International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 2 (2017), pp. 303-311

http://www.ripublication.com

Nearest Neighbor Classifiers: A Review

R. Raja Kumar[1*] P. Viswanath[2] and C.Shobha Bindu[3]

[1]Rajeev Gandhi College of Engg. and Technology, Nandyal.
[2]IIITS,Chittoor.
[3]JNTUA College of Engineering, Anantapuramu.

**Corresponding Author*

Abstract

Nearest Neighbor classifier (NNC) is a simple classifier which is popular in the fields of Data mining, pattern recognition etc. It is easy to understand this classifier. This paper presents the issues , some of the prominent methods of Nearest Neighbor classification method. It gives an overview of the Nearest Neighbor Classifiers.

Keywords: Pattern Recognition, Data Mining, Prototypes, nearest Neighbors Feature Reduction

1. INTRODUCTION

Nearest Neighbor classification is one of the popularly known classification technique in Pattern Recognition [1]. This classification technique is simple to use and easy to understand. Given an unknown pattern, it finds the nearest neighbor and assigns the class label of this nearest neighbor to the unknown pattern [2]. A simple extension of NNC is to consider the k nearest neighbors and the class label is given based on majority voting decision [1].

The rest of this paper is organized as follows. In Section 2, the Literature Survey is presented. In section 3, working of nearest neighbor method is discussed. In Section 4, the problem with the Nearest Neighbor Classification technique is presented. In section 5, some of the methods to cope up with the problem were discussed. In Section 6 we reviewed the methods. In section 7 conclusion and direction for future scope is presented.

2. LITERATURE SURVEY

Considerable numbers of papers were published on how to reduce the training set. The early method in this context was condensed nearest neighbor (CNN) proposed by Hart [4]. In this method, a pattern is selected and put in the condensed set. Later each pattern in the training set is classified. If a pattern is misclassified, it is included in the condensed set. After one pass through the training set, iteration is carried out. This iterative process is carried out till there are no misclassified samples present in the training set. Now, the condensed set is used instead of training set to classify the unlabeled patterns. The condensed set has two properties. First, it is a subset of original training set. Second, it ensures consistency property over the training set. Consistency means the reduced set gives 100 percent accuracy over the training set from which it is derived. CNN results in two disadvantages 1) retention of unnecessary samples and 2) occasional retention of internal rather than boundary patterns [5]. In 1976, Ivan Tomek proposed two modifications to CNN which remove the above disadvantages [5]. Swonger proposed an iterative condensation algorithm (ICA) [6] which allows additions and deletions from the condensed set. Gates proposed reduced nearest neighbor rule (RNN) [7] in 1972. This method is contrary to CNN approach. Here the training set is considered as condensed set because by default a training set is a condensed set over itself. Instead of growing the condensed set as in CNN, now from this set unnecessary patterns are removed without degrading classification accuracy. Proximity graphs are used by Sanchez, Pla, Ferri for prototype selection [8]. Devi and Murthy have used stochastic techniques for prototype selection on optical character recognition data [9]. Kuncheva and Jain used soft computing techniques like tabu search, simulated annealing and genetic algorithms [10], [11].

In 1994, Dasarathy proposed an approach to find an optimal subset called as Minimal Consistent Set(MCS) of training set [13]. This approach is based on Nearest unlike Neighbor Subset. MCS sees that for every sample, the sufficient condition for its correct classification. In 2002 devi and murthy [12] proposed a prototype selection method called Modified condensed nearest neighbor (MCNN). In this algorithm,prototype set is built in an incremental manner. R-Tree or KD-Tree can be used for this purpose. The training set patterns information is stored in the multidimensional tree using these trees [3]. But these trees have the drawback that the design time is increased and does not address the outliers or noise problem.

3. NEAREST NEIGHBOR CLASSIFICATION

The Nearest Neighbor Classifier is one of the simple classifier used in Machine Learning, Pattern Recognition and Data Mining and its related fields. We try to understand the bird's view of the NNC using Figure 1. There are two classes present in the Figure 1. One class is represented as circles and the diamonds represent other. Let '*' be the newly arrived pattern for which class label has to find out. The 1-NN rule or NN rule infers the label its nearest neighbor. The '*' is close to circle class than diamond class. Hence it infers circle class to '*' which means saying '*' belongs to circle class.

A simple extension of NNC is instead of considering only one nearest neighbor,

We can increase the number of Nearest Neighbors for a given test sample. Generalizing this is called k-Nearest Neighbor Classification where k = 1, 2...n a positive integer value. For example, consider k=3 then the class label infers to diamond class by majority voting. We illustrate the NNC rule with the sample dataset shown in table 1. It has two attributes X and Y along with the class labels information present in the last column. Two class labels 1 and 2 are present. The working of 3-NNC is presented in Figure 2.

Table 1: Example Dataset

No.	**X**	**Y**	**Class**
X_1	1	1	**1**
X_2	1	2	**1**
X_3	2	1	**1**
X_4	2	2	**1**
X_5	5	1	**2**
X_6	5	2	**2**
X_7	6	1	**2**
X_8	6	2	**2**

Table 2: Euclidean distances between Q And patterns of table 1

No.	**Q**
X_1	2.12
X_2	1.75
X_3	1.75
X_4	0.7
X_5	2.91
X_6	2.54
X_7	3.8
X_8	3.53

Let a query pattern Q= (2.5, 2.5) has to be classified. Then NNC finds the distances from this query pattern 'Q' to all the patterns in the dataset 1. We used Euclidean distance for finding Nearest Neighbor. The distances from the pattern 'Q' to the patterns in the dataset shown in table 1 are shown in the table 2. The Nearest Neighbor for 'Q' is pattern X_4. The class label of X_4 is yes. Hence 'Q' is inferred the class label Yes and it is said to be 'Q' belongs to class 'Yes'. The rule used here is known as NNC or 1-NNC since only one nearest neighbor is taking into picture to decide class label.

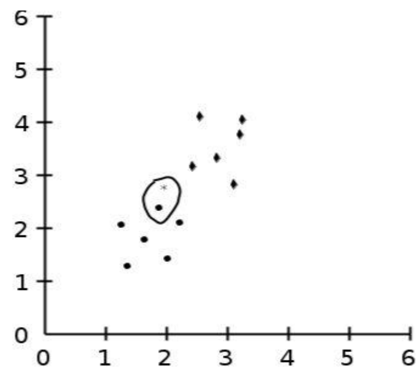

Figure 1: Working of 1-NNC

4. PROBLEM WITH NNC

Even though it's simple NNC has some severe limitations. The following are the major draw backs of NNC.

(1). It has to store each and every pattern in the training set *i.e,* entire training set should be stored. This seems good if the training set size is small but not preferable if the training set size is large [12].

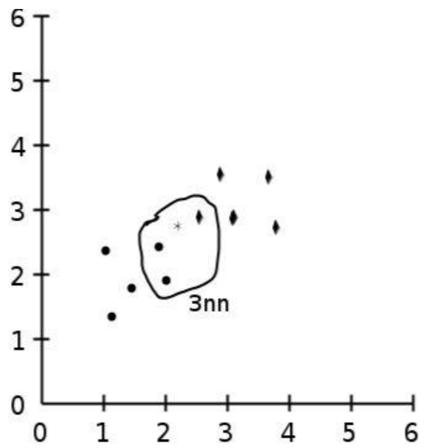

Figure 2: Working of 3NNC

(2) NNC suffers with the curse of dimensionality problem. The feature set is important for a classifier as it discriminates the patterns. A good number of features produce good classifier. If the number of features increases it affects the performance of the classifier. The dimensionality problem is well discussed in [15], [16].

(3) The outlier effect. An outlier is an unwanted object in the training set. The classifiers which are basically depend on distance measures are affected by outliers. The effect of outliers on sample set size was discussed in [17].

Hence NNC attracts the eyes of researches.

5. METHODS

There fore the researches always tries to rectifying the problems of NNC. From its invention(late 1950's) to the state of art. The space complexity and dimensionality effect are the two major problems than the outlier effect. One has to reduce the training set size and number of features to increase the performance of NNC. Some of the methods used for this purpose are outlined below.

Condensed Nearest Neighbor (CNN) method:

Condensed Nearest Neighbor [4] begins with the pattern selected from the training set which forms the initial condensed set. Then each pattern in the training set is classified using the condensed set. If a pattern in the training set is misclassified, it is included in the condensed set. After one pass through the training set iteration is carried out. This iterative process is carried out till there are no misclassified patterns. The final condensed set has two properties [4].

It is a subset of original training set.

It ensures 100% accuracy over the training set which is the source for deriving the condensed set.

Modified Condensed Nearest Neighbor method (MCNN): In this algorithm [12], prototype set is built in an incremental manner. A typical pattern is selected from each class and it is kept in the set. This is considered as initial prototype set to begin with. The training set patterns are classified by using this prototype set [12]. From the misclassified samples in the training set, a representative pattern is computed from each class and is added to the prototype set. Now again the training set is classified with the prototype set. This process is an iterative process and repeated till there are no misclassified samples present in the training set. Now the final typical pattern set is the desired prototype set. It is a subset of original training set and ensures consistency over the training set [12]. In a class of patterns, the pattern which is near to centroid may be selected as typical pattern [12].

Other Class Nearest Neighbor (OCNN) method:

This method is based on the intuitive notion of retaining patterns that are near to decision boundary [18]. It is obvious that the patterns which are near decision boundary plays very important role in the classification of a process [19]. The boundary patterns are computed by using Other Class Nearest Neighbor (OCNN) method [20]. It computes the nearest neighbors that are not from its own class. This process is repeated for each pattern in the training set. OCNN is well discussed in [18], [19], and [20].

Principal Component Analysis:

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are linear transformation methods [21]. They are related to each other. In PCA, we are interested to find the directions of the component that maximize the variance in the given dataset, where as in LDA, we are additionally interested to find the directions that maximize the discrimination power between the different classes present in the dataset. In other words, Using PCA, we are projecting the entire set of data (Class labels are not taken into account) onto a different subspace [21]. But in LDA we are trying to find the suitable subspace that separates the classes present in the dataset. Frankly speaking, in PCA we find the axes where the most of the data is spread over where as in LDA we additionally consider the class labels too. In a single work PCA treats the whole dataset belongs to one class but LDA considers the different labels of the dataset [18], [19], and [21].

Sequential Forward Selection:

Sequential Forward selection (SFS) is a feature selection algorithm which selects the subset of features based on objective function J (.). It aims to find the best subset of features from the given original set of features [20]. It starts with an empty set and finds the feature which best optimizes the objective function [20]. Generally the objective function is accuracy. Then that feature includes into the set. This process repeats till all the features are explored or we reach a satisfaction level. SFS process is well discussed in [20].

6. COMPARISONS

In this section we compare the methods. The comparison is based on the experimentation and the average of all the results obtained in [18], [19], [20], and [14]. The algorithms are implemented on DELL OPTIPLEX 745 n model having Intel core 2 DUO 2.2 Ghz processor with 1GB DDR 2 RAM capacity. The comparisons of the methods are presented in Table 3. We considered the following Attributes for reviewing the methods.

Attribute #1: Prototype Selection= this refers whether the method able to reduce the patterns/samples in the training set. It indicates Prototype Reduction.

Attribute #2: Feature Reduction= this refers whether the method able to reduce the features in the training set. It indicates Dimensionality Reduction.

Attribute #3: Both 1 & 2 = This refers whether the method able to reduce both patterns as well as features in the training set.

Attribute #4: Reduction Rate= this refers to the ratio of cardinality of the set after applying the method to before applying the method. If it is less than 1 then we say that it is possible to achieve the reduction rate.

Attribute #5: Accuracy: The accuracy is the percentage of samples that are correctly classified from the test set. It refers the performance of the classifier.

Attribute #6: Applicable to Large Datasets: This refers whether the method feasible over large data sets or not.

Method A+ B means first method A was applied followed by method B. For example, CNN+LDA means first the prototype method CNN was applied. After that Feature reduction method LDA was applied on the resultant reduced set.

If the aim is Prototype reduction then one of CNN, MCNN, and OCNN can be used. If the aim is Feature Reduction then LDA or SFS can be used. If the aim is both prototype reduction and Feature Reduction then combination of Prototype reduction with the Feature Reduction method or vice versa can be applied.

Table 3: Comparisons of Methods

Method	**Attribute #1**	**Attribute #2**	**Attribute #3**	**Attribute #4**	**Attribute #5**	**Attribute #6**
CNN	Yes	No	No	Possible	Good	Yes
MCNN	Yes	No	No	Possible	Good	Yes
OCNN	Yes	No	No	Possible	Good	Yes
LDA	No	Yes	No	Possible	Good	Yes
SFS	No	Yes	No	Possible	Good	No
CNN+LDA	Yes	Yes	Yes	Possible	Average	Yes
LDA+CNN	Yes	Yes	Yes	Possible	Good	Yes
OCNN+LDA	Yes	Yes	Yes	Possible	Average	Yes
LDA+OCNN	Yes	Yes	Yes	Possible	Good	Yes
CNN+SFS	Yes	Yes	Yes	Possible	Good	No
SFS+CNN	Yes	Yes	Yes	Possible	Good	No
OCNN+SFS	Yes	Yes	Yes	Possible	Good	No
SFS+OCNN	Yes	Yes	Yes	Possible	Good	No
No Method	No	No	No	Not Possible	Good	Yes

7. CONCLUSIONS AND FUTURE SCOPE

In this Paper review of some of the prominent methods which can be used for reducing the computational burdens of Nearest Neighbor Classifiers. The future scope of this paper is, the interested researches further can work to overcome the limitations of the above said methods. The extensions of the methods were discussed in [18], [19], [20], and [14].

REFERENCES

[1] Babu, V. S., Viswanath, P. (2007, December). Weighted k-nearest leader classifier for large data sets. In International Conference on Pattern Recognition and Machine Intelligence (pp.17-24). Springer Berlin Heidelberg.

[2] Cover, T., Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[3] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509-517.

[4] P.E. Hart. (1967). The condensed nearest neighbor rule, IEEE Trans. Inform. Theory IT-14(3), 515-516.

[5] Tomek, I. (1976). Two modifications of CNN. IEEE Trans. Systems, Man and Cybernetics, 6, 769-772

[6] Swonger, C. W. (1972). Sample set condensation for a condensed nearest neighbor decision rule for pattern recognition,511-519.

[7] G.W. Gates(1972). The reduced nearest neighbour rule. IEEE Trans. Inform. Theory IT-18 (3),431-433.

[8] Sanchez J. S. Pla, F. and Ferri, F. J. (1997). Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognition Letters, 18(6), 507-513.

[9] Devi, V. S., Murty, M. N. (2000). Handwritten digit recognition using soft computing tools.In Soft Computing for Image Processing (pp. 507-524). Physica-Verlag HD.

[10] Kuncheva, L. I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letters, 16(8), 809-814.

[11] Kuncheva, L. I., and Jain, L. C. (1999). Nearest neighbor classifier: simultaneous editing and feature selection. Pattern recognition letters, 20(11), 1149-1156.

[12] Devi, V. S., and Murty, M. N. (2002). An incremental prototype set building technique. Pattern Recognition, 35(2), 505-513.

[13] Dasarathy, B. V. (1994). Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Transactions on Systems, Man, and Cybernetics, 24(3), 511-517.

[14] Raja kumar, R., Viswanath, P., Shoba Bindu, C.(2014). A New Prototype Selection Method for Nearest Neighbor Classification, Proc. of Int. Conf. on Advances in Communication, Network, and Computing, CNC, 2014 ACEEE, pp 1-8.

[15] http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

[16] Keogh, Eamonn, and Abdullah Mueen. ”Curse of dimensionality.” Encyclopedia of Machine Learning. Springer US, 2011. 257-258

[17] Selst, Mark Van, and Pierre Jolicoeur. ”A solution to the effect of sample size on outlier elimination.” The quarterly journal of experimental psychology 47.3 (1994): 631-650.

[18] Kumar, R. R., Viswanath, P., and Bindu, C. S. (2016). An Approach to Reduce the Computational Burden of Nearest Neighbor Classifier. Procedia Computer Science, 85, 588-597

[19] Kumar, R. R., Viswanath, P., and Bindu, C. S. (2016). Nearest Neighbor Classifiers: Reducing the Computational Demands. 6 th IEEE International Advanced Computing Conference (IACC 2016).

[20] Kumar, R. R., Viswanath, P., and Bindu, C. S. (2016). A Cascaded Method to Reduce the Computational Burden of Nearest Neighbor Classifier. 1 st International Conference on Computational Burden of Nearest Neighbor Classifier (ICCII-2016).

[21] Sebastian Raschka (2013):http://www.sebastianraschka.com/.