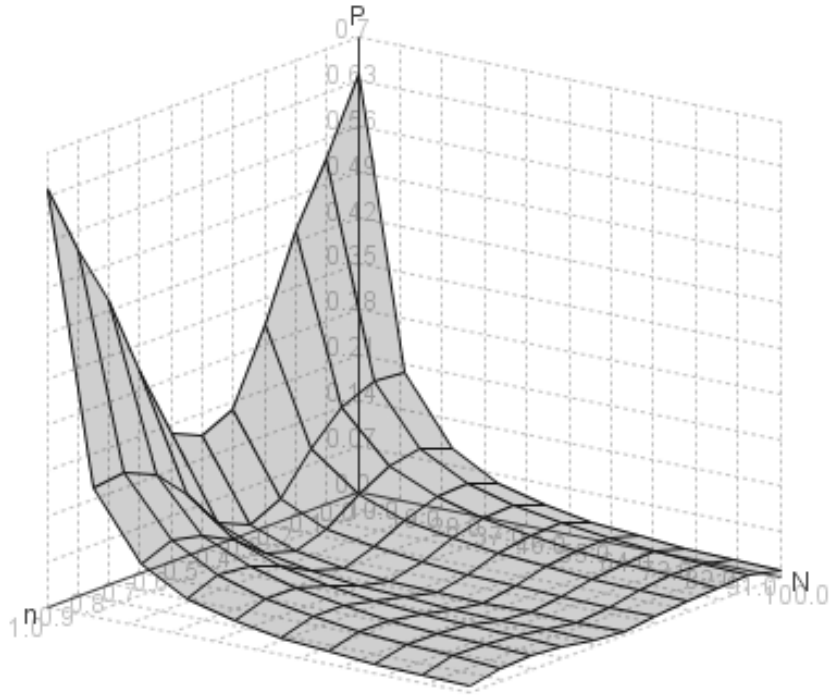


Figure 3. Mean value error function for different (n_1, N) values

In the case of this distribution, the accuracy of classifiers has changed. The teaching accuracy of Naive Bayesian classifier remained highly similar to other cases and the use of summation hack brought up the accuracy to the Naive Bayesian. In the testing phase the experience shows that in some cases the summation hack solution can improve the efficiency of classification and in many cases it exceeds the Naive Bayesian. It confirms the assumptions that the usage of summation hack in Bayesian classification can increase accuracy when the samples contain a great number of untrained attribute values.

The accuracy of classification depends on many parameters of the environment. One of the most important factors is the maximum attribute value parameter. Fig. 4 shows the accuracy functions for the following maximum attribute parameter values: 20 (NB20,SH20), 100 (NB100,SH100) and 500 (NB500,SH500). The notation NB is for Naive Bayesian algorithm and SH for the modified Bayesian algorithm. The accuracy of both algorithms has increased with increasing the size of the training set.

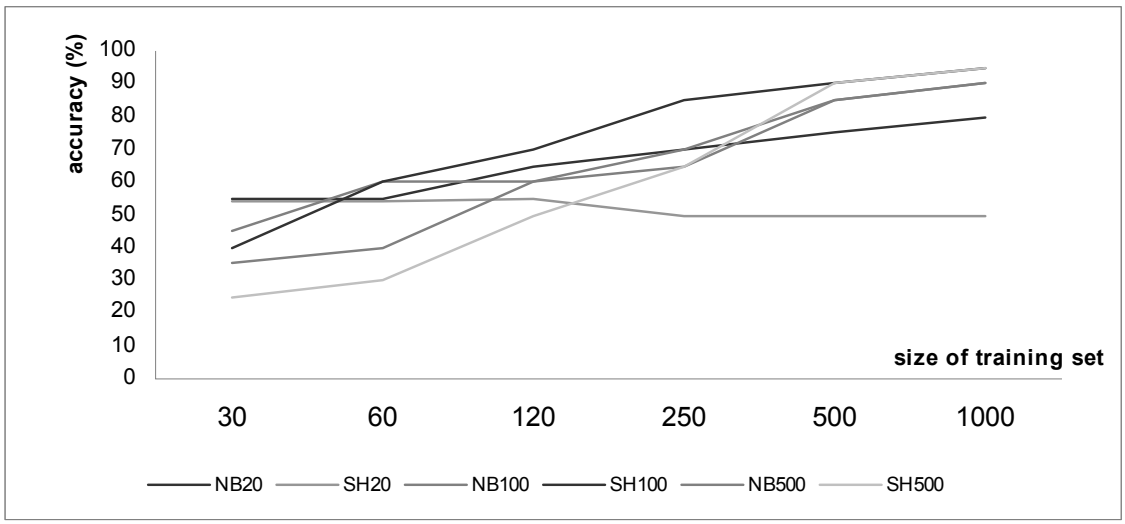


Figure 4. Relative accuracy of algorithms according to number of teaching samples

6. Conclusions

Summation hack is an alternative for the Naive Bayesian classifier with larger probability approximation errors. Taking a decision tree as a reference classifier, we have compared the Naive Bayesian classifier with the Bayesian classifier using summation hack. The test results show that both methods can yield the same accuracy as the decision tree method has in the case of large training sets.

REFERENCES

[1] THOMAS P. MINKA: *The 'summation hack' as an outlier model*, technical note, August 22, 2003

[2] HOLSTROM L, KOISTIEN P, LAAKSONEN J., OJA E: *Neural and Statistical Classifiers - Taxonomy and Two Case Studies*, IEEE Trans. On Neural Networks, Vol 8, No 1, 1997.

[3] KOVÁCS L., TERSTYÁNSZKI G.: *Improved Classification Algorithm for the Counter Propagation Network*, Proceedings of IJCNN 2000, Como, Italy.

[4] JOAQUIM P. MARQUES DE SÁ: *Applied Statistics Using SPSS*, Statistica, Matlab and R, Springer, 2007, pp. 223-268

[5] FUCHUN PENG, DALE SHUURMANS, SHAOJUN WANG: *Augmenting Naive Bayes Classifiers with Statistical Language Models*, Information Retrieval, 7, Kluwer Academic Publishers, 2004, Netherlands, pp. 314-345

[6] ROBERT P.W. DUNN: *Small sample size generalization*, 9th Scandinavian Conference of Image Analysis, June 6-9, 1995, Uppsala, Sweden

[7] BARABÁS P., KOVÁCS L.: *Usability of summation hack in Bayes Classification*, 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, November 6-8, 2008, Budapest, Hungary