

IMPROVED DBSCAN CLUSTERING ALGORITHM USING SR-TREE

***Livee Chakrawarty**

****Prof Prateek Gupta**

ABSTRACT

The tremendous amount of data produced now a days in various application domains such as molecular biology or geography can only be fully exploited by efficient and effective data mining tools. One of the primary data mining tasks is clustering, which is the task of partitioning points of a data set into distinct groups (clusters) such that two points from one cluster are similar to each other whereas two points from distinct clusters are not. The detection of clusters in a given dataset is important for data analysis. This paper presents a possible DBSCAN clustering algorithm implementation. DBSCAN algorithm is based on density reachable and density connected point. Adding distributed SR-tree technique to density clustering algorithm we can improve clustering results.

***Keywords—* Clustering, knowledge discovery, DBSCAN, SR-tree.**

***Student, M.Tech IV Semester, SRIST, Jabalpur**

**** H.O.D, SRIST, Jabalpur**

I. Introduction

Due to the enormous amount of data in various application domains, the requirements of database systems have changed. Techniques to analyze the given information and find so far hidden knowledge are mandatory to draw maximum benefit from the collected data. Knowledge Discovery in Databases (KDD) is an interdisciplinary field, aimed at extracting valuable knowledge from large databases [5]. At the core of the KDD [10, 11] process is the Data Mining step which embraces many data mining methods, one of them is clustering.

Clustering is the process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters [1]. Clustering approaches can be classified into partitioning methods, and hierarchical methods. Partitioning clustering algorithms compute a “flat” partition of the data into a given number of clusters, i.e. a unique assignment of each data object to a cluster. The number of clusters k is often a user specified parameter. There are several types of partitioning methods, optimization Based Methods, Distribution- (or Model-) Based Methods, Graph Theory Based Methods, Density-Based Methods. In this paper we are using Density- Based DBSCAN method [3].

The density-based notion is a common approach for clustering, used by various algorithms such as DBSCAN, DBCLASD, DENCLUE, and OPTICS [4]. All these methods search for regions of high density in a feature space that are separated by regions of lower density. DBSCAN [7, 9] was the first density-based spatial clustering method proposed to define a new cluster or to extend an existing cluster, a neighborhood around a point of a given radius (ϵ) must contain at least a minimum number of points (MinPts), the minimum density for the neighborhood. DBSCAN starts from an arbitrary point q . It begins by performing a region query, which finds the neighborhood of point q . If the neighborhood is sparsely populated, i.e., it contains fewer than MinPts points, then point q is labeled as noise. Otherwise, a cluster is created and all points in q 's neighborhood are placed in this cluster. Then the neighborhood of each of q 's neighbors is examined to see if it can be added to the cluster. If so, the process is repeated for every point in

this neighborhood, and so on. If a cluster cannot be expanded further, DBSCAN chooses another arbitrary unlabelled point and repeats the process. This procedure is iterated until all points in the dataset have been placed in clusters or labelled as noise.

II. Improved DBSCAN Algorithm

When looking at the sample sets of points depicted in figure 1, we can easily and unambiguously detect clusters of points and noise points not belonging to any of those clusters.

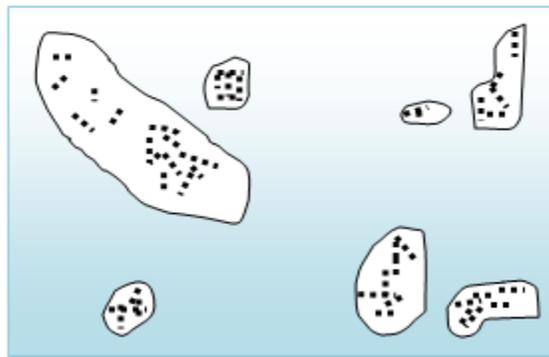


Fig. 1 Sample Databases

The main reason why we recognize the clusters is that within each cluster we have a typical density of points which is considerably higher than outside of the cluster. Furthermore, the density within the areas of noise is lower than the density in any of the clusters [2].

The DBSCAN algorithm is a fundamental density-based clustering algorithm. Its advantage is that it can discover clusters with arbitrary shapes. The algorithm typically regards clusters as dense regions of objects in the data space which are separated by regions of low density objects.

Recently, similarity queries on feature vectors have been widely used to perform content-based retrieval of images. To apply this technique to large databases, it is required to develop multidimensional index structures supporting nearest neighbor queries efficiently. **Improved DBSCAN Algorithm**, In order to enhance the performance of the DBSCAN algorithm suggest

using **SR-trees** for determining the Eps-neighborhood of an object. The SR-tree is similar to the R*-tree and SS-tree in a certain manner, namely, in both trees the distance equals between the root and the leaves, and the number of the children of each node is limited. To determine the Eps-neighborhood of an object its encapsulating region has to be determined, and the tree has to be traversed from the children of the object to the leaves.

The **SR-tree (Sphere/Rectangle-tree)** had been proposed to outperform other index structures such as the R*-tree [8] and SS-tree. The SR-tree is an index structure for high-dimensional nearest neighbor queries. One of its most important features is that it integrates bounding spheres and bounding rectangles. A region of the SR-tree is specified by the intersection of a bounding sphere and a bounding rectangle. This enhances the performance on nearest neighbor queries especially for high dimensional and non-uniform data which can be practical in actual image/video similarity indexing.

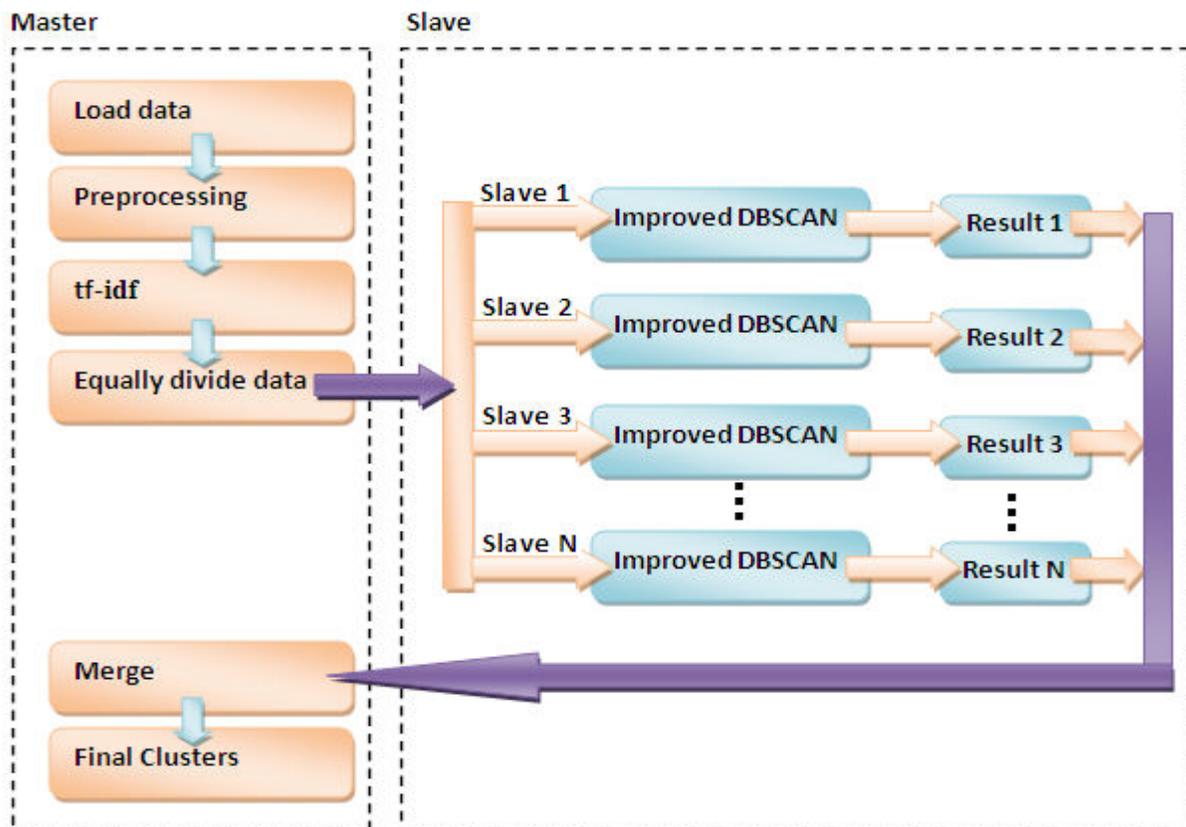


Fig. 2 The structure of the framework based on Master-Slave

The clustering algorithm is described as follow:

- **Step 1:** The master loads the DARPA dataset.
- **Step 2:** The preprocessing (cleaning) of data set is done by master i.e. collecting the necessary information to model.
- **Step 3:** tf-idf stands for “Term Frequency, Inverse Document Frequency”. It is a way to score the importance of words (or “terms”) in a document based on how frequently they appear across multiple documents.
- **Step 4:** The master equally divides the data set and distribute data to the number of slaves connected to the master.
- **Step 5:** All the slaves perform Improved DBSCAN algorithm (using SR-Tree to determine nearest neighbor) simultaneously. Individually, the slaves generate the result and send the result to the master
- **Step 6:** On receiving the result, master merge the homogenous cluster.
- **Step 7:** final clusters are displayed after the merging

III. Data Set

In this paper, we get information of system calls firstly, and use tf-idf frequency [12-15] method to pretreatment that calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. We would make pretreatment process mapping for the text classification process.

In the experiment, we extract a large amount of DARPA data sets [6] a total of approximately 10,000 data as training data sample. We use the frequency of the system calls in the process, and ignore the context of system calls. We will greatly reduce the dimensions of the data in this way. The difference of characteristic between normal and abnormal behavior become greater through the above data processing methods. And not only makes a higher rate of true positive, but also decrease the rate of false positives.

IV. Simulation Results

To be clearer, we use two performance measures in our experiments. The true positive rate (TPR) measures the percentage of correctly classified in the test set. The false positive rate (FPR) measures the percentage of classified in the test set that are detected false.

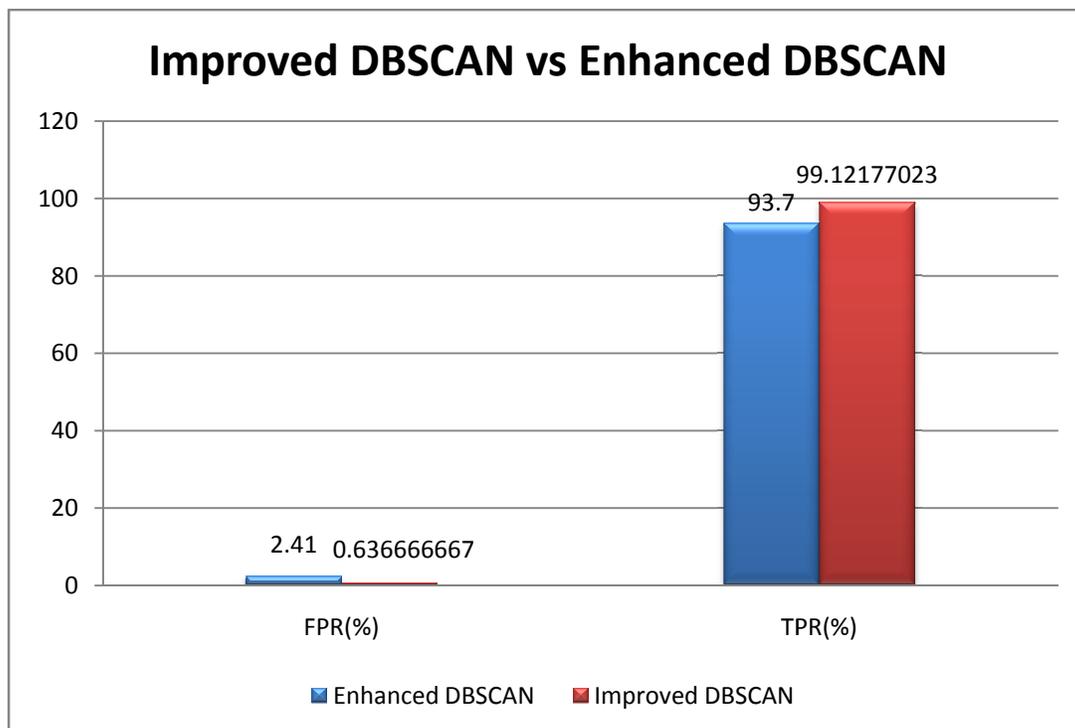


Fig 3. Comparing the number of steps of the different algorithms

According to the experiment results, the average results which are generated by the two algorithms tested with the same suite of data sets are shown in figure 3. It is obviously that we get a higher true positive rate and a low rate of false positives of DARPA data sets. So our method is more feasible and effective.

V. Conclusion

This paper has investigated the use of Improved DBSCAN algorithm in analyzing the results of the performance measurements, the conclusion can be drawn that the clustering based on Improved DBSCAN algorithm achieves the higher recognition accuracy than other method. We get lots of data, and proved the validity of such data through the data pre-processing algorithm, we had done the preparatory work for future after the experiment. And we found an increase clustering algorithm which can dynamic rectify the profile. When we test the algorithm, it has special fast speed.

REFERENCES

- [1] Jiawei Han Micheline Kamber “Data Mining Concept and Techniques”, San Francisco, California
- [2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Published in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining.
- [3] Xin Wang, “Density-Based Spatial Clustering Methods for Very Large Datasets”, December 2006.
- [4] Karin Kailing Tag, “New Techniques for Clustering Complex Objects”, 2004.
- [5] S. Guha, R. Rastogi, and K. Shim, “An efficient clustering algorithm for large databases”, In Proc. SIGMOD, pages 73{84, 1998.
- [6] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba, K.Das. The 1999 DARPA off-line intrusion detection evaluation, Computer Networks 34. 2000. pp.579–595.
- [7] H.P. Kriegel, J. Sander, X. Xu, M. Ester, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)1996, pp. 226–231.
- [8] A.Guttman, “R-trees: a Dynamic Index Structure for Spatial Searching,“ Proc. ACM SIGMOD, Boston, USA, page 47-57, Jun. 1984.

- [9] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)1996, pp. 226–231.
- [10] Brachmann R. and Anand T., 1996: “*The Process of Knowledge Discovery in Databases: A Human Centered Approach*”, in: Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park.
- [11] Fayyad, U. M., .J., Piatetsky-Shapiro, G., Smyth, P. 1996: “*to Knowledge Discovery and Data Mining: Towards a Unifying Framework*”, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, Oregon, AAAI Press, Menlo Park, California.
- [12] Oren, Nir. (2002). Reexamining *tf.idf* based information retrieval with Genetic Programming. In *Proceedings of SAICSIT 2002*, 1-10.
- [13] Salton, G. & Buckley, C. (1988). Term-weighting approache sin automatic text retrieval. In *Information Processing & Management*, 24(5): 513-523.
- [14] H. Wu and R. Luk and K. Wong and K. Kwok. "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.
- [15] K. Sparck Jones. "A statistical interpretation of term specificity and its application in retrieval". Journal of Documentation, 28 (1). 1972.