International Journal of Engineering Sciences Paradigms and Researches (IJESPR)
(Vol. 27, Issue 01) and (Publishing Month: December 2015)
(An Indexed, Referred and Impact Factor Journal)
ISSN (Online): 2319-6564
www.ijesonline.com

Enhanced Rank Based Similarity Search on Social Networking

Chandrapal Singh Arya[1] and Pradeep Kumar Sharma[2]

**[1]M.Tech. Scholar, Department of Computer Science and Engineering,
Sobhasaria Engineering College, Gokulpura, Sikar, Rajasthan (India)**
aryanchandrapal@gmail.com

**[2]Associate Professor, Department of Computer Science and Engineering,
Sobhasaria Engineering College, Gokulpura, Sikar, Rajasthan (India)**

Publishing Date: December 31, 2015

Abstract

The advancement in the technology and increased usage of the social networking websites leads to huge amount of data available for processing. This data exhibits the property of Big Data and needs to be processed efficiently. The Big data processing is necessary for the extraction of information from big data. The numerous numbers of researches have been done in the field of data mining but this work focus on the classification techniques. The study of various techniques exhibits that the RCT algorithm is most efficient for the low dimension dataset but its performance gets degraded with the increase in the dimensionality of the dataset. The RCT algorithm defines a data structure for K-NN classification technique. The MRCT modifies the RCT by including a step for the dimension reduction. The performance enhancement of the MRCT over the RCT signifies the effectiveness of the work.

Keywords: *RCT, MRCT, K-NN, Big Data, SASH.*

1. Introduction

The size of the internet data is increasing day by day. This can be easily understood by following few facts: one fact is the study that how fast a media covers the 50 million people [1]. The entrainment devices like TV cover the world in about 20 years while the Facebook covers the world in 2 years. This shows the increasing demands of social media. Now a day, games like angry bird covers the 50 million people in just 35 days. These figures can give you an idea for the enhanced social media data. Another fact the size of internet data, the data indexed by Google is 1 million in the 1998, to 1 billion in 2000 and 1 trillion in 2008. This can also show the rapid expansion of the internet data. Nowadays, a number of social media application and coming in the market and covering the market a month. This is a big business opportunity that can be done only by the mining this BIG data. Moreover, the variety of data is large as different people pay bill, play games, and chat etc i.e. a large number of activity. Each activity provides different type of data that can be used for different purpose. This huge volume data with immense variety is differed from the existing data that why named as Big Data The Big Data is nothing but a data, which is generated from the heterogeneous and autonomous sources, also available in extreme amount, gets updated in fractions of second. It is huge in size and keeps on changing time to time from different phases. It is free from the influence, guidance or control of anyone. It is too much complex in nature, thus hard to handle[2].

2. K-NN Classifier

The K-nearest neighbor algorithm was given by Cover and Hart in 1968. There were many criteria's to find the nearest neighbor but the Euclidean distance method is most suitable due to its efficiency and simplicity [3]. The concept of nearest neighbor is simple and motivated from the general life. This algorithm considers the similarity between the neighbor and the present element to decide the class of the element. The K-NN is a non parameter technique with enhanced results with the optimized value of neighbors i.e. k. The class assigned to the test sample in the K-NN is the most occurred class in the neighbors of the test sample. If two or more classes cover same number of neighbor then the average is assigned to the test sample. If the value of k approaches to

infinity then this classifier performs same as Bayes decision rule [4]. The K-NN classifier, determine the class of any particular instance on the basis of the class known of related instance. The assigned class directly depends upon the similarity with the known class instance [4].

The *k*-nearest neighbor (*k*-NN) technique, due to its interpretable nature, is a simple and very intuitively appealing method to address classification problems. However, choosing an appropriate distance function for *k*-NN can be challenging and an inferior choice can make the classifier highly vulnerable to noise in the data. The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques. While in case of problem having only two classes, the value of k should be odd. It allows to select the class of the element appropriately otherwise same number of samples belongs to each class will create a problem. The concept of Euclidean distance is used in continuous parameters for other the concept of hamming distance can be used.

The KNN algorithm is shown in the following form:
Input: D, the set of k training objects, and test object z= (x', y').

Process: Compute d(x', x), the distance between z and every object, (x, y) ∈ D. Select $D_z \subseteq D$, the set of k closet training objects to z.

Output: $y' = \text{argmax}_v \sum_{(xi, yi) \subseteq Dz} I(v= y_i)$

- v is a class label
- y_i is the class label for the i^{th} nearest neighbors
- I (.) is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

In this system, KNN algorithm is used the suitable result by mixing the Euclidean distance among the various kinds of distance metric. The Euclidean distance is as shown in below:

$$d_{ij} = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2 + \cdots + \left(x_{ip} - x_{jp}\right)^2} \quad (1)$$

Where

d_{ij} = the distance between the training objects and test object

x_i = input data for test object

x_j = data for training objects stored in the database

In KNN algorithm, there are several advantages and disadvantages.

3. Spatial Approximation Sample Hierarchy (SASH)

The K-NN classifier disadvantages can be covered by developing architecture for the K-NN search. The SASH architecture[5], a multilevel hierarchy is developed for the K-NN classifier. A SASH is an acronym for "Spatial Approximation Sample Hierarchy". It is a structure with more than one level formed by using the recursion over a subset $\bar{O} \subset O$ of the sample space of objects say O and remaining elements are treated as the neighbor to be connected. The elements of $\bar{O}$ are connected to its approximate neighbors that are outside the $\bar{O}$. When any query occurs then the neighbor of test sample is firstly found in the $\bar{O}$ then approximate neighbor is found outside the $\bar{O}$. The SASH uses pair distance to measure the similarity. The SASH is build up in bottom to top manner i.e. by adding one point at each level especially at parent level. Assume, we have a leaf node say v at the level 1, now the probability that the particular node v is present at level j is $1/2^j$. In the SASH each node is connected to one node at its parent level which can be selected as its neighbor. In the SASH each node as a constant connections i.e. degree by taking care that each node can act a parent at most c= 4p children. If any node tries to connect with more than c children then only c closest children to parent are selected all other are rejected.

When, the algorithm starts to search for the similarity then firstly a limit for the maximum number of neighbor selection is assigned to each level say lj for level j. The search always begin with the root node then expands to its children nodes; but only closest nodes to the test sample are selecting by taking care of upper limit.

In the K-NN search, the $l_j = MAX\left\{l^{1-\frac{j}{\log_2 n}}, \frac{1}{2}pc\right.$.

While, the total number of nodes visited can be limited by

$$\frac{l^{1+\frac{1}{\log_2 n}}}{l^{\frac{1}{\log_2 n}}} + \frac{pc^2}{2}\log_2 n = \tilde{O}(l + \log n) \quad (2)$$

The SASH is of the heuristic nature due to its structure and the execution time limit is implemented for the fast results on the real world data. The SASH produces the fast and accurate results even for real world datasets.

4. Rank Cover Tree (RCT)

The Rank Cover Tree[6] modifies the SASH algorithm to improve the performance. However, it maintains the execution speed and also compromise the accuracy due to restricted number of neighbors at each level. The RCT includes the random leveling concept. It defines a partial RCT for the leaf nodes moreover, one ancestor exist for a node at one level.

This algorithm also begins its searching from the root node and expands the tree by determining the cover tree of the node. It means the selected node covering set is responsible for the expansion of the tree. In this if a node is available in the result then its parent will also be available. For a particular test sample, a particular node with its parent is analyzed for the nearest neighbor.

One parameter say p is used for the coverage at each node; contain only real numbers. It shows the number of level affected by particular node to determine the accuracy and the execution time. While the RCT is build up by insertion of nodes at the random level on the basis of the neighbor of the nodes. If a node is neighbor of another then it can be inserted as its child node. The RCT is efficient only if the nearest neighbor of a node is the parent of that particular node. The whole concept is also shown using the algorithm and flow diagram in section 4.1 and figure 1 respectively.

4.1 RCT Algorithm

1. Determine the level of each Item say u.
2. Select lowest level index say s.
3. Connect s to its parents i.e. s-1 level.
4. For level=s-2 to 0
5. If current element not exist in level then current element=k nearest neighbor.
6. End for
7. End

This algorithm can also be understood by following figure 1.

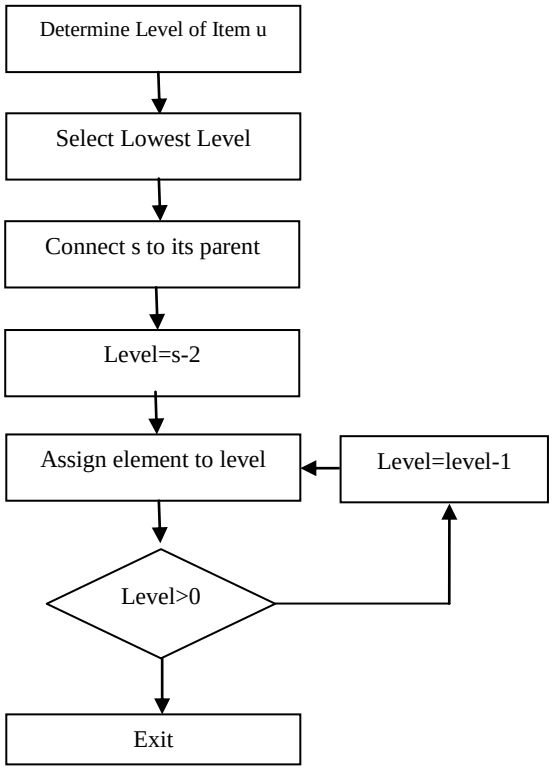

Figure 1: RCT Algorithm

The modification of the RCT algorithm is described in next section.

5. Proposed Work (MRCT)

The RCT algorithm is most efficient for the low dimension dataset but its performance gets degraded with the increase in the dimensionality of the dataset. The MRCT modifies the RCT by including a step for the dimension reduction. The dimension reduction can be achieved by selecting the principal components of the dataset. The principal components are found by determining the scatter matrix of the attribute data. For the process of dimension reduction Suppose, input attributes has n_c classes for the a_n attributes of the dataset. The reduction process is described from equation (3) to (6).

$$\mu_{i=\frac{1}{n_i}\sum a_i} \quad (3)$$

The equation (3) calculates the mean of the particular class i.e. the elements of ith class are

considered for the calculation of μ_i. Here a_i is the attribute of class i. The number of elements in each class can be different. The mean is calculated by averaging the elements belonging that particular class. The scatter matrix S is

$$S = \frac{1}{a_n} \sum_{i=1}^{n_c} S_i \qquad (4)$$

Where S_i can be calculated by using the equation (4).

$$S_i = \sum_{i=1}^{n_i} (a_{n_i} - \mu_i)(a_{n_i} - \mu_i)^t \qquad (5)$$

The equation (5) calculates the scatter matrix for each class. The scatter matrix for each class is calculated by finding the deviation of each data point from the mean of the belonging class. The scatter matrix basically squares the variation of the data point from the mean. The square is calculated by a mathematical property i.e. $\|a\|^2 = a.a^t$. Thus the scatter matrix is calculated for each class while the overall scatter matrix is calculated by adding scatter matrix of each class.

The main attributes can be selected by evaluated the Eigen vector of the S. The Eigen vector of the S is used to select the principal attribute on the basis of equation (4).

$$PA = Min(E_v(S)) \qquad (6)$$

The component having the minimum Eigen vector component of the scatter matrix are selected. In other words, the Attributes satisfying the criteria of equation (4) are marked as principal attributes. Where Eigen Vector is mathematical property evaluated as $|A - \lambda I|X = 0$; where I is the identity matrix, A is the matrix whose Eigen vector has to be evaluated. The λ is the Eigen values and the X is the Eigen vector. The selected attributes after the dimension reduction step are goes through the RCT method i.e. searching from the root node and expanding the tree by determining the cover tree of the node(or expanding by node covering set) to classify the data. The process can be easily understood by the following algorithm:

5.1 MRCT Algorithm

1. Input the dataset.
2. Calculate mean of each class instance by using equation (1).
3. Calculate the scatter matrix by using equation (2).
4. Select the principal attributes by using equation (4).
5. Determine the level of each Item say u.
6. Select lowest level index say s.
7. Connect s to its parents i.e. s1 level.
8. For level=s-2 to 0
9. If current element not exist in level then current element=k nearest neighbor.
10. End for
11. End

This process can also be explained by using flowchart of figure 2.

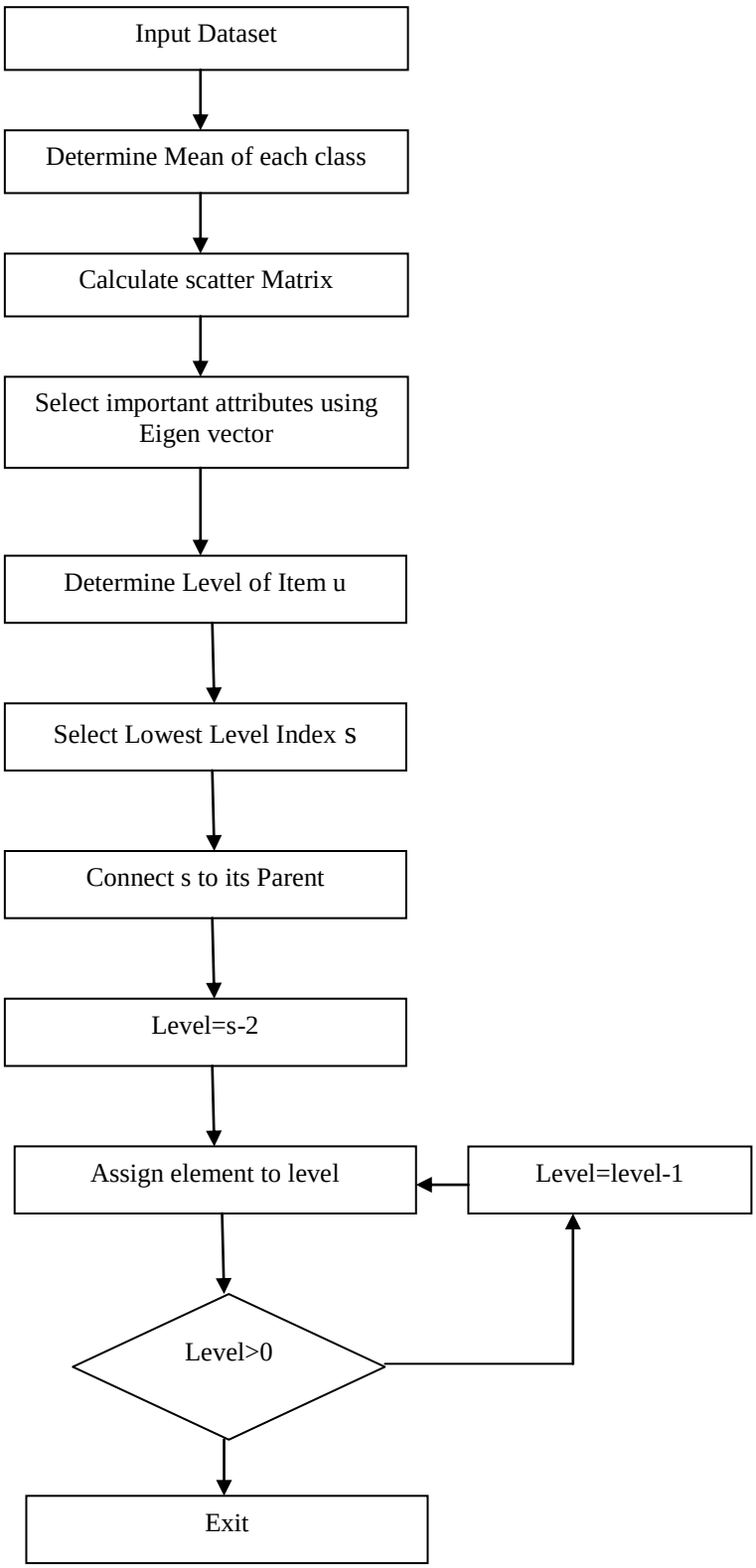

Figure 2: MRCT Algorithm

The figure 2 describes the MRCT algorithm flow. The next section discusses the implementation of the described algorithm.

6. Data Set Description

The dataset used to analyze the proposed algorithm over WEKA is the dataset of buzz prediction on twitter. This dataset is downloaded from the UCI repository [7]. This dataset contains 38393 instances in a file having format csv i.e. comma separated values.

Table 1: The Detail of Other Datasets

Dataset name	Twitter	Cloud	Airport	CPU
Number of Attributes	77	7	7	7
Number of Instance	38393	108	3379	209
Data Type	Nominal+ Numeric	Nominal+ Numeric	Nominal+ Numeric	Nominal+ Numeric
Dataset Area	Social	Agriculture	Airport	Computer
Reference	UCI Repository	Statlib	UCI Repository	WEKA+ UCI repository

The table 1 shows the detail of different dataset used for analysing the performance of the MRCT. The table describe the features as well as the reference of the dataset used. The results on these datasets are described in the next chapter.

7. Results and Discussion

The simulation results show the comparison of the MRCT with RCT on different datasets. The results can be visualized that the error rate has been reduced in dataset by using MRCT algorithm. Check the performance of MRCT algorithm and how this algorithm works when the dimension reduction technique applied on datasets. In evaluating the results of RCT and MRCT algorithms, different statistics or performance parameters are used. In this subsection, we discuss the essential performance evaluation parameters required to produce the efficient results, Correlation Coefficient, Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error defined in [8]. The results are obtained by using the WEKA tool. These results can be compared graphically as shown in figure 3 to 7.

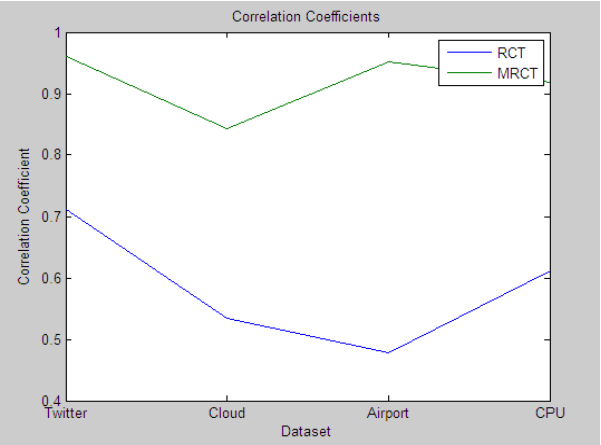

Figure 3: Correlation Coefficient Comparison on Different Dataset

The figure 3 shows the comparison of the correlation coefficients on different datasets. It can be visualized that the correlation has been improved in all dataset by the algorithm MRCT as compared to RCT. This is due to the removal of useless attributes.

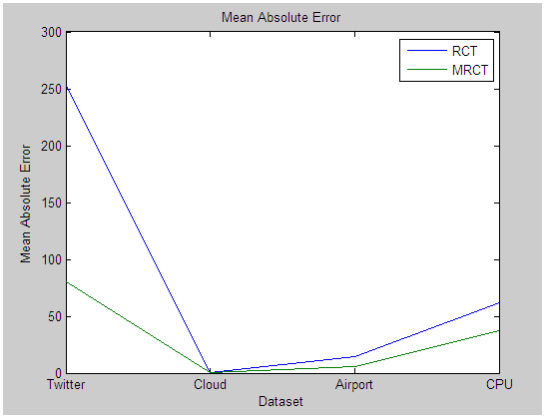

Figure 4: Mean Absolute Error Comparison on Different Dataset

The figure 4 shows the comparison of the Mean absolute error on different datasets. It can be visualized that the error has been reduced in all dataset by the algorithm MRCT as compared to RCT. The reduction is more when the number of attributes and instance are large in the dataset.

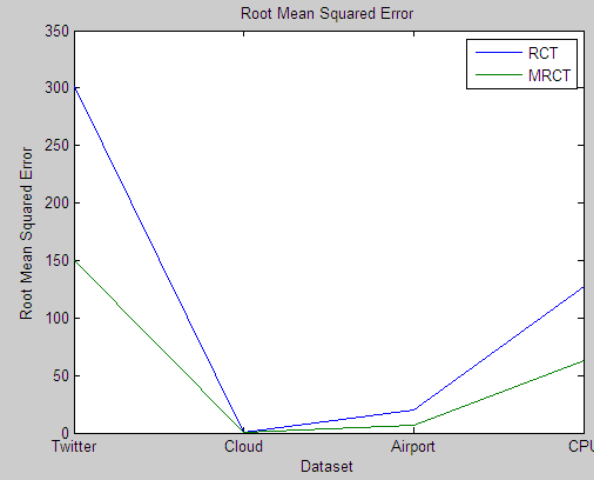

Figure 5: Root Mean Squared Error Comparison on Different Dataset

The figure 5 shows the comparison of the root mean squared error on different datasets. It can be visualized that the error has been reduced in all dataset by the algorithm MRCT as compared to RCT. The reduction is more when the number of attributes and instance are large in the dataset.

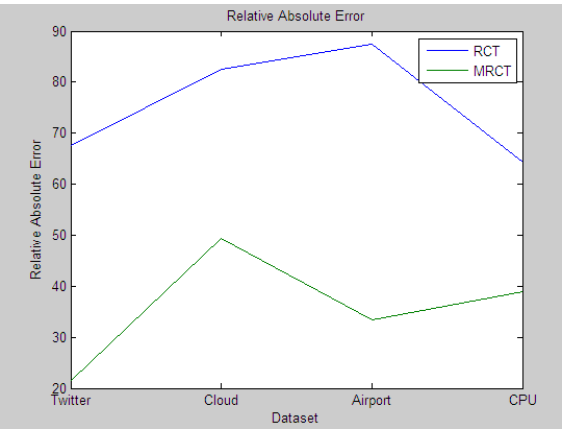

Figure 6: Relative Absolute Error Comparison on Different Dataset

The figure 6 shows the comparison of the relative absolute error on different datasets. It can be visualized that the error has been reduced in all dataset by the algorithm MRCT as compared to RCT. The reduction is more when the number of attributes and instance are large in the dataset.

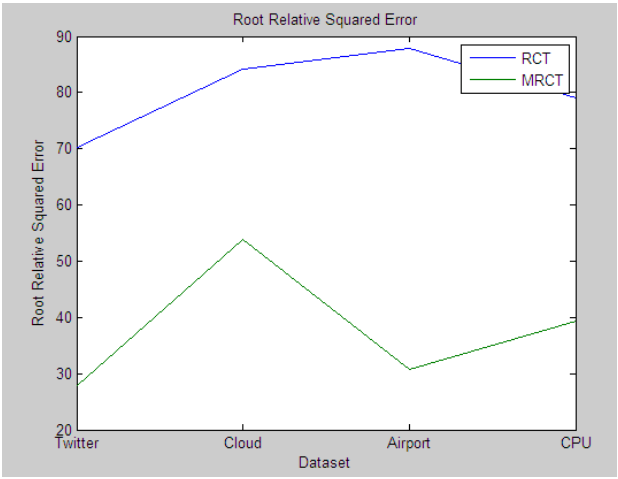

Figure 7: Mean Absolute Error Comparison on Different Dataset

The figure 7 shows the comparison of the Mean absolute error on different datasets. It can be visualized that the error has been reduced in all dataset by the algorithm MRCT as compared to RCT. The reduction is more when the number of attributes and instance are large in the dataset. Moreover, the enhancement in the value of the correlation coefficient can also be recognized. The increase in correlation coefficient value and decrease in the error value shows the enhanced performance of MRCT as compared to RCT.

8. Conclusion & Future Scope

The Big Data is nothing but a data, available at heterogeneous, autonomous sources, in extreme large amount, which get updated in fractions of seconds. The existing data mining technique doesn't provide efficient results on the big data. The optimization of existing techniques may provide the efficient results on such data. The main reason for the degradation of the performance on the dataset is the large number of attributes available to be processed. This work reduces the number of attributes. The MRCT modifies the RCT by including a step for the dimension reduction. The dimension reduction is done by selecting the principal components of the dataset. The principal components are found by determining the scatter matrix of the attribute data. The simulation results show the comparison of the MRCT with RCT different datasets. It can be visualized that the error has been reduced in all dataset by the algorithm MRCT as compared to RCT. The reduction is more when the number of

attributes and instance are large in the dataset. The results show the significance of the proposed technique. In future the work can be extended to use the soft computing techniques for the adaptability of the algorithm according to dataset.

References

[1] Houle, M. E., & Nett, M. (2015). Rank-Based Similarity Search: Reducing the Dimensional Dependence. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 37(1), 136-150, 2015.

[2] Wu, Xindong, et al. "Data mining with big data." Knowledge and Data Engineering, IEEE Transactions on 26.1 (2013), pp: 97-107

[3] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[4] Danso, O. S. (2006). An Exploration of Classification Prediction Techniques in Data Mining: The insurance domain (Doctoral dissertation, M. Sc. Thesis, Bournemouth University, United Kingdom).

[5] M. E. Houle and J. Sakuma, "Fast approximate similarity search in extremely high-dimensional data sets," in Proc. 21st Intern. Conf. Data Eng., pp. 619–630, 2005.

[6] Dasarathy, B. V., "Nearest Neighbor (NN) Norms,NN Pattern Classification Techniques". IEEE Computer Society Press, 1990.

[7] James Reason, Human Error. Ashgate. ISBN 1-84014-104-2, 1990.

[8] http://www.r-tutor.com/elementarystatistics/ numerical-measures/correlation-coefficient.