Usability of Summation Hack in Bayes Classification

P. Barabás, L. Kovács

University of Miskolc, Department of Information Technology, Miskolc, Hungary, {barabas,kovacs}@iit.uni-miskolc.hu

Abstract

Bayesian classifiers provide relatively good performance compared with other more complex algorithms. Misclassification ratio is very low in case of trained samples, but outliers can cause lots of wrong classifications. Using of „summation hack" in Bayesian classification algorithm can reduce the rate of misclassifications in case of untrained samples, but at the same time the accuracy of classification can be decreased much or less in case of trained samples. The goal is to optimize the usage of summation hack in Bayesian classifiers generally after analysis and comparison of algorithms in a test environment.

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, measure of which can be separated in two parts: measure of accuracy in case of trained samples and measure of accuracy in case of untrained samples. Naive Bayesian classification is generally very accurate in first case since all testing samples are trained before and has no outliers; in second case the efficiency is worse because of outliers. Role of outliers[4] has to be examined in classification methods, naive Bayesian classification is reactive to outliers, and they can cause misclassification. Using of summation hack can make the classifier unaffected by outliers. The goal of our research is to analyze the generalization capability of Bayesian classification with using of summation hack. In second chapter a short summary about naive Bayesian classification is given. In the third chapter the concept of summation hack is introduced and examined. A test environment has been worked out for measuring the goodness of algorithms and for comparing them. The description of the environment is the topic of fourth chapter. Finally, the test results and conclusions have been summarized in the last chapter.

It is assumed that the objects to be classified are described with d-dimensional pattern vectors $\boldsymbol{x} = (x_1,\dots,x_n) \in \boldsymbol{R}^n$. Every pattern vector is associated with a class c_j, where the total number of class is *m*. Thus, a classifier can be regarded as a function

$$g(x) : R^n \to \{c_1, \dots, c_m\} \tag{1}$$

The optimal classification function is aimed at minimizing the misclassification risk.[1] The R risk value depends on the probability of the different classes and on the misclassification cost of the classes.

$$R(g(x)|x) = \sum\nolimits_{c_j} b\big(g(x) \to c_j\big) P(c_j|x) \tag{2}$$

where $P(c_j \mid \mathbf{x})$ denotes the conditional probability of c_j for the pattern vector **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 b cost function has usually the following simplified form:

$$b\big(c_i \to c_j\big) = \begin{cases} 0, if\ c_i = c_j \\ 1, if\ c_i \neq c_j \end{cases} \tag{3}$$

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

$$R(g(x)|x) = \sum\nolimits_{g(x) \neq c_j} P(c_j|x) \tag{4}$$

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

$$\sum\nolimits_{c_j} P\big(c_j|x\big) = 1 \tag{5}$$

thus if

$$P(g(x)|x) \to max \tag{6}$$

then the $R(g(x)|x)$ has a minimal value. The decision rule which minimizes the average risk is the Bayes' rule which assigns the **x** pattern vector to the class that has the greatest probability for **x**.[2]

2 Bayes classification

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:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{7}$$

If $\bar{A}$ is the complementary event of A, called „not A". The theorem can be stated as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \tag{8}$$

Let A_i is a partition of the event space. The general orm of theorem is given as:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}. \tag{9}$$

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

$$P(c_k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c_k)P(c_k)}{P(\boldsymbol{x})}. \tag{10}$$

Since $P(\boldsymbol{x})$ is the same for all *k* we have to maximize only the $P(\boldsymbol{x}|c_k)P(c_k)$ tag. The value $P(c_k)$ is given a priori or can be appreciated with relative frequencies from the samples. The $P(\boldsymbol{x}|c_k)$ is calculated with the following formula:

$$P(c_k|x_1,\dots,x_n) = \frac{P(c_k)P(x_1,\dots,x_n|c_k)}{P(x_1,\dots,x_n)}. \tag{11}$$

The numerator is equivalent to the joint probability $P(c_k,x_1,\dots,x_n)$. It can rewritten using the definition of conditional probabilities as follows

$$\begin{aligned} &P(c_k,x_1,\dots,x_n) \\ &\qquad = P(c_k)P(x_1,\dots,x_n|c_i) \\ &\qquad = P(c_k)P(x_1|c_k)P(x_2,\dots,x_n|c_k,x_1) \\ &\qquad = P(c_k)P(x_1|c_k)P(x_2|c_k,x_1)P(x_3,\dots,x_n|c_k,x_1,x_2) \\ &\qquad = P(c_k)P(x_1|c_k)P(x_2|c_k,x_1)P(x_3|c_k,x_1,x_2)P(x_4,\dots,x_n|c_k,x_1,x_2,x_3) \\ &\qquad = P(c_k)P(x_1|c_k)P(x_2|c_k,x_1)P(x_3|c_k,x_1,x_2)\dots\ P(x_n|c_k,x_1,x_2,x_3,\dots,x_{n-1}) \end{aligned} \tag{12}$$

According to the assumption of Naive Bayes classification the attributes in a given class are independent. This means that $P(x_i|c_k,x_j) = P(x_i|c_k)$ for every attributes where $j \neq i$. So the joint probability model can be expressed as

$$\begin{aligned} P(c_k, x, \dots, x) &= P(c_k)P(x_1|c_k)P(x_2|c_k)P(x_3|c_k)\dots P(x_n|c_k) \\ &= P(c_k)\prod_{i=1}^{n} P(x_i|c_k). \end{aligned} \tag{13}$$

Using above equation the probability of class c_k for an object featured by vector **x** is equal to

$$P(c_k|x_1, \dots, x_n) = \frac{P(c_k)\prod_{i=1}^{n} P(x_i|c_k)}{\prod_{i=1}^{n} P(x_i)}. \tag{14}$$

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

$$C^* = \underset{c_j \in C}{argmax}\{P(c_j|\boldsymbol{x})\} = \underset{c_j \in C}{argmax}\left\{P(c_j)\prod_{i=1}^{n} P(x_i|c_j)\right\}. \tag{15}$$

If a given class and feature never occur together in the training set than 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 huge number of samples the distortion of probabilities is marginal and the information loss through the zero tag can be eliminated successful. This technique called Laplace estimation.[3] 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 belong to class c_i.

3 Summation hack

Outliers in classification can indicate faulty data which causes misclassification. The classifier can be made unaffected by outliers with using summation hack. The „summation hack" is an ad-hoc replacement of a product by a sum in a probabilistic expression.[4] 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 values $x_1,x_2,\ldots,x_n$ and a class c. In Bayes classification where the vector values are conditionally independent:

$$p(\boldsymbol{x}|c) = \prod_{i=1}^{n} p(x_i|c) \tag{16}$$

In this case this 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:

$$p(\boldsymbol{x}|c) \approx \sum_{i=1}^{n} p(x_i|c) \tag{17}$$

In this case the resultant will be zero if all $p(x_i|c)$ values are equal to 0. Using (15) and (17) the computing of winner class is based upon the next equation:

$$C^* = \underset{c_j \in C}{argmax}\{p(c_j|\boldsymbol{x})\} \approx \underset{c_j \in C}{argmax}\left\{p(c_j)\sum_{i=1}^{n} p(x_i|c_j)\right\}. \tag{18}$$

Applying „summation hack" the error of classification can be reduced. In every equation above the frequency probability has been used which is an approximated value:

$$p(x) = \lim_{n\to\infty} \frac{n_x}{n_t} \tag{19}$$

where n_t is the total number of trials and n_x is the number of trials where event x occured. If number of trials approaches infinity, the relative frequency will coverage exactly to the probability. In many classification tasks; small number of samples are given[6], the number of trials are low, so we can compute with use of approximated values. We can write the probability as follows:

$$p(x_i = v|c_j) = \frac{N(x_i = v|c_j)}{N(x_i)} + \Delta_i = p_{ij} + \Delta_i \tag{20}$$

where Δ_i means the error of approximation. The classification error in case of summation hack can be computed:

$$\sum_{i=1}^{n}(p_{ij} + \Delta_i) - \sum_{i=1}^{n} p_{ij} = \sum_{i=1}^{n} \Delta_i. \tag{21}$$

Computing the resultant classification error for product of probabilities is more complex than in previous case:

$$\begin{aligned}\prod_{i=1}^{n}\left(p_{ij}+\Delta_i\right)-\prod_{i=1}^{n}p_{ij} &= (p_{1j}+\Delta_1)\,(p_{2j}+\Delta_2)\dots(p_{nj}+\Delta_n)-\left(p_{1j}p_{2j}\dots p_{nj}\right)\\ &= p_{1j}p_{2j}\dots p_{n-1j}\Delta_n+p_{1j}p_{2j}\dots p_{n-2j}\Delta_{n-1}p_{nj}+\cdots\\ &+\Delta_1p_{2j}p_{3j}\dots p_{nj}+p_{1j}p_{2j}\dots\ p_{n-2j}\Delta_{n-1}\Delta_n+\cdots\\ &+\Delta_1\Delta_2p_{3j}\dots p_{nj}+p_{1j}p_{2j}\dots\ p_{n-3j}\Delta_{n-2}\Delta_{n-1}\Delta_n+\cdots\\ &+\Delta_1\Delta_2\Delta_3p_{4j}\dots p_{nj}+\cdots+p_{1j}\Delta_2\dots\Delta_n+\Delta_1p_{2j}\Delta_3\dots\ \Delta_n\\ &+\Delta_1\Delta_2\dots\Delta_3.\end{aligned}\tag{22}$$

4 Test environment

One of the tasks of our research is to compare the naive Bayesian classification and with the Bayesian classification using summation hack. A generated decision tree as a reference classifier is used in test environment. The generated samples are completely stochastic. The generator algorithm has the following input parameters:

- number of classes,
- number of attributes,
- maximum attribute value,
- number of samples for generating decision tree,
- number of training samples,
- number of testing samples (untrained).

In first step reference points are generated randomly in Euclidean space not less as the number of classes. Every classes are assigned by different reference points. The number of attributes of points defines the dimensions of space; each attributes can take values between 0 and maximum axis value. The points are assigned to the class of the closest reference point. Euclidean distance is used to measure the distance between two points in the n-dimensional space. In the next step; the decision tree has been built by the ID3 algorithm from sample points.

Bayesian classifier is trained with randomly generated points which are classified by the decision tree; misclassified or unclassifiable points are dropped. Comparison process has two phases: measure of accuracy of teaching and measure of accuracy of testing. In first phase only trained samples are tested. All trained samples have to be classified by Bayesian classifiers which result is compared with the real class of the sample. In second phase new, untrained samples are classified by the Bayesian classifiers. Outputs of the testing environment are the accuracy values of the classifications. Fig. 1. shows the structure of testing environment.

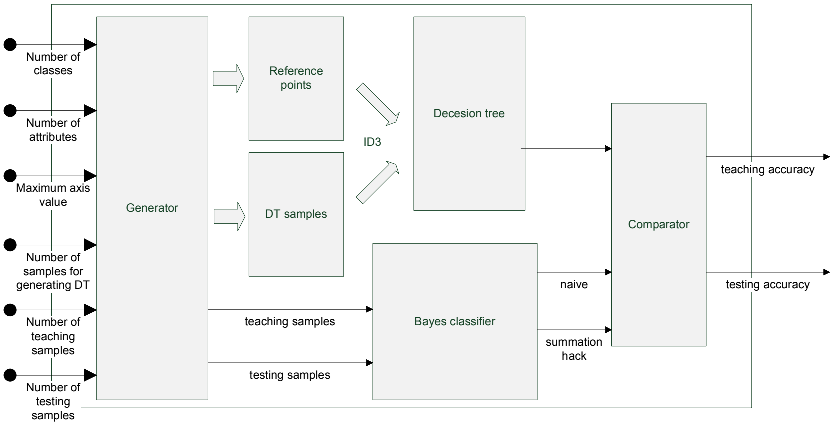

Figure 1. Structure of testing environment

5 Test results

According to our experiences, the efficiency and accuracy of ’summation hack’ depends on the environment. In first tests the reference points were generated with uniform distribution in the space. The winner was the naive Bayesian classifier in teaching and testing phase equally. The teaching accuracy took values from 80% to 100% depending from environment parameters. Using summation hack this accuracy decreased with about 10%. The testing accuracy is lower by far, it is between 40 and 70 percent in case of naive Bayesian classifier and lower using summation hack. The relative large interval of result values can be explained with the overtraining of the model which can be controlled by the correct choice of environment parameters.

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

In the case of this distribution, the accuracy of classifiers has been changed. The teaching accuracy of naive Bayesian classifier stayed high similar in other case and the using of summation hack brought up the accuracy to the naive Bayesian. In testing phase the experiences shows that in some cases the summation hack solution can improve the efficiency of classification and in a lot of cases it exceeds the naive Bayesian. It confirms the assumptions that the usage of summation hack in Bayesian classification can raise accuracy when the samples contain a lot of untrained attribute values.

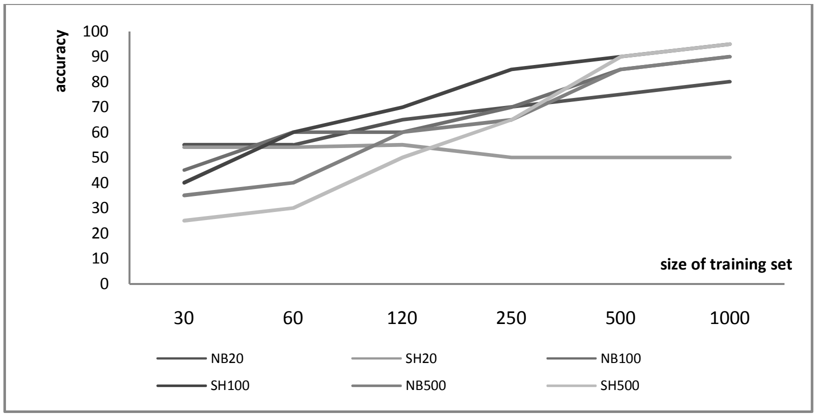

Figure 2. Relative accuracy of algorithms according to number of teaching samples

Accuracy of classification depends on many parameters of the environment. One of the most important factors is the maximum attribute value parameter. Fig. 2. 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 if for the modified Bayesian algorithm. The accuracy of both algorithms has increased with raising the size of the training set.

Conclusions

The 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] 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.

[2] L. Kovács, G. Terstyánszki: Improved Classification Algorithm for the Counter Propagation Network, Proceedings of IJCNN 2000, Como, Italy.

[3] Joaquim P. Marques de Sá: Applied Statistics Using SPSS, Statistica, Matlab and R, Springer, 2007, pp. 223-268

[4] Thomas P. Minka: The 'summation hack' as an outlier model, August 22, 2003

[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