A Novel K-Nearest Neighbor Technique for Data Clustering using Swarm Optimization

Malini Devi, G.,[1] Seetha, M.[1] and Sunitha, K. V. N[2]
[1]CSE, GNITS, Hyderabad, India, E-mail: gmalini12@gmail.com, seetha.maddala@gmail.com
[2]BVRIT Hyderabad, India, E-mail: k.v.n.sunitha@gmail.com

Abstract
Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. The k nearest neighbors is selected by using a predefined distance metric (Hamming distance, Euclidean distance etc) to sense the selected similarity metrics. With KNN technique dimensionality reduction is applied to avoid the effects of the curse of dimensionality. The PSO algorithm optimizes the performance of a KNN classifier by finding the best k values for production of the best clustering performance. This paper presents enhanced method of clustering using K-Nearest neighbor with particle swarm optimization (KNN_PSO) over K-Nearest neighbor (KNN) algorithms which can be traceable even for large datasets. The KNN, PSO and KNN_PSO clustering algorithms are analyzed for different datasets using accuracy as the performance measure. The experimental results exhibit that the clustering using K-Nearest Neighbor with PSO approach outperforms K-Nearest Neighbor algorithm with respect to overall accuracy.

1. Introduction

Cluster analysis seeks to systematize information about variables so that relative homogeneous groups or clusters can be formed. Highly internal homogenous (elements which are similar to one another) and highly external homogenous (elements which are dissimilar to one another) clusters are formed with this type of method. Different measures of similarity may be used to place elements into classes depending on the nature of data, purpose and also similarity measure controls how the clusters are formed. Cluster analysis (Taneja et al., 2014) is an iterative process optimization differs with automatic task and involves more of knowledge discovery or interaction with multi objective trial and failure. KNN uses experiences from earlier training patterns for analyzing the data. The input test data is clustered into a certain class by using majority voting among the K-nearest neighbors. By using a pre-defined distance measures (Hamming distance, Euclidean distance etc.,), the K nearest neighbors are selected (Zhou Hong and Jun-Tao, 2014). This is in turn used to compute and select nearest training pattern that are neighbor to input sample of selected measures. Particle Swarm Optimization (PSO) (Kennedy, 1997) is a population based technique. To realize a self-evolution method in bird flocking or fish schooling behavior, this algorithm can be implemented. Even though the search process is not random it searches for optimum solution. Depending on different problems, it decides the search mode for evaluating the fitness function.

PSO (Eberhart and Kennedy, 1995) needs smaller parameters to decide the solution, compared to other evolutionary algorithms. PSO has a stable convergence character with great computational efficiency and is easily implemented. KNN, the traditional clustering analysis method, inefficient at local minima and cannot be robust to cluster complex data sets. A highly capable evolutionary based clustering method by PSO is provided to find the near optimal solution in search space to trounce the previous problems. KNN (Liu et al., 2014) is a type of instance-based learning or lazy learning where the fitness function is approximated locally and all computation is deferred in internal clustering. The k- nearest classifier (Andrade Silva and Hruschka, 2013) is used to evaluate each data set. The scaled or selected data are used as input to another nearest neighbor classifier in the second stage. Particle Swarm Optimization (PSO) is a swarm intelligence method for global optimization. Each individual of the population adjusts its trajectory to its own previous best pattern, and towards the previous best position attained by any member of its topological neighborhood in classical PSO. So, finding an optimal solution is a challenging task in data clustering. From the current literature, static techniques cannot be useful without certain assumptions about data. To combat this, a novel method is proposed for improving clustering performance of static techniques combined with PSO.

This paper emphasizes a novel K-Nearest Neighbor Technique for Data Clustering using Swarm Optimization for large datasets within the given time limit on various data sets. Hence, this paper describes KNN technique in section 2.Particle swarm optimization with steady convergence spirit has been presented in section 3.Section 4 illustrates the KNN technique combined with particle swarm optimization which elucidates gleaming clustering accuracy. The results of hybrid of PSO and KNN are discussed and analyzed in section 5 and finally conclusions are depicted in section 6.

2. K- Nearest Neighbor Technique

The *k*-nearest neighbor technique (KNN) (Jing et al., 2014) is a method for classifying objects based on closest training examples in the feature space. Clustering using a KNN classifier uses experience from the previous training patterns. The input test data is classified into a certain class, by using majority voting among the nearest k neighbors. It also overcomes the defects of uneven distribution of training data. KNN pseudo-code can be illustrated as shown below:

- In the classification phase, *k* is a user defined constant, and an unlabeled vector (a query or test point) is classified by calculating the distance between the query point and all the points in the dataset.
- The loop repeats until all the points in the dataset are completed.
- The obtained distances are then sorted and according to the k value, those neighbors are plotted on the graph.
- If the neighbor lies in a particular class, then it is considered as true positive otherwise it is taken as true negative.
- Depending on the true positive and true negative value the accuracy is calculated.

The naive version of the algorithm is easy to implement, but which is computationally intensive for large data sets. Using (Balasaravanan and Duraiswamy, 2011) a suitable nearest neighbor search technique, this makes KNN computationally traceable even for large data sets. Several nearest neighbor search algorithms (Liu et al., 2014) have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed. The k-nearest neighbors are selected by using a predefined distance metric (Hamming distance, Euclidean distance etc.) to sense the selected metric.

2.1 Similarity Measure

In most algorithms, the dataset to be classified is represented as a set of vectors X={x1, x2, …., xn}, where the object xi corresponds to a single object and is called the feature object. The feature object should include proper features to represent the object. Measuring (Jing et al., 2013) the similarity between two data objects is used in cluster analysis. Two well-known behaviors have been projected to evaluate the similarity between objects m_p and m_j.

2.2 Euclidean Distance

It can be estimated as

If $p = (p_1, p_2,..., p_n)$ and $q = (q_1, q_2,..., q_n)$ are two data points in euclidean *n*-space, then the distance from q to p or from p to q, is represented by:

$$d(p, q)=d(q, p)= \sqrt{(q_1 - p_1)^2+(q_2 - p_2)^2 + \cdots} \\ +(q_n - p_n)^2=\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Equation 1

2.3 Minkowski Distance

It can be computed as:

$$D_n (m_p, m_j)=[\Sigma_{i=1}^{dm}|m_{i,p}-m_{i,j}|^n]^{1/n}$$

Equation 2

This algorithm uses the normalized Euclidean distance as the similarity metric of two data objects, *mp* and *mj*, in the dimension space.

$$d(m_p,m_j)= \sqrt{([\Sigma_{i=1}^{dm}|m_{pk}-m_{jk}|^2]/d_m)}$$

Equation 3

Where m_p and m_j are two data objects; d_m denotes the dimension number of the vector space; m_{pk} and m_{jk} stand for the objects m_p and m_j's weight values in dimension *k*.

2.4 Cosine Correlation

The other frequently used similarity measure in data clustering is the cosine correlation measure, given by:

$$Cos(m_p, m_j)=m_p^t m_j/|m_p * ||m_j|$$

Equation 4

2.5 Hamming Distance

It is measured as:

$$S_H(x,y)=x^T y+(x1)^T(y1)$$

Equation 5

The term x^Ty denotes the optimistic matches, i.e. the number of "1" bits that match between x and y. The term $(x1)^T(y1)$ is the pessimistic matches, i.e. the number of "0" matching bits.

2.6 City Block Distance

It can be defined with a formula:

$$d_{st} = \sum_{j=1}^{n} |x_{sj} - y_{tj}|$$

Equation 6

City block distance is also called as Manhattan distance. City block distance is a special case of the minkowski distance where p=1.

2.7 Fitness Function

In each iteration, the particle adjusts the centroid vector position in the vector space according to its own experience and those of its neighbors. The average distance between a cluster centroid and a data objects is used as the fitness value to evaluate the solution represented by each particle. The fitness value is measured by the equation below:

$$f = \sum_{i=1}^{N_c} \{\sum_{j=1}^{P_i} d(o_i, m_{ij})/P_i\}/N_c$$

Equation 7

Where O_i is the centroid vector of i^{th} cluster; m_{ij} denotes the j^{th} data objects, which belongs to cluster i; $d(o_i, m_{ij})$ is the distance between objects m_{ij} and the cluster centroid O_i.; P_i stands for the object number, which belongs to cluster C_i; N_c stands for the cluster number.

3. Particle Swarm Optimization

PSO is an evolutionary based computation method that performs robust and efficient optimization. It is a population-based optimization technique, (Ghorpade-Aher and Metre, 2014) where a population is known as a swarm. PSO follows a stochastic optimization method based on swarm intelligence. The basic idea is that each particle represents a potential solution which it updates according to its own experience and that of its neighbors. The PSO algorithm searches in parallel using a group of individuals. Individuals in as swarms, approach to the optimum through the experience of its neighbors, present velocity and previous experience. By adjusting the trajectories of moving points in the multi-dimensional space, the PSO searches the problem domain. (Prasanna and Seetha, 2012). The motion of individual particles for the optimal solution is governed through the interactions. Best accuracy of individuals and their neighbors, velocity and position are used in the connectivity between the individual particles. Improving the performance in terms of convergence rate as well as error rate is very crucial in clustering.

3.1 Notations for PSO

1. The position of the i^{th} particle of a swarm of size n, is projected by the D-dimensional object space, xi= (xi1, xi2, ..., xiD)
2. The best previous position (i.e., the position giving the best function value pBest) of the i^{th} particle is recorded and represented by pi= (pi1, pi2, ..., piD).
3. The position change (velocity) of the i^{th} particle is Vi=(Vi1, Vi2, ..., ViD) .
4. The position of the best particle of the swarm (i.e., the particle with the smallest function value) is denoted by index gBest.
5. The particles are then manipulated according to the following equations.

$$V_{id}(t+1) = V_{id}(t) + C_1 * rand * (pBest(t) - x_{id}(t)) + C_2 * rand * (gBest(t) - x_{id}(t));$$

Equation 8

$$x_{id}(t+1) = x_{id}(t) + V_{id}(t+1)$$

Equation 9

Where d=1, 2,…..D and i=1,2,…..n.
V: Velocity
C1, C2: positive constant parameters =1.49
rand: random numbers between (0, 1).

4. K-Nearest Neighbor Technique with PSO

Clustering techniques are applied in the form of intelligent hybrid system, where two or more techniques are judiciously combined to exploit the strong point of all combined techniques. KNN-PSO (Ma et al., 2011) optimizes the performance of a KNN classifier by finding the best k value that produces the best clustering performance. The KNN-PSO (Imran et al., 2013) takes the output of K-NN technique as input to KNN-PSO algorithm which gives out the global best positions and local best positions. Illustration of PSO pseudo-code can be shown below:

1. begin
2. for each particle: Initialize particle: *xi*
3. do:
4. for each particle:
5. Calculate fitness value: *pi*
6. If the fitness value is better than the best fitness value (*pg*) in History
7. Set current value as the new fitness value.
8. end

9. do:
10. for each particle:
11. Finding the particle with the optimum fitness with respect to the particle neighborhood.
12. Calculate particle velocity: Veli, using equation (8)
13. Apply the velocity constriction.
14. Update particle position: *xi* , using equation (9)
15. end
16. Apply the position construction.
17. end.

With the combined approach i.e., KNN_PSO, the particles come very closer to each other and better fitness can be obtained when compared to KNN technique.

5. Results and Discussions

The experiments have been carried out to authenticate the efficiency of the proposed model. The properties of the dataset are described as:

5.1 Data Sets

Two experimental data sets i.e. glass and zoo are used to test the behavior of the respective clustering methods. These data sets (Muza and, Lowe 2014) represent examples of data with low, medium and high dimensions [6].

- Glass dataset: The glass dataset is from the UCI machine learning repository. This data set has 214 data points and 10 real attributes. Each glass point is classified into 6 classes.
- Zoo dataset: The zoo dataset is from the UCI machine learning repository. This data set has 101 data points, which contain information about an animal in terms of 18 categorical attributes. Each animal data point is classified into 7 classes.

5.2 K-NN Algorithm Results

An assortment of similarity measures like Euclidean distance, Cosine distance, Minkowski distance, Hamming distance and City block distance are analyzed for the number of nearest neighbour from different classes of dataset for a particular seed point. The table 1 represents the summary statistics for glass and zoo data set with no missing values. The glass data set was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence. If it is correctly identified! The zoo dataset contains 17 Boolean-valued attributes. The type attribute appears to be the class attribute.

Table 1: Names of the dataset attributes are minimum, maximum, mean, and standard deviation for glass and zoo dataset

Attributes	Min	Max	Mean	Std Dev
For Glass dataset				
Id Number	1	214	101.75	61.921
Refractive Index	1.511	1.534	1.518	0.003
Sodium	10.73	17.38	13.408	0.817
Magnesium	0	4.49	2.685	1.442
Aluminum	0.29	3.5	1.445	0.499
Silicon	69.81	75.41	72.651	0.775
Potassium	0	6.21	0.497	0.522
Calcium	5.43	16.91	8.957	1.423
Barium	0	3.15	0.175	0.497
Iron	0	0.51	0.057	0.097
Class type	1	6	2.542	1.708
For Zoo dataset				
Animal Name	0	1	0.426	0.497
Hair	0	1	0.198	0.4
Feathers	0	1	0.584	0.495
Eggs	0	1	0.406	0.494
Milk	0	1	0.238	0.428
Airborne	0	1	0.356	0.481
Aquatic	0	1	0.554	0.5
Predator	0	8	0.604	0.492
Toothed	0	1	0.822	0.385
Backbone	0	1	0.792	0.408
Breathes	0	1	0.079	0.271
Venomous	0	1	0.168	0.376
Fins	0	1	2.842	2.033
Tail	0	1	0.743	0.439
Domestic	0	1	0.129	0.337
Cat size	0	1	0.436	0.498
Class type	1	7	2.832	2.103

Table 2: Accuracy of glass and zoo dataset for different distance measures with respect to number of clusters as k value with KNN technique

k	Euclidean Distance	Cosine Distance	Minkowski Distance	Hamming Distance	City Block Distance
For Glass dataset					
6	14.2857	14.2857	14.2857	28.5714	14.2857
12	14.2857	14.2857	14.2857	57.1429	14.2857
18	28.5714	14.2857	28.5714	57.1429	28.5714
For Zoo dataset					
7	33.333	33.333	33.333	16.667	33.333
14	50	66.667	50	16.667	50
21	66.667	83.33	66.667	16.667	66.667

Table 3: Accuracy of glass and zoo dataset based on distance measures and number of clusters as k value with KNN_PSO

k	Euclidean Distance	Cosine Distance	Minkowski Distance	Hamming Distance	City Block Distance
For Glass dataset					
6	-0.9409	6.3905	-1.0293	9.2539	-0.7665
Time	12.885485	12.962274	11.674620	12.997591	12.896736
12	-4.1083	4.6956	-6.3534	8.7903	-18.7077
Time	12.886033	12.920666	12.896877	12.920670	12.982294
18	-7.8790	4.4602	-14.7649	8.2156	-1.4775
Time	12.922794	12.810396	12.913025	12.526539	12.982275
For Zoo dataset					
7	-0.9409	6.3905	-1.0293	9.2539	-0.7665
Time	12.885485	12.962274	11.674620	12.997591	12.896736
14	-4.1083	4.6956	-6.3534	8.7903	-18.7077
Time	12.886033	12.920666	12.896877	12.920670	12.982294
21	-7.8790	4.4602	-14.7649	8.2156	-1.4775
Time	12.922794	12.810396	12.913025	12.526539	12.982275

5.3 KNN_PSO Algorithm Results

The output of K-NN algorithm is given as input to KNN-PSO algorithm which gives the global best positions and local best positions. KNN-PSO optimizes the performance of a KNN algorithm by finding the best k value that produces the best clustering performance. From table 1 it can be stated that glass dataset is from the UCI Machine Repository. This set has 214 data points and 10 real attributes. Each Glass data point is classified into 6 classes. The rank value is better for fourth and fifth attributes i.e., Magnesium and Aluminum with values (0.66, 0.543) compared to other attributes [13]. For zoo dataset it can be depicted that the rank value is better for fourth and thirteenth attributes i.e., Eggs and Fins with values (0.9743, 1.363) compare to other attributes. From table 2 for glass dataset it can be stated that when the KNN was implemented by varying the k-values, it was observed that the maximum accuracy of 83.33 was obtained at k=18 with cosine distance. It was ascertained that the accuracy is efficient as the number of clusters are increased. The aforementioned results indicate that the different distance measures with number of clustering show different cluster accuracy value. As the number of clusters is increased accuracy is better for cosine distance because training is very fast, robust to noisy training data and effective if training data is large. For hamming distance as the number of clusters is increased the accuracy remained constant because it works well in comparing two fixed-length bit patterns. From table 2 for zoo dataset it can be affirmed that the zoo dataset is from the UCI Machine Repository. This set has 101 data points, which contain information about an animal in terms of 18 categorical attributes. Each animal data point is classified into 7 classes. When the KNN was implemented by varying the k-values, it was observed that the maximum accuracy of 57.1429 was obtained at k=18 with hamming distance. It was observed that the accuracy is efficient as the number of clusters is increased. This algorithm is easily fooled by irrelevant attributes. From table 3 i.e., for glass dataset it can be stated that when compared to all similarity measures, city block distance for k=12 achieved optimized fitness value and takes more time because cluster centers are formed to select nearest neighbor to overcome the defect of uneven distribution of data objects. Table 3 for zoo dataset it states that when compared to all Similarity measures, hamming distance for k=21 achieved optimized fitness value and takes more time as being a supervised learning lazy technique i.e. runs slowly.

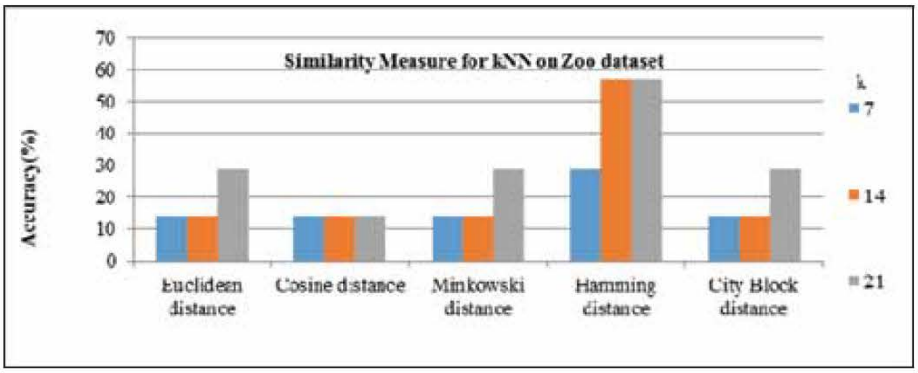

Figure 1: Comparison between the accuracy of Glass dataset for different distance measures and k value

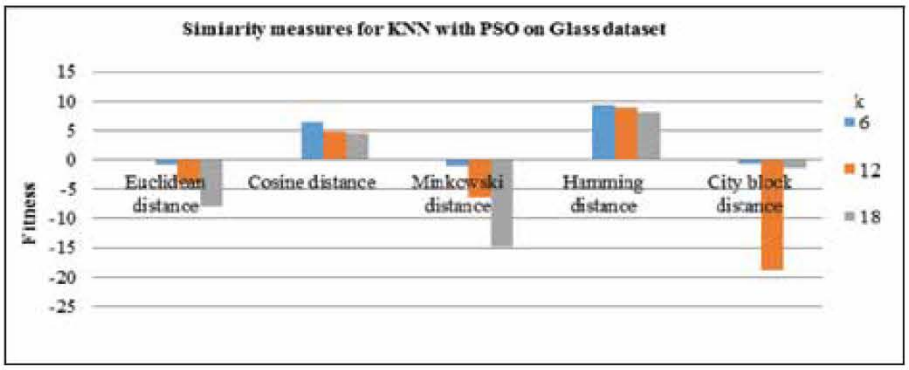

Figure 2: Comparison between the accuracy of Zoo dataset for different distance measures and k value

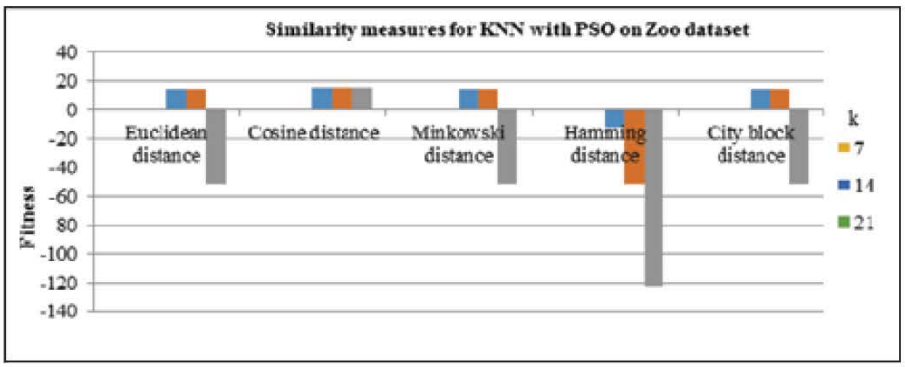

Figure 3: Comparison between the fitness of Glass dataset for different measures and k value of kNN with PSO

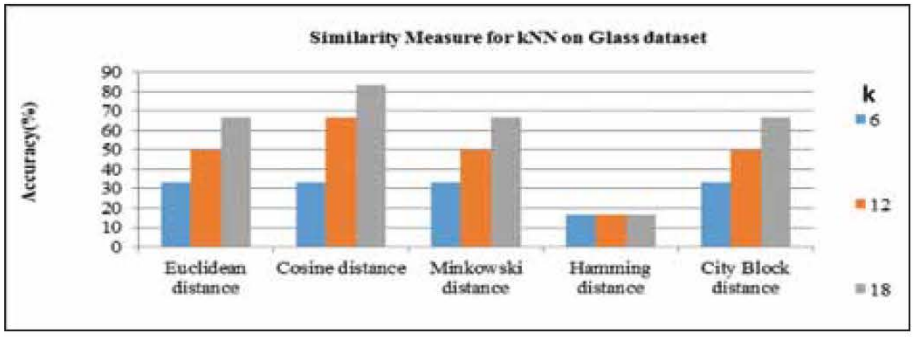

Figure 4: Comparison between the fitness of Zoo dataset for different measures and k value of kNN with PSO

The curse of dimensionality i.e. attribute subset selection depends on the performance of the number of dimensions in the dataset. In figure 1 the accuracy is better for cosine distance on glass dataset when compared to all other similarity measures and hamming distance has the same accuracy for all the k values. As the number of clusters is increased data objects are falling in more number of classes. For cosine distance when k=6, data objects are declining in only two classes, when k=12, data objects are falling in four classes and when k=18, data objects are lying in five classes. With this analysis it can be stated that accuracy is better for k=18 i.e., 83.33%. In figure 2 the accuracy is better for hamming distance on zoo dataset when compared to all other similarity measures for all k values. For cosine distance has the same accuracy for all the k values because selection of threshold parameter is difficult before running algorithm. In figure 3 the fitness values decreased as the number of neighbors increased for all the similarity measures except for city block distance on glass dataset due to high dimensionality. It is very expensive to trace the nearest k neighbors when the data set is very big. In figure 4 as the number of neighbors is increased the fitness value is reduced for all the similarity metrics except for cosine distance on zoo dataset. By the comparison of all similarity measures, optimized fitness value is achieved with hamming distance for k=21 and takes faintly more time. This technique is biased by value of k computation complexity. Hence it is observed that the intelligent hybrid technique is obtained by combining KNN with PSO and better accuracy is achieved compared to KNN.

6. Conclusions

The proposed method elucidates the enhancement of K-nearest neighbor technique with PSO for data clustering. To exploit the strong point of all combined techniques, clustering is useful in the form of intelligent hybrid method where two or more techniques are wisely combined. The KNN, KNN_PSO data clustering techniques are implemented and fitness function value is measured based on similarity measures. The similarity measures like Euclidean distance, Cosine distance, Minkowski distance, Hamming distance and City block distance are implemented for KNN & KNN_PSO techniques. The similarity measures are analyzed on well-known real-life glass and zoo datasets and results are compared. The results display that the k-nearest neighbor with PSO has better overall accuracy when compared to k-nearest neighbor technique. Thus it is illustrated that the proposed method KNN_PSO technique is a viable and an efficient heuristic compared to KNN technique. The work can be extended further with more similarity measures.

References

Andrade Silva, J. and Hruschka, E. R., 2013, An Experimental Study on the use of Nearest Neighbor-Based Imputation Algorithms for Classification Tasks, Elseivier, Vol. 84, 47-58.

Balasaravanan, K. and Duraiswamy, K., 2011, Marginal Object Weight Ranking for Nearest Neighbor Search in Spatial Databases, IJCSI, Vol. 8, 139-142.

Eberhart, R. C. and Kennedy, J., 1995, A New Optimizer using Particle Swarm Theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. 39-43.

Ghorpade-Aher, J. and Metre, V. A., 2014, PSO Based Multidimensional Data Clustering: A Survey, International Journal of Computer Applications (0975 -8887),Volume 87, 41-48.

Imran, M., Hashim, R. and Khalid, N. E. A., 2013, An Overview of Particle Swarm Optimization Variants, Procedia Engineering, Vol. 53, 491–496.

Jing, Y. X., Gou, H. and Zhu, Y., 2013, An Improved Density-Based Method for Reducing Training Data in KNN, Fifth International Conference on Computational and Information Sciences (ICCIS), Issue no-INSPEC NO13874366, 10.1109/ICCIS.2013.261

Kennedy, J. and Eberhart, R. C., 1995, Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. 1942-1948.

Kennedy, J., 1997, The Particle Swarm: Social Adaptation of Knowledge. Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, IN. 303-308.

Liu, C., Cao, L. and Yu, P. S., 2014, Coupled Fuzzy K-Nearest Neighbors Classification of Imbalanced Non-IID Categorical Data- 2014. International Joint Conference on Neural Networks (IJCNN)- ISBN 978-1-4799-6627-1, 10.1109/IJCNN.2014.6889773

Liu, C., Cao, L. and Yu, P. S, 2014, A Hybrid Coupled K-Nearest Neighbor Algorithm on Imbalance Data, International Joint Conference on Neural Networks (IJCNN), Issue no-ISBN978-1-4799-6627-1. 10.1109/IJCNN.2014.6889798.

Ma, X. B. Zhang, C., Shekhar, S., Huang, Y. and Xiong, H., 2011, On a Multi-Type Nearest Neighbor Search, Elseivier, Vol. 70, 955-983.

Muza,, M. and Lowe, D. G., 2014, Scalable Nearest Neighbor Algorithms for High Dimensional Data, IEEE Transcations on Pattern Analysis and Machine Intelligence, Vol. 36, 2227-2240.

Prasanna, K. and Seetha, M., 2012, Association Rule Mining Algorithms for High Dimensional Data- A Review, International Journal of advances in Engineering and Technology, Vol. 2, 443-454.

Salerno, J., 1997, Using the Particle Swarm Optimization Technique to Train a Recurrent Neural Model. IEEE International Conference on Tools with Artificial Intelligence, 45-49.

Taneja, S., Gupta, C., Goyal, K. and Gureja, D., 2014, An Enhanced K-Nearest Neighbor Algorithm using Information Gain and Clustering 2014, Fourth International Conference on Advanced Computing and Communication Technology. 10.1109/ACCT.2014.22.

YinanJing ,Ling Hu ; Wei-Shinn Ku ; Shahabi, C.,2014, Authentication of k Nearest Neighbor Query on Road Networks- IEEE Transactions on Knowledge and Data Engineering.vol no-26

Zhou Hong, B. and Jun Tao, G., 2014, An Automatic Clustering Method Based on Distance Evaluation Function. IEEE Workshop on Electronics, Computer and Applications. 10.1109/IWECA.2014.6845701.