Volume 2, No. 2, Mar-Apr 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Analysis of K-NN and Naïve Bayes Classifiers for Spam Filtering Applications

M. Aqeel Iqbal*
Department of Computer Engineering
College of Electrical and Mechanical Engineering
National University of Sciences and Technology (NUST), Pakistan
maqeeliqbal@hotmail.com

Arslan Shoukat
Department of Computer Engineering
College of Electrical and Mechanical Engineering
National University of Sciences and Technology (NUST), Pakistan
arslan_asp@gmail.com

Shoaib A. Khan
Department of Computer Engineering
College of Electrical and Mechanical Engineering
National University of Sciences and Technology (NUST), Pakistan
kshoab@yahoo.com

Saleem Iqbal
Department of Computer Engineering
College of Electrical and Mechanical Engineering
National University of Sciences and Technology (NUST), Pakistan
siqbal00pk@yahoo.com

Abstract – Pattern classification is one of the most important and leading aspects of modern image processing systems. By training a classifier on a set of data, the unseen samples can be categorized as much accurate as training has been done. There are many different classifiers having varying accuracies, design complexities and performance. With different design strategies these classifiers may have different characteristics. In this paper a performance analysis of K-NN and Naïve Bayes classifiers have been presented for the classification of spam emails. Different design aspects of both classifiers have also been presented in terms of computational complexity and classification accuracy against their performance.

Keywords – K-NN Classifier, Naïve Bays Classifier, Spam Filtering, Performance Analysis of K-NN and Naïve Bays Classifiers

I. INTRODUCTION

There have been three main categories of pattern recognition techniques or trends:

A. Supervised Learning

The supervised learning techniques are also known as classification or regression techniques. In these techniques each element/object of the data set in fact comes with a pre-assigned category/class label. In other words we can say that there is a teacher who is guiding for the true answers or categories. The fundamental task being run is to train a given classifier to perform the classification and hence do the labeling job by using the information that has been provided by the teacher. A procedure or algorithm which itself tries to leverage the teacher‘s answer to perform the transformation to generalize the problem, and hence in this way obtains his learnt knowledge, is known as learning algorithm.

Mostly it has been observed that this kind of learning procedure cannot be fully described in a human understandable format, like most prominent one’s including Artificial Neural Networks based classifiers etc. In such type of learning cases, the data set and the teacher‘s labeling both are provided to the machine so that to run the procedure of learning over the given data set [1], [2]. In large number of new kind of classifiers or classification systems, it has been tried to investigate and minimized the errors and propose a kind of emerging solution to compensate such kind of procedural errors. There have been a large number of classification and clustering techniques which has been adopted as the combinational approaches.

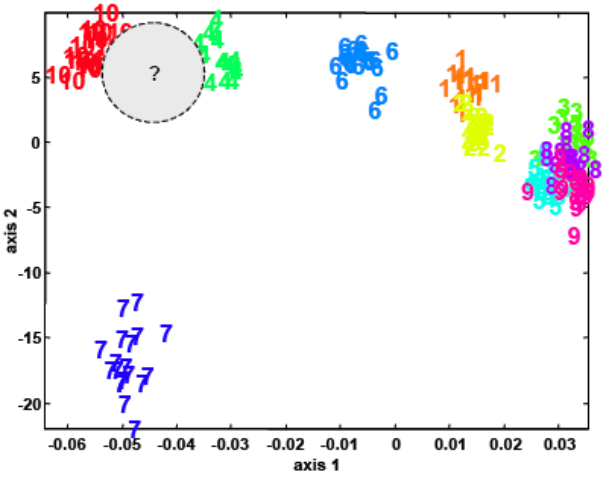

Figure-1: Pattern Classification Concept

B. Semi-supervised (Reinforcement) Learning

While the in the case of supervised learning technique, the algorithm (supervised learning) tries to learn from the truly available labels which are basically the answers of learner questions to the teacher, in semi-supervised learning the learner in fact conversely uses teacher just for the sake of approval or disapproval against the given data information. Hence in the case of semi-supervised learning actually it should be noted that there is no available teacher for supervision. The process of semi-supervised learning the procedure first starts with fully random manner and when it reaches the final state, it looks to the condition whether he

wined or loosed [3]. For example in the case of famous chess game, that there may be no supervisor at all, but system is gradually trained to play better by trail-and-error process [3], [4]. After all this it is to look at the end of the game to find you wined or loosed.

C. Unsupervised Learning

In this case of unsupervised learning, the system forms the clusters or natural grouping of the input patterns. There is no teacher who provides the pre-computed labels or classifications.

II. NEAREST NEIGHBOR CLASSIFIERS

A. Basic Idea:

A new data sample arrived after training is a bird with certain characteristics. The task is to classify it either it is a duck or not. Rule for the classification of the given data sample would be quite simple. The classifier needs to check, does it walks more likely to that of a duck and also is it true that it quacks more likely similar to that of a duck. If both of above statements are true then most probably it should be a duck.

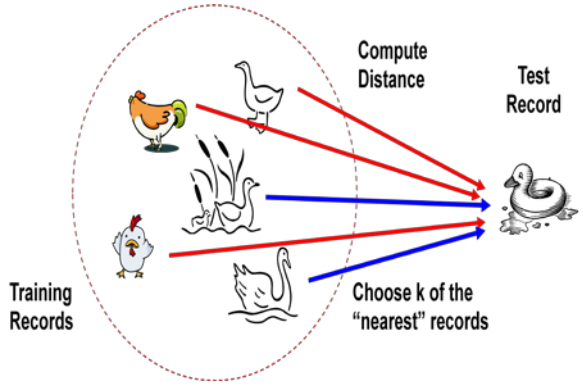

Figure-2: Nearest Neighbor Concept

K-NN classifier may be described by the following parametric aspects.

Required: The Distance Function on Instances

Model: Labeled Training Data (a1; c1),..., (aN; cN)

Objective: To Classify New Instance "A" As Follows:

- Let (aj1 ; cj1) ,....., (ajK; cjK) be the K training instances whose attributes are closest to a.

- Label a with the class label that occurs most frequently among cj1 ,....., cjK.

The K-NN classifier is based on non-parametric density estimation techniques

- Let us assume we seek to estimate the density function P(x) from a dataset of examples

- P(x) can be approximated by the expression

$$P(x) \cong \frac{k}{NV}$$ where { V is the volume surrounding x; N is the total number of examples; k is the number of examples inside V

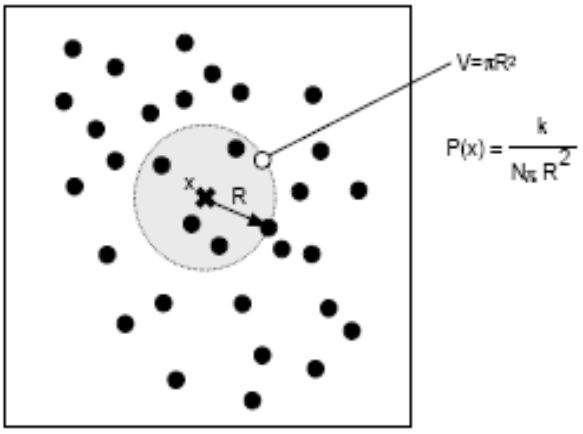

Figure-3: K-NN Classifier Implementation

The volume V is determined by the D-dim distance Rk D(x) between x and its k nearest neighbor

$$P(x) \cong \frac{k}{NV} = \frac{k}{N \cdot c_D \cdot R_k^D(x)}$$

Where cD is the volume of the unit sphere in D dimensions

- The unconditional density is, again, estimated with

$$P(x|\omega_i) = \frac{k_i}{N_i V}$$

- And the priors can be estimated by

$$P(\omega_i) = \frac{N_i}{N}$$

- The posterior probability then becomes

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)} = \frac{\frac{k_i}{N_i V}\cdot\frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

- Yielding discriminant functions

$$g_i(x) = \frac{k_i}{k}$$

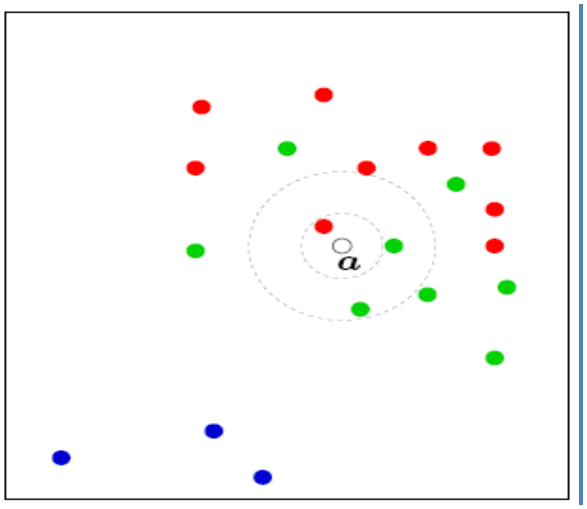

Figure-4: Nearest Neighbor Classification

This process is generally called the k Nearest Neighbor (K-NN) classifier. In terms of the K-nearest neighbor, the

training point X which is most nearest in terms of the distance, to the arrived new testing point is taken as the nearest one. Labeling for the training instances on the basis of given data set should be like (Class Labels = red, green, blue). Nearest neighbor is red classify a as red 2 out of 3 nearest neighbors are green classify a as green.

III. NAÏVE BAYES CLASSIFIER

Naive Bayes classifiers have been working very well in many real life complex situations irrespective of their naive design and apparently over-simplified assumptions. In last decade the analysis of the Bayesian classification problem has shown that there have been certain theoretical reasons for the apparently unreasonable low efficacy of naive Bayes classifiers [5],[6]. But even still, a large number of comprehensive comparisons presented in this era, with other classification methods have shown that Bayes classification is outperformed by more current approaches, such as boosted trees or random forests [7], [8]. An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification [9], [10], [11]. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix [12].

A. Naïve Bayesian Classifier Case Study

[a] Training Dataset

Consider the following table containing the details of some data set for the calculation of the probability of certain aspect.

[b] *Class:*

- C1:buys_computer='yes'
- C2:buys_computer='no'

[c] Data Sample:

X = (age <=30, Income = medium, Student = yes Credit_rating = Fair)

Table-1: Extracted Features of Data Set

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

P(Ci):

P(buys_computer = "yes") = 9/14 = 0.643

P(buys_computer = "no") = 5/14= 0.357

Compute P(X|Ci) for Each Class:

P (age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

P (age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

P (income = "medium" | buys_computer = "yes") = 4/9 = 0.444

P (income = "medium" | buys_computer = "no") = 2/5 = 0.4

P (student = "yes" | buys_computer = "yes) = 6/9 = 0.667

P (student = "yes" | buys_computer = "no") = 1/5 = 0.2

P (credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

P (credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

B. Classification Example

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

P (X|Ci) :

P (X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P (X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P (X|Ci) * P(Ci) :

P (X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P (X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

IV. PROPOSED IMPLEMENTATION

The followings are the logical steps being followed for the implementation of the proposed design. The proposed design has been implemented and evaluated for the filtering of the spam emails. Data set for the demonstration of classifiers have been taken from UCI Repository of machine learning databases:

http://www.ics.uci.edu/~mlearn/MLRepository.html.

A. Proposed Design Flow

[a] Data=:Load data

[b] N=: Number of instances
[c] M=: Number of features
[d] Labels=: Class type
[e] Features= Features of instances
[f] Divide Data into 10 folds
[g] For each fold
[i] Testing data =: Examples in the fold
[ii] Training data =: Remaining 9 folds
[iii] For each testing data point x
a. Measure distance to every training data point
b. Find the k closest points to the test data point
c. Identify the most common class among the k closest points
d. Predict the class identified in previous step.
e. Calculate Accuracy for KNN
[iv] End
[v] d =: M
[vi] μ =: Mean of training data
[vii] ∑ =: Covariance of the training data

[viii] $$P(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]$$

[ix] P(0/x) =: P(x)*P(0)
[x] P(1/x) =: P(x)*P(1)
[xi] If (P(0/x) >P(1/x)
f. Predict class with label 0
Else Predict class with label 1
g. Calculate Accuracy for Naïve Bayes
[h] Calculate mean accuracy for KNN
[i] Calculate mean accuracy for Naïve Bayes

V. QUANTITATIVE RESULTS ANALYSIS

The proposed design has been implemented using Matlab-7. The following results have been obtained from the implementation.

A. Results of K-NN Classifier

Following is a set of results obtained by choosing different values of K. Where K = Number of Nearest Neighbors

Value of K	3	5	7	9	11
Accuracy	0.742	0.740	0.729	0.722	0.715

B. Graphical Representation:

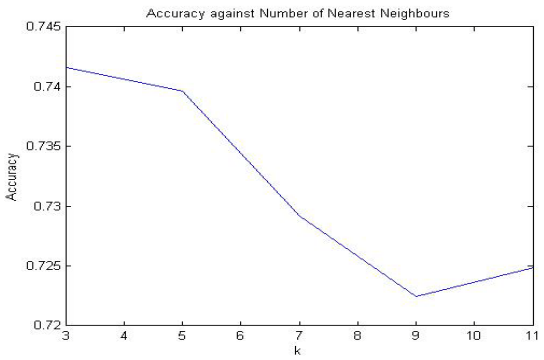

Figure: 5

C. Results of Naïve Bayes Classifier

Accuracy: 0.745

[a] Confusion Matrix:

2722(Non-Spam/True Predict)66 (Non-Spam/False Predict)
1107 (Spam/False Predict) 706 (Spam/True Prediction)

From the results obtained for the classification of spam filtering of emails, it has been observed that the Naïve Bayes classifier would be much better to use for this purpose as compared to the K-NN classifier.

VI. CONCLUSION

Choice of classifiers for pattern categorization is highly dependant on the feature space being available for data objects as well as the accuracy and performance demands of the applications. The experimental work done for the classification of a data set by using both classifiers have shown that no doubt the K-NN classifier is very simple to demonstrate and implement, but as a whole the overall performance of the Naïve Bayes classifier is better then it.

VII. REFERENCES

[1]. L. I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, New York: Wiley, 2005.
[2]. H. Parvin, H. Alizadeh, B. Minaei-Bidgoli and M. Analoui, CC HR: Combination of Classifiers using Heuristic Retraining, In Proc. of the Int. Conf. on Networked Computing and advanced Information Management by IEEE CS, (NCM 2008), Korea, Sep. 2008.
[3]. H. Alizadeh, M. Mohammadi and B. Minaei-Bidgoli, Neural Network Ensembles using Clustering Ensemble and Genetic Algorithm, In Proc. of the Int. Conf. on Convergence and hybrid Information Technology by IEEE CS, (ICCIT08), Nov. 11-13, 2008, Busan, Korea.
[4]. H. Parvin, H. Alizadeh and B. Minaei-Bidgoli, A New Method for Constructing Classifier Ensembles, International Journal of Digital Content: Technology and its Application, JDCTA, ISSN: 1975-9339, 2009 (in press).
[5]. H. Parvin, H. Alizadeh and B. Minaei-Bidgoli, Using Clustering for Generating Diversity in Classifier Ensemble, International Journal of Digital Content: Technology and its Application, JDCTA, ISSN: 1975-9339, 2009 (in press).
[6]. S. Bermejo, J. Cabestany, Adaptive soft k- nearest neighbor classifiers. Pattern Recognition, Vol. 33, pp. 1999-2005, 2000.
[7]. K. ITQON, Shunichi and I. Satoru, Improving Performance of k-Nearest Neighbor Classifier by Test Features, Springer Transactions of the Institute of Electronics, Information and Communication Engineers 2001.
[8]. R. Bekkerman, A. McCallum, and G. Huang. Automatic categorization of email into folders: Bechmark experiments on enron and sri corpora. Technical report, Department of Computer Science. University of Massachusetts, Amherst., 2005.
[9]. P. Bermejo, J. A. G´amez, and J. M. Puerta. Attribute construction for email foldering by using wrappered forward greedy search. In 9th International Conference

on Enterprise Information Systems, pages 247- 252, 2007.

[10]. S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning, 10:57– 78, 1993.

[11]. T. Joachims. Learning to classify text using support vector machines. Kluwer Academic Publishers, 2002.

[12]. B. Klimt and Y. Yang. The ENRON corpus: a new dataset for email classification research. In 15th European Conference on Machine Learning, pages 217–226, 2004.