# Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers

Mohammed J. Islam, Q. M. Jonathan Wu, Majid Ahmadi, Maher A. Sid-Ahmed
Department of Electrical and Computer Engineering
University of Windsor, Windsor, ON, Canada
islam11@uwindsor.ca, jwu@uwindsor.ca, ahmadi@uwindsor.ca, ahmed@uwindsor.ca

## Abstract

*Probability theory is the framework for making decision under uncertainty. In classification, Bayes' rule is used to calculate the probabilities of the classes and it is a big issue how to classify raw data rationally to minimize expected risk. Bayesian theory can roughly be boiled down to one principle: to see the future, one must look at the past. Naive Bayes classifier is one of the mostly used practical Bayesian learning methods. K-Nearest Neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of K-Nearest Neighbor category. The classifiers do not use any model to fit and only based on memory/ training data. In this paper, after reviewing Bayesian theory the Naive Bayes classifier and K-Nearest Neighbor classifier is implemented and applied to a dataset "Credit card approval" application. Eventually the performance of these two classifiers is observed on this application in terms of the correct classification and misclassification and how the performance of K-Nearest Neighbor classifier can be improved by varying the value of k.*

## 1. Introduction

Classification is an area of machine learning that takes raw data and classifies it as belonging to a particular class based on the required parameter set. Bayesian theory is basically woks as a framework for making decision under uncertainty- a probabilistic approach to inference [10, 6]. Bayes theorized that the probability of future events could be calculated by determining their earlier frequency. The appeal of the Bayesian technique is its deceptive simplicity. The predictions are based completely on data culled from reality- the more data obtained, the better it works. Another advantage is that Bayesian models are self-correcting, meaning that when data changes, so do the result. One highly practical Bayesian learning method is the Naive Bayes Classifier. It is based on the so called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods [8, 5]. In some domains its performance has been shown to be comparable to that of neural network and decision tree learning. In machine learning literature, nonparametric methods are called instance-based or memory based learning algorithms, since what they do is store the training instances in a lookup table and interpolate from these. The most basic instance-based method is the K-Nearest Neighbor algorithm.

## 2. Problem Statement

Classification scheme usually uses one of the following approaches: statistical and syntactic. Statistical pattern classification is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. Structural pattern recognition is based on the structural interrelationships of features [6]. A wide range of algorithms can be applied for pattern recognition, from very simple Bayesian classifiers to much more powerful neural networks.

Supervised classification is a technique in which we identify examples of information classes of interest within the dataset known as training data. Statistical characterization of each of the information class is obtained. Once a statistical characterization has been achieved for each information class, the test data is then classified by examining the best possible match against the training data and making an informed decision about the class it resembles most [5].

Unsupervised classification is a method which examines a large number of datasets and divides into a number of classes based on natural groupings present within the dataset. Unlike supervised classification, unsupervised classification does not require a priori labeling of training data. The basic phenomenon is that data within a given class

0-7695-3038-9/07 $25.00 © 2007 IEEE
DOI 10.1109/ICCIT.2007.148

IEEE computer society

should be as close as possible in the measurement space, whereas data in different classes should be comparatively well separated.

Although classification depends on the application and the information available from that problem, still research is going on to generalize these techniques and to conclude which technique is suitable for what kind of problems [9]. The main objective of this paper is to review, implement Bayesian theory and investigating the performance of two widely used Naive Bayes and K- Nearest Neighbor (KNN) classifier based on an application ”Credit card approval”, and also to show how classification rate improves in KNN by varying the value of k. After reviewing literature in section 3, the proposed approach is implemented and applied in section 4, in section 5 the experimental results and performance evaluation are shown and finally conclusions are discussed in section 6.

## 3. Literature Review

To meet the objectives of the paper an exhaustive literatures survey was done.

### 3.1. Bayesian Theory

Bayesian learning algorithm is the most practical learning approach for most learning problems and is based on evaluating explicit probabilities for hypotheses. Bayes learning classifier is extremely competitive with other learning algorithms and in many cases outperforms them. Bayesian learning algorithms are extremely important in machine learning since they provide unique perspective for understanding many learning algorithms that do not explicitly manipulate probabilities [2].

Bayes theorem states that:

$$P(h|D) = \frac{P(D|h)P(h))}{P(D)} \tag{1}$$

where-

$P(h)$ : Prior probability of hypothesis $h$- Prior

$P(D)$ : Prior probability of training data $D$-Evidence

$P(D|h)$: Probability of $D$ given $h$- Likelihood

$P(h|D)$: Probability of $h$ given $D$- Posterior probability

In the general case, we have $K$ mutually exclusive and exhaustive classes; $h_i, i = 1, ......K$; $P(D|h_i)$ is the probability of seeing $D$ as the input when it is known to belong to class $h_i$. The posterior probability of class $h_i$ can be calculated as-

$$P(h_i|D) = \frac{P(D|h_i)P(h_i)}{P(D)} = \frac{P(D|h_i)P(h_i)}{\sum_{i=1}^{K} P(D|h_i)P(h_i)} \tag{2}$$

In order to choose the best hypotheses from amongst the set of generated hypothesis the maximally probable hypothesis $h_{MAP}$ is selected and is known as maximum a posteriori (MAP) hypothesis and if we assume that $P(h)$ is same for all the hypothesis then the maximally probable hypothesis reduces to maximum likelihood hypothesis.

### 3.2. Naive Bayes Classifier

The Naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given target value [7]. In other words, the assumption is that given the target value of the instance, the probability of observing the conjunction $a_1, a_2......a_n$ is just the product of the probabilities for the individual attributes:

$$P(a_1, a_2, ......a_n|v_j) = argmax_{v_j \in V} \prod_i P(a_i|v_j) \tag{3}$$

$$v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \tag{4}$$

So, basically the Naive Bayes Classifier ignores the possible dependencies, namely, correlations, among the inputs and reduces a multivariate problem to a group of univariate problems. It is noticed that in a Naive Bayes classifier the number of distinct $P(a_i|v_j)$ terms that must be estimated from the training data is just the number of distinct attribute values times the the number of distinct target values- a much smaller number than if we were estimate the $P(a_1, a_2.....a_n|v_j)$ terms as needed for Bayesian theory.

### 3.3. K-Nearest Neighbor Classifier (KNN)

Among the various methods of supervised statistical pattern recognition, the Nearest Neighbor rule achieves consistently high performance, without a priori assumptions about the distributions from which the training examples are drawn. It involves a training set of both positive and negative cases. A new sample is classified by calculating the distance to the nearest training case; the sign of that point then determines the classification of the sample [9, 3]. The KNN classifier extends this idea by taking the k nearest points and assigning the sign of the majority. It is common to select k small and odd to break ties (typically 1, 3 or 5). Larger k values help reduce the effects of noisy points within the training data set, and the choice of k is often performed through cross-validation. In parametric methods, whether for density estimation or classification, we assume a model valid for over the whole input space. In classification, when we assume a normal density, we assume that all examples of the class are drawn from this same density. But practically this assumption does not always hold and we may incur a large error if it does not. If we cannot make such assumptions and cannot come up with a parametric model, one possibility is to use semi parametric mixture model. In nonparametric estimation, all we assume is that similar inputs have similar outputs. This is a reasonable assumption: The world is smooth and functions, whether they are densities, discriminants or regression functions, changes slowly. Similar instance means similar things. Therefore, our algorithm is composed of finding the similar past instances from the training set using a suitable distance measure and interpolating from them to find the right output [4, 11].

## 4. Implementation, Training and Testing

In this section the classifier is implemented, trained using the training set and then tested using the other set for testing. The data set is collected for this application from the source [1]. The brief description of the data file is given below:

Raw datafile: crx-raw.data

Training datafile: crx.train

Testing datafile: crx.test

*Story behind the datafile*

These files concern credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset is interesting because there is a good mix of attributes- continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values.

The credit-approval data set contains information about 671 applicants for credit and whether they were approved or rejected. Each application is described by 15 attributes and classified as approved ("+") or rejected ("-").

The raw data base contains some missing values, and it also contains numerical attributes. The cases with missing values is removed, and the numerical attributes is discarded to simplify the task, leaving 9 categorical attributes. Each of the 9 attributes is a one letter symbol that is a shorthand for a more meaningful English language description.

Number of Instances: 671

Number of Attributes: 9 + class attribute

Attribute Information: Attributes are shown in Table 1.

| Attributes | Information |
|---|---|
| A1 | b, a |
| A2 | u, y |
| A3 | g, p, h |
| A4 | i, k, c, g, q, d, a, m, x, w, j, r, e, b |
| A5 | h, v, f, d, b, j, z, m, o |
| A6 | t, f |
| A7 | t, f |
| A8 | t, f |
| A9 | g, p, s |

**Table 1. Attributes for Credit Application**

The Naive Bayes probability table and KNN classifier for each attributes is created using the training set that are shown in results section. Taking information from these tables, the posterior probability of each instance from testing set is estimated for each class 'accept' or 'reject' and finally the maximum probability predicts the class for this testing examples. The correct classification and misclassification is calculated that reflects the performance of the classifier and is shown in section 5.

## 5 Results and Performance

Using Naive Bayes classifier at first the tables are constructed from the attributes A1 to A9 that are shown in Table 2 through Table 10. The test set is classified based on the probabilities estimated in Table 2 through 10. Each example is picked up from the test set and then its class is predicted. The predicted class is compared with target value that is given in test set. If they mismatch, this example becomes error. The same process is repeated for all 201 examples and last of all the percentage of correct classification and misclassification is calculated that is shown in Table 11. Intuitively the trained classifier is tested on training set by which it is trained and surprisingly the information found is also not error free. It has some percentage of misclassification which is close to the value that is found is testing set. The information on training set found is shown in Table 12.

In K- Nearest Neighbor, the value of k is given like k=1, 3, 5, 11, 51, and 101. The distance metric is also given which is called Euclidean distance. Initially all 470 training examples are stored in a look up training set. The dimension of the distance vector is 470x1. Based on the value of k, the k numbers of smallest distance training examples are picked up and then calculate their corresponding accept or reject. Since k is always odd, so there is no chance to tie. For example, for k=1, the minimum distance training examples output will be the predicted value for this testing examples. For k=3, 3 minimum distance training examples output are considered and calculate the number of accept and reject. The bigger one will be the predicted value for this testing example. The same procedure is repeated for k=5, 11, 51 and 101. Last of all for each k, the (%) or error is calculated that is shown in Table 13.

From the Table 13, it is clear that for this particular problem at K=5, we get the minimum error which is 9.45(%). The tables from 2 to 13 are shown below:

| A1 | Accept | Reject |
|---|---|---|
| b | 0.6837 | 0.6980 |
| a | 0.3163 | 0.3020 |

**Table 2. P(A1|accept) and P(A1|reject)**

| A2 | Accept | Reject |
|---|---|---|
| u | 0.8236 | 0.6902 |
| y | 0.1674 | 0.3098 |

**Table 3. P(A2|accept) and P(A2|reject)**

| A3 | Accept | Reject |
|---|---|---|
| g | 0.8326 | 0.6902 |
| p | 0.1674 | 0.3098 |
| h | 0.0047 | 0.0000 |

**Table 4. P(A3|accept) and P(A3|reject)**

| A4 | Accept | Reject |
|---|---|---|
| i | 0.0419 | 0.0863 |
| k | 0.0419 | 0.0902 |
| c | 0.2326 | 0.2314 |
| g | 0.0233 | 0.1255 |
| q | 0.1628 | 0.0824 |
| d | 0.0233 | 0.0667 |
| a | 0.0605 | 0.0863 |
| m | 0.0558 | 0.0549 |
| x | 0.0884 | 0.0157 |
| w | 0.1302 | 0.0902 |
| j | 0.0093 | 0.0118 |
| r | 0.0047 | 0.0000 |
| e | 0.0372 | 0.0314 |
| b | 0.0884 | 0.0275 |

**Table 5. P(A4|accept) and P(A4|reject)**

| A5 | Accept | Reject |
|---|---|---|
| h | 0.2977 | 0.1412 |
| v | 0.5814 | 0.6314 |
| f | 0.0233 | 0.1294 |
| d | 0.0093 | 0.0118 |
| b | 0.0698 | 0.0706 |
| j | 0.0047 | 0.0078 |
| z | 0.0093 | 0.0039 |
| m | 0.0 | 0.0039 |
| o | 0.0047 | 0.0000 |

**Table 6. P(A5|accept) and P(A5|reject)**

| A6 | Accept | Reject |
|---|---|---|
| t | 0.9535 | 0.2275 |
| f | 0.0465 | 0.7725 |

**Table 7. P(A6|accept) and P(A6|reject)**

| A7 | Accept | Reject |
|---|---|---|
| t | 0.7116 | 0.2549 |
| f | 0.2884 | 0.7451 |

**Table 8. P(A7|accept) and P(A7|reject)**

| A8 | Accept | Reject |
|---|---|---|
| t | 0.4279 | 0.4431 |
| f | 0.5721 | 0.5569 |

**Table 9. P(A8|accept) and P(A8|reject)**

| A9 | Accept | Reject |
|---|---|---|
| g | 0.9581 | 0.9020 |
| p | 0.0047 | 0.0039 |
| s | 0.0372 | 0.0941 |

**Table 10. P(A9|accept) and P(A9|reject)**

| Classification | Number | Percentage(%)) |
|---|---|---|
| Correct | 176 | 87.57 |
| Incorrect | 25 | 12.43 |

**Table 11. Classification table using Naive Bayes on testing set. Total examples: 201**

| Classification | Number | Percentage(%) |
|---|---|---|
| Correct | 410 | 87.24 |
| Incorrect | 60 | 12.76 |

**Table 12. Classification table using Naive Bayes on training set.Total examples: 470**

| K | Correct(%) | Error(%) |
|---|---|---|
| 1 | 80.10 | 19.90 |
| 3 | 85.57 | 14.43 |
| **5** | **90.55** | **9.45** |
| 11 | 86.57 | 13.43 |
| 51 | 86.07 | 13.93 |
| 101 | 85.57 | 14.43 |

**Table 13. (%) of error for different K using K-Nearest Neighbor**

## 6. Conclusions

The main objective of this paper is to review the Bayesian theory and explore two simple machine learning techniques: a Naive Bayes classifier and K- Nearest Neighbor Classifier and investigate the performance of these two widely used classifiers. To fulfill these objectives, rigorous methodological steps are followed to implement the machine learning algorithms and application of these techniques to a dataset credit card approval. All the tables like probability, classification and other tables that were necessary to estimate the probability and to predict the classification were shown in section 5. The process of calculating these tables and from these tables how to estimate the probability table, basically reflects the details process in case of Bayes classifier, Naive Bayes classifier and K-Nearest Neighbor classifier. From these tables, we can see the performance of each classifier. Still, some conclusions can be made from this experiment:

1. The result of Bayesian inference depends strongly on prior probabilities, which must be available in order to apply the method directly. Since the Bayes theorem provides a principled way to calculate the posterior probability of each hypothesis given the training data, we can use it as the basis for straightforward learning algorithm that calculates the probability for each hypothesis then outputs the most probable.

2. The Naive Bayes classifier is applied to the credit card approval testing data set and found 12.43 (%) error of misclassification.

3. Instance-based methods are sometimes referred to as ”Lazy” learning methods, because they delay the process until a new instance must be classified. A key advantage of this kind of delayed or lazy learning is that instead of estimating the target function once for the entire space, these methods can estimate it locally and differently for each new instance to be classified.

4. In K-Nearest Neighbor method, the selection of k values is tricky and application dependent. To simplify our problem it is always fixed to odd number so that no tie can happen. In our experiment we tried to classify the credit approval testing data set for different values of k and for k=5 we got the maximum correct classification rate. At k=5, the (%) of error classification is 9.45 (%) that is shown in Table 13.

## References

[1] *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening*.

[2] E. Alpaydin. *An Introduction to Machine Learning*. The MIT press, Cambridge, Massachusetts, London, England, 2004.

[3] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, 13:21–27, 1967.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley Interscience, 2nd ed.

[5] S. Eyheramendy, D. Lewis, and D. Madigan. On the naive bayes model for text categorization. *Proceedings Artificial Intelligence Statistics*, 2003.

[6] D. Lewis. *Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval*. ATT lab Research, NJ, USA.

[7] T. Mitchell. *Machine Learning*. McGraw-Hill.

[8] I. Rish. An empirical study of the naive bayes classifier. *Proceedings of IJCAI-01*, 2001.

[9] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, 1995.

[10] K. Weise and W. Woger. Comparison of two measurement results using the bayesian theory of measurement uncertainty. *Measurement Science and Technology*, 5:879–882, 1994.

[11] Q. J. Wu. *Class Notes- Machine Learning and Computer Vision*. University of Windsor, Windsor, ON, Canada, 2007.