International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 7– No. 2, April 2014 – www.ijais.org

Implementation of DBSCAN Algorithm using Similarity Measure from Rapid Miner

Tanu Verma
Student
CSE, ITM University

Renu
Student
CSE, ITM University

Deepti Gaur
Associate Professor
CSE, ITM University

ABSTRACT
Clustering methods can be categorized into two main types: fuzzy clustering and hard clustering. In fuzzy clustering, data points can belong to more than one cluster with probabilities. In hard clustering, data points are divided into distinct clusters, where each data point can belong to one and only one cluster[1]. In this paper, we have calculated similarity measure in Rapid miner.

Keywords
DBSCAN, similarity measure,

1. INRODUCTION
The DBSCAN [10] is density fundamental cluster formation. Its advantage is that it can discover clusters with arbitrary shapes and size. T he algorithm typically regards clusters as dense regions of objects in the data space that are separated by regions of low-density objects. The algorithm has two input parameters, radius Eps and MinPts.
According to the above definitions, it only needs to find out all the maximal densityconnected spaces to cluster the data points in an attribute space. And these density- connected spaces are the clusters. Every object not contained in any cluster is considered noise and can be ignored.

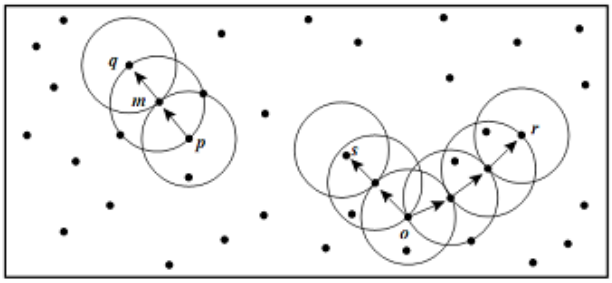

Figure 1 Density reachability and density connectivity in density-based clustering

Explanation of DBSCAN Steps:

1. DBSCAN requires two parameters: Eps and MinPts. It starts with an arbitrary starting point that has not been visited. From the strating point the neighbor points within distance Eps of the starting point.
2. A cluster is formed when the number of neighbors is greater than or equal to MinPts. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the evaluation process for all the neighbors' recursively.
3. If the number of neighbors is less than MinPts, the point is marked as noise.
4. If all points within reach are visited then the algorithm proceeds to iterate through the remaining unvisited points in the data set.

DBSCAN Problems []:

1. When applying the DBSCAN algorithm, we must input two parameters:
   - Eps
   - MinPts
2. Another issues in Eps value that it is determined globally, so due to a single global parameter Eps, it is impossible to detect some clusters using one global-MinPts.
3. DBSCAN performance is poor on multi-density data sets. In the multi-density data set, DBSCAN may merge different clusters and may also neglect other clusters that assign them as noise.
4. Also the runtime complexity of constructing R*-tree and implementation of DBSCAN are not linear

2. CALCULATING SIMILARITY MEASURE IN RAPIDMINER

Process documents Operator in Rapid miner

It generates word vectors from a text object**.**
This operator uses one single TextObject as input for generating a term vector and the result consist of only one single example. This makes this operator especially useful for applying a model on one single text.

Data to Similarity Operator in Rapid Miner

This operator measures the similarity of each example of the given ExampleSet with every other example of the same ExampleSet.

The Data to Similarity operator calculates the similarity among examples of an ExampleSet. Same comparisons are not repeated again e.g. if example *x* is compared with example *y* to compute similarity then example *y* will not be compared again with example *x* to compute similarity because the result will be the same. Thus if there are *n* examples in the ExampleSet, this operator does not return *n^2* similarity comparisons. Instead it returns *(n)(n-1)/2* similarity comparisons. This operator provides many different measures for similarity computation. The measure to use for calculating the similarity can be specified through the parameters. Four types of measures are provided: *mixed measures*, *nominal measures*, *numerical measures* and *Bregman divergences*.

The behavior of this operator can be considered close to a certain scenario of the Cross Distances operator, if the same ExampleSet is provided at both inputs of the Cross Distances operator and the *compute similarities* parameter is also set to true. In this case the Cross Distances operator behaves similar to the Data to Similarity operator. There are a few differences though e.g. in this scenario examples are also compared with themselves and secondly the signs (i.e.+ive or -ive) of the results are also different.

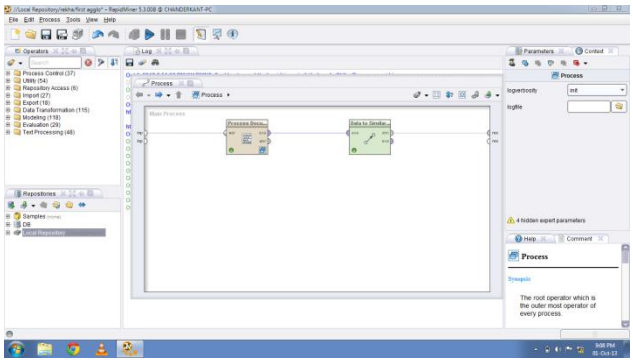

Fig.2 Processing document and data to similarity in Rapid miner

Extract information Operator in Rapid miner

This operator extracts information from a document with structured content.

The purpose of this operator is to extract informations from the structured content of a document. The extracted information will be added as meta data to the document and if wished might be added as attribute later. There are several options available for specifying which information should be extracted. In String Matching mode you may specify a start String and an end String, if both are found in the document, the characters between are extracted. Regular Expressions let you specify any expression and will use the first matching group as extraction. If it's to difficult to include the intermediate characters into the expression in a well defined way, you might find Regular Region mode useful, where you can define two regular expressions. As on String Matching mode, the first defines the start and the last the end and anything intermediate will be extracted. The most sophisticated variant is the XPath mode, where you can enter an arbitrary XPath expression. This proves usefull, especially when trying to extract information from a website. Since XPath expressions are only available for XML files, you will have to take care, that the documents are well defined XML. This might be ensured by the assume_html parameter of the Document Processing operator, that will use a special parser to correct errors in the HTML.

Tokenize Operator in Rapid miner

Tokenization is the process of breaking a stream of text up into phrases, words, symbols, or other meaningful elements called tokens. The goal of the tokenization is the exploration of the words in a sentence. Textual data is only a textual interpretation or block of characters at the beginning. In information retrieval require the words of the data set. So we require a parser which processes the tokenization of the documents. This may be trivial as the text is already stored in machine-readable formats. But Still there are some problems that has been left, for e.g., the removal of punctuation marks as well as other characters like brackets, hyphens, etc. The main use of tokenization is identification of meaningful keywords. Another problem are abbreviations and acronyms which need to be transformed into a standard form.

- Tokenize :- This operator splits the text of a document into a sequence of tokens. There are several options to define the splitting points. The options are as follows:
- mode:-This selects the tokenization mode. Depending on the mode, split points are chosen differently. The Range is non letters, specify characters, regular expression and the default value is non letters
- characters:- The incoming document will be split into tokens on each of this characters. For example enter a '.' for splitting into sentences. The Range is string and the default value is '.:'
- expression:- This regular expression defines the splitting point. The Range is string.

Transform cases Operator in Rapid miner

Transforms cases of characters in a document. This operator transforms all characters in a document to either lower case or upper case, respectively.

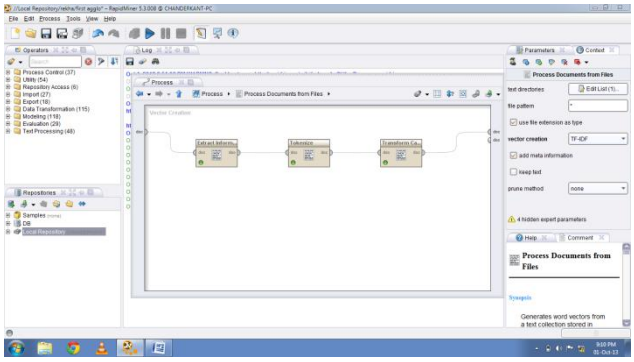

Fig. 3. Calcuation of simirality measure in Rapid Miner

3. IMPLEMENTATION AND RESULT ANALYSIS

The user has to load file in the user interface as shown in the figure.

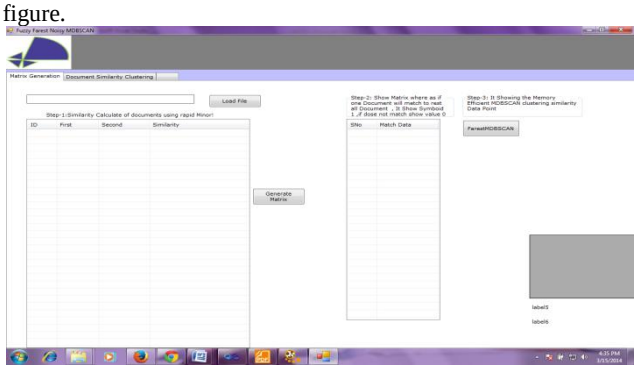

Fig 4. Output of rapid miner file is loaded.

The output is generated in the in the form matrix. The output is zero if one document match with rest of all document otherwise the output is zero.

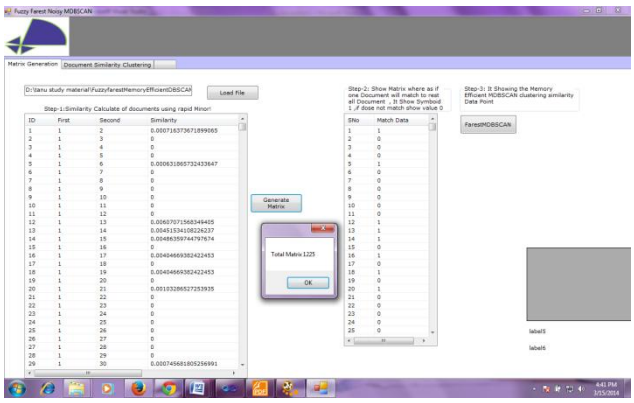

Fig 5. The matrix generated by the file.

On clicking the button from figure 5 the output give sthe clusters as showm in figure 6.

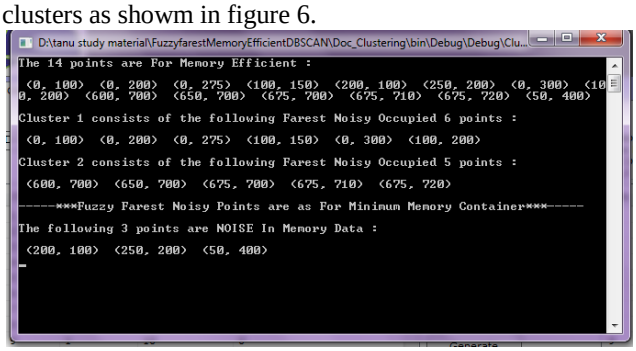

Fig 6. This figure shows the number of clusters.

4. RESULT AND ANALYSIS

This paper gives a user friendly user interface for clustering Experimental results demonstrate that our algorithm is effective and efficient and outperform DBSCAN in detecting clusters of different densities and in eliminating noises.
The experiments show the efficiency of the new algorithms, and get best results with minimum errors.

5. CONCLUSION AND FUTURE SCOPE

The future work can be focused on to reduce the time complexity of algorithms. Future research will have to consider cases when the points inside of a cluster are non-uniformly distributed. Other unknown distributions of the points should be investigated.

6. ACKNOWLEDGEMENT

The authors thank to Dr. Deepti Gaur for her technical support and the reviewers for their valuable suggestion so that we can improve this topic.

7. REFERNCES

[1] J., Data Mining Concepts and Techniques. Kaufman, 2006

[2] Margaret H. Dunham, Data Mining "Introduction and Advanced Topics".

[3] IEEE Trans Xu R. Survey of clustering algorithms. Neural Networks 2005;16.

[4] Anant Ram, Sunita Jalal, Anand S. Jalal, Manoj kumar, "A density Based Algorithm for Discovery Density Varied cluster in Large spatial Databases", International Journal of Computer Application Volume 3,No.6, June 2010.

[5] Derya Birant, Alp Kut, "ST-DBSCAN: An Algorithm for Clustering Spatial-temporal data" Data and Knowledge Engineering 2007 pg 208-221.

[6] Peng Liu, Dong Zhou, Naijun Wu," Varied Density Based Sp6ial Clustering of Application with Noise", in proceedings of IEEE Conference ICSSSM 2007 pg 528-531.

[7] A.K.M Rasheduzzaman Chowdhury, Md.Asikur Rahman, "An efficient Mehtod for subjectively choosing parameter k automatically in VDBSCAN",proceedings of ICCAE 2010 IEEE ,Vol 1,pg 38-41.

[8] Mohammad N. T. Elbatta, An improvement of DBSCAN algorithm for best results in varied densities.