www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 9 Sep 2015, Page No. 14192-14196

An Incremental Shared Nearest Neighbour Clustering Approach For Numerical Data Using An Efficient Distance Measure

B. Naveena Bai[1], *Dr. A. Mary Sowjanya*[2]

[1]Department of Computer Science and Systems Engineering, College of Engineering(A), Andhra University,
Visakhapatnam-530003, India
noveena.singh@gmail.com

[2]Department of Computer Science and Systems Engineering, College of Engineering(A), Andhra University,
Visakhapatnam-530003, India
sowmaa@gmail.com

Abstract: ***Clustering is one of the prominent fields of data mining. A major drawback of traditional clustering algorithms is that they perform clustering on static databases. But in real time databases are dynamic. Therefore incremental clustering algorithms have become an interesting area of research wherein clustering is performed on the incremental data without having to cluster the entire data from scrape. In this paper, a new incremental clustering algorithm called Incremental Shared Nearest Neighbor Clustering Approach (ISNNCA) for numeric data has been proposed. This algorithm performs clustering based on a similarity measure which is obtained from the number of nearest neighbors that two points share. In order to identify nearest neighbors, a distance measure is used. A distance measure that performs well with this algorithm has been identified in this work. This algorithm could find clusters of different shapes, sizes and densities.***

Keywords: Clustering, Incremental Clustering, Similarity, ISNNCA.

1. Introduction

In current scenario, vast amount of data is obtained from various resources .This data has to be maintained in the databases effectively. Data mining and in particular clustering is used to identify the patterns among the data by analyzing them.

Clustering is the process of grouping large datasets where objects in the same group are as similar as possible and different to objects in other groups. It is known as unsupervised learning as no a priori information about the data is required. Clusters emerge naturally from the data under analysis using some distance function to measure the similarity among objects [1]

In real time, new data gets added to the existing databases. For such dynamic databases, the patterns extracted from original database become obsolete. Conventional clustering algorithms handle this problem by repeating the process on entire database [2]

Therefore, in order to handle this problem Incremental clustering algorithms are developed.

In ISNNCA, we make use of Shared Nearest Neighbor (SNN) algorithm. This is a density based algorithm. It makes use of a similarity measure. The similarity is obtained from the number of neighbors shared by two points. Density based algorithms identify the clusters with different sizes and shapes. Using SNN we can identify clusters with different densities also [3]. Clustering on initial dataset is performed using SNN algorithm. Then for incremental updates we make use of ISNNCA.

2. Literature Survey

Incremental clustering is an interesting area of research due to the incremental nature of databases. Several incremental clustering algorithms have been proposed like Incremental K-Means [4], Incremental DBSCAN [5]

In Incremental K-Means, initial clustering is performed using k-means. Whenever incremental updates are available, new cluster centers are obtained by recalculating the mean value. It was stated that this algorithm gives better performance to K-Means but the changes should not exceed a certain limit.

An Incremental clustering algorithm called CFICA (Cluster Feature-Based Incremental Clustering Approach for numerical data) to handle numerical data was proposed by A. M. Sowjanya and M. Shash[5].

In [6] the first incremental version of DBSCAN, called Incremental DBSCAN, was proposed. The main principle of this incremental approach is that on insertion or on deletion of an object p, the set of affected objects is the set of objects in the neighborhood of p with a given radius Eps, NEps (p), and all objects that are density reachable from p or from one of its neighbors. The affected objects may change clusters' membership after an update or a delete operation. All the remaining objects, outside the affected area, will not change cluster membership.

In the DBSCAN algorithm, the Eps parameter represents the radius that limits the neighborhood area of an object, while MinPts represents the minimum number of objects that must exist in the Eps neighborhood of an object to form a cluster.

Later on, a grid density based clustering algorithm (GDCA) was proposed [7]. This algorithm partitions the data space into units and DBSCAN is applied to these units instead of points. An incremental version of this algorithm is also proposed IGDCA (Incremental Grid Density Based Clustering Algorithm) could handle bulks of data.

An incremental version of SNN was proposed [8]. It was stated that it has a high memory usage compared to SNN. Later on another approach was proposed for clustering the spatial data [9]

3. ISNNCA Approach

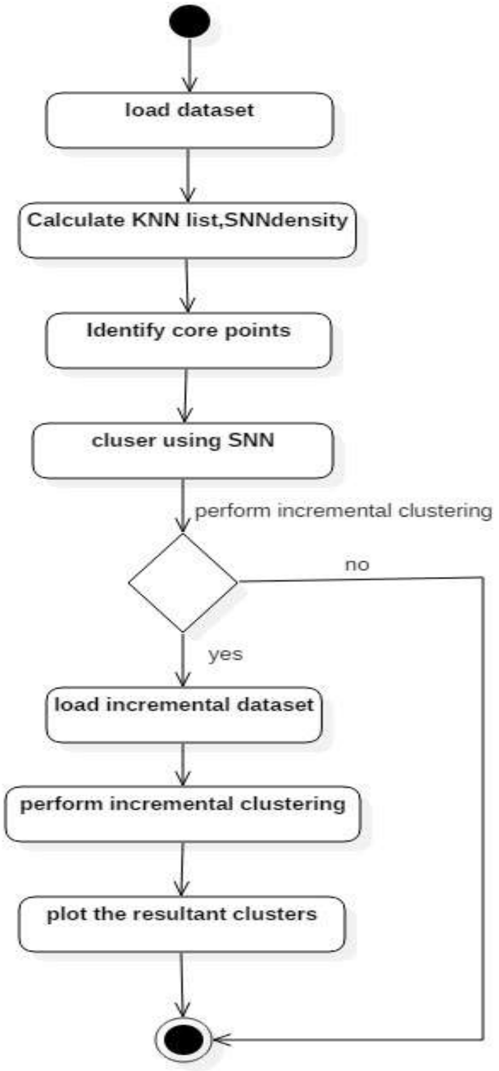
load dataset
Calculate KNN list,SNNdensity
Identify core points
cluser using SNN
perform incremental clustering
no
yes
load incremental dataset
perform incremental clustering
plot the resultant clusters

Figure: 1 Model diagram of ISNNCA

4. Shared Nearest Neighbor Algorithm

The Shared Nearest Neighbor (SNN) [3] is a density based clustering algorithm which is capable of finding clusters of arbitrary shapes, sizes and densities and we need not mention the number of clusters as parameter. This algorithm makes use of a similarity measure which is obtained from the number of neighbors two points share. This can be computed from the k-nearest neighbors of each point. In order to identify the k-nearest neighbors, we need a distance function like Euclidean distance.

This algorithm requires the following inputs:

- K: Number of nearest neighbors to be identified for each point.
- Eps: Density threshold-minimum number of points shared by two points in order to be considered close to each other.
- MinPts: minimum density a point should have to be considered a core point.

Algorithm includes the following steps [10]:

1. Create the distance matrix using a given distance function and identify for each point, the k nearest neighbors.
2. For each two points, calculate the similarity, which is given by the number of shared neighbors.
3. Establish the SNN density of each point. The SNN density is given by the number of nearest neighbors that share Eps or more neighbors.
4. Identify the core points of the data set. Each point that has a SNN density greater or equal to MinPts is considered a core point.
5. Build clusters from core points. Two core points are allocated to the same cluster if they share Eps or more neighbors with each other.
6. Handle noise points. Points not classified as core points and that are not within Eps of a core point are considered noise.
7. Assign the remaining points to clusters. All non-core and non-noise points are assigned to the nearest cluster.

The important step in the SNN algorithm is identification of k-nearest neighbors. In order to identify the nearest neighbors, we make use of a distance function.

In this paper, we identify an efficient distance measure to implement the algorithm.

Various distance functions that could be applicable to numerical data are:

➢ **Euclidean Distance:**

It is the straight line distance between two points. It computes the root of square difference between co-ordinates of pair of objects.

$$\mathrm{Dist}_{XY} = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \quad (1)$$

➢ **Manhattan Distance:**

Manhattan distance computes the absolute differences between coordinates of pair of objects

$$\mathrm{Dist}_{XY} = \sum_{i=1}^{n}\left((|X_i - Y_i|)\right) \quad (2)$$

➢ **Minkowski Distance:**

Minkowski Distance can be defined as the generalized metric distance. It is formulated as

$$\mathrm{Dist}_{XY} = \left(\sum_{i=1}^{n}|X_i - Y_i|^p\right)^{\frac{1}{P}} \quad (3)$$

When p=2, it becomes Euclidean distance.

The algorithm has been implemented by using the above 4 measures in order to obtain the efficient distance measure. It is found that using Manhattan and Chebychev as distance measure, the algorithm doesn’t perform well.

Using Euclidean it has produced the effective results. Since Minkowski also behaves like Euclidean, it also has produced same results.

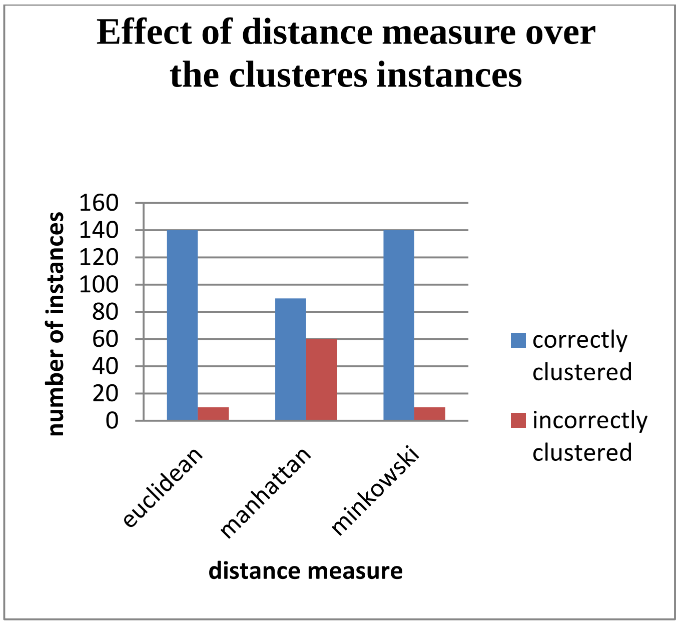

Figure 2: Histogram representing effect of distance measure over clustered instances

Therefore we have implemented the algorithm by using Euclidean distance in this paper.

In the first step, we create a distance matrix and identify the k-nearest neighbours of each point. Then for each two points we calculate similarity i.e. number of nearest neighbours two points share. Then SNN density is computed. The points whose SNN density is greater than MinPts are considered as core points. All core points that share Eps or more neighbours are allocated to the same cluster and continue to be core points. Then we start clustering remaining points with the help of core points.

5. Incremental Shared Nearest Clustering Approach for Numerical Data

Incremental Shared Nearest Neighbour Clustering Approach for numerical data (ISNNCA) uses the same input parameters as that of SNN. Using this approach we calculate k-nearest neighbours for only the incrementally updated points .

Algorithm*:*

1. Read incremental dataset.
2. Calculate the k-nearest neighbours of each new object.
3. Measure the similarities and the densities of objects .
4. Assign the objects to the appropriate cluster.
5. Build new clusters for unassigned objects by identifying core points.
6. Identify noise points by iterating through remaining points.

For each new dataset, above steps are repeated. Whenever a new dataset is read, k-nearest neighbours are computed for each new object. We do not need to recompute the distance between existing points as this approach uses the list computed in previous iteration. Then based on similarities and densities they are assigned to appropriate clusters. For unassigned objects new clusters are formed.

6. Results

Incremental Shared Nearest Neighbor Clustering Approach for Numerical Data was implemented using IRIS dataset and WINE dataset. These datasets were obtained from UCI repository.

A. Results with IRIS dataset:

IRIS dataset has a total of 150 instances.

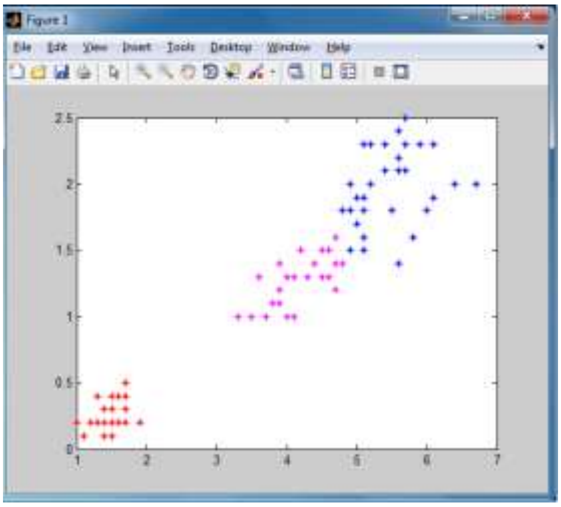

Figure 3: Initial clustering using IRIS dataset

Above screen represents the result after clustering 80 instances of the IRIS dataset.

In the next step remaining instances were added in increments.The following screen represents the incremental clustering

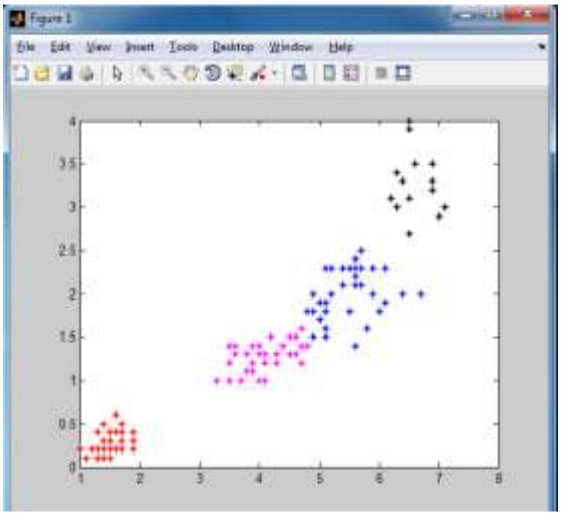

Figure: 4 Incremental clustering using IRIS dataset

The above result shows that after adding new data points, some points were added to the existing clusters and the others which do not belong to any cluster formed new cluster.

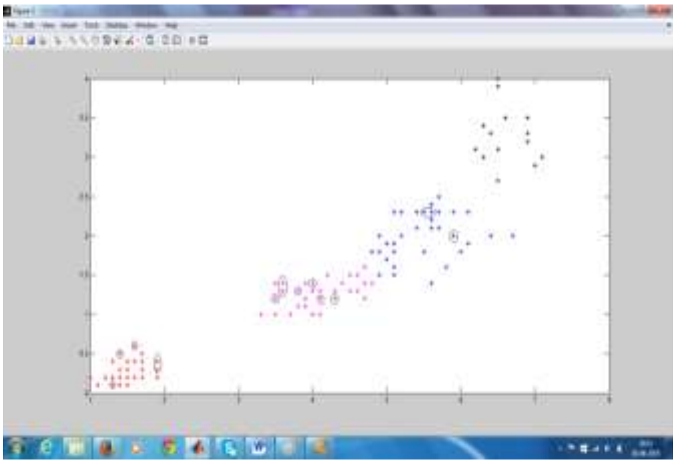

Figure: 5 Ovals represent incrementally added points

to existing clusters.

B. Results with WINE dataset*:*

From WINE dataset,100 instances were choosen in the initial step from among 178 instances.

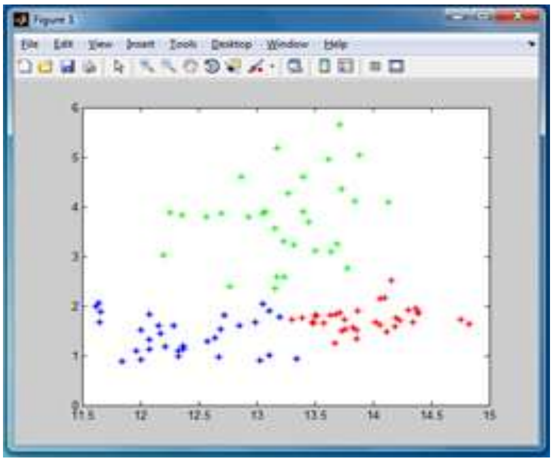

Figure: 6 Initial Clustering of WINE dataset

After initial clustering, ISSNCA was applied on incrementally added points.
The resultant screen is as follows:

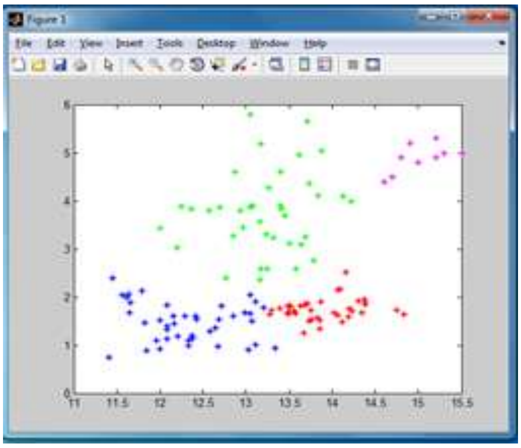

Figure: 7 Incremental Clustering of WINE dataset

7. Conclusions:

SNN (Shared Nearest Neighbor), is a density based algorithm, which has several advantages when analyzing the data due to its ability of identifying clusters of different shapes, sizes and densities.

Incremental Shared Nearest Neighbor Clustering Approach for numerical data (ISNNCA) is based on the SNN algorithm. The ISNN upholds the abilities of the SNN with added advantages of being able to process new data, integrating the new data in the existing clusters without the need to re-compute the entire nearest neighbors list and repeat the whole clustering process.

Moreover, processing huge amounts of data using increments considerably decreases the number of distances that need to be computed to identify the points' nearest neighbors. In Incremental Shared Nearest Neighbor Approach for Numerical data, the algorithm is specifically implemented for only numerical data and using Euclidean distance measure.

In future, this algorithm can be extended to categorical data as well with necessary modifications. We can improve the method of identification of nearest neighbors' list in order to reduce processing time.

References

[1] N. a. G. Goyal, P. and Venkatramaiah, K. and PS, S., "An Efficient Density Based Incremental Clustering Algorithm in Data Warehousing Environment," in 2009 International Conference on Computer Engineering and Applications, 2011.

[2] A. M. Sowjanya, M. Shashi "A new proximity estimate for incremental update of non-uniformly distributed cluster" presented in International Journal of Data Mining and Knowledge Discovery Process(IJDKP),Vol.3,No.5,September 2013

[3] L. Ertöz, M. Steinbach, and V. Kumar, "A new shared nearest neighbor clustering algorithm and its applications," presented at the Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining, 2002.

[4] S. Chakraborty and N. Nagwani, "Analysis and Study of Incremental K-Means Clustering Algorithm," in High Performance Architecture and Grid Computing, ed: Springer, 2011, pp. 338--34

[5] Sowjanya A.M, and M. Shashi,2010.Cluster Feature Based Incremental Clustering Approach (CFICA) for numerical data.IJCSNS Int.JComp.Sci.Network Security.

[6] Ester, H. P. Kriegel, J. Sander, M. Wimmer, and X. Xu,"Incremental clustering for mining in a data warehousing environment," presented at the Proceedings of the International Conference on Very Large Data Bases, 1998.

[7] S. Singh and A. Awekar, "Incremental Shared Nearest Neighbor Density Based Clustering," 1st Indian Workshop on Machine Learning, 2013.

[8] "Dynamics Analytics for Spatial Data with an Incremental Clustering Approach " Fernando Mendes, Maribel Yasmina Santos, João Moura-Pires.

[9] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data,"
presented at the SIAM international conference on data mining, 2003.

[10] Archana Singh, Avantika Yadav,Ajay Rana "K-means with Three different Distance Metrics"

DOI: 10.18535/ijecs/v4i9.22

Author Profile

Naveena Bondili received the B.Tech. degree from Bapatla Engineering College in 2013 and pursuing M.Tech. in Computer Science and Systems Engineering in Andhra University. Presented papers on cloud computing, Brain gate technologies. Served as an organizer for Association of Computer Engineers (ACE) and Module, a departmental session that contains technical and non technical events conducted by students in Bapatla Engineering College.