ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Efficient Analysis of Pharmaceutical Compound Structure Based on Enhanced K-Means Clustering Algorithm

V. Palanisamy[1], A. Kumarkombaiya[2]

Professor and Head, Department of Computer Science and Engineering, Alagappa University, India[1]

Assistant Professor, Department of Computer Science, Chikkanna Government Arts College, India

ABSTRACT: In this paper to focuses on discovery of functional group of the connectivity atom for drug effects of chemical compound structured data with position of each atom. A simple Kmeans algorithm, select an initial centroid distance randomly for analysing the data. In the proposed method an Enhanced K means algorithm, forms a functional group of inter connected atoms based on calculate initial centroid distance instead of random selected. The pharmaceutical compounds specifically represented as atom number, atom name like carbon, hydrogen, nitrogen, oxygen with connected atoms. Here it can be experimented the number of iterations are reduced and performance of time accuracy can improve when compare with chameleon and Birch algorithm.

KEYWORDS: Enhanced K-Mean clustering algorithm; Chameleon algorithm; Birch algorithm;

I. Introduction

The main purpose of energy efficient algorithm is to maximize the network lifetime. These algorithms are not just related to maximize the total energy consumption of the route but also to maximize the life time of each node in the network to increase the network lifetime. Energy efficient algorithms can be based on the two metrics: i) Minimizing total transmission energy ii) maximizing network lifetime. The first metric focuses on the total transmission energy used to send the packets from source to destination by selecting the large number of hops criteria. Second metric focuses on the residual batter energy level of entire network or individual battery energy of a node Data Mining is the core of the KDD process, involving the inferring of algorithms that explore the data, develop the model and discover previously unknown patterns. The model is used for understanding phenomena from the data, analysis and prediction.

The knowledge discovery process is an iterative and interactive, consisting of nine steps.

- Developing an understanding of the application domain.
- Selecting and creating a data set on which discovery will be performed.
- Preprocessing and cleansing
- Data transformation
- Choosing the appropriate Data Mining task
- Choosing the Data Mining algorithm
- Employing the Data Mining algorithm.
- Evaluation
- Using the discovered knowledge

Discovery methods are those that automatically identify patterns in the data. The discovery method branch consists of prediction methods versus description methods. It also develops patterns, which form the discovered knowledge in a way which is understandable and easy to operate upon. Some Prediction-oriented methods can also help provide understanding of the data. Most of the discovery-oriented Data Mining techniques (quantitative in particular) are based on inductive learning, where a model is constructed, explicitly or implicitly.

The goal of clustering is descriptive, that of classification is predictive. Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. Clustering is the grouping of similar instances or objects, some sort of measure that can determine whether two objects are similar or dissimilar is required. There are two main type of measures used to estimate this relation: distance measures and similarity measures. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances *xi* and *xj* as: d (*xi*, *xj*).

II. Related work

Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually are either agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

The hierarchical clustering methods could be further divided according to the manner that the similarity measure is calculated.

A. *Single-link clustering*

This method that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

B. *Complete-link clustering*

These methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster (King, 1967).

C. *Average-link clustering*

Methods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster.

Partition algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering. In Non-Hierarchical clustering encompass several methods to build a cluster. Such as,

- A single-pass method is one in which the partition is created by a single pass through the data set or, if randomly accessed, in which each compound is examined only once to decide which cluster it should be assigned to.
- A relocation method is one in which compounds are moved from one cluster to another to try to improve on the initial estimation of the clusters. The relocating is typically accomplished based on improving a cost function describing the ‘‘goodness’’ of each resultant cluster.
- The nearest-neighbor approach is more compounds centered than are the other non-hierarchical methods. In it, the environment around each compound is examined in terms of its most similar neighboring compounds, with commonality between nearest neighbors being used as a criterion for cluster formation.

III. Proposed research Methodology

A. *Data Collection in Proposed Method*

In research involves molecular structure of drugs contains groups of atoms like carbon, hydrogen, oxygen and nitrogen (pharmaceutical drug discovery, databases available in Pubchem) are connected to gather to form different functional groups. The data set is saxagliptin pharmaceutical compound which is taken from PubChem.

The values contain number of atoms and its position in coordinates, connectivity of each atom with bond type which can able to interconnect the atom based on atom type for form a functional group.

B. *BIRCH Clustering Algorithm*

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [2] an algorithm is an agglomerative hierarchical clustering method which builds a dendrogram called clustering feature tree (CF tree) while scanning the data set to condense information about sub-cluster of points. It contains two key phases such as

i) Scans the database to build an in-memory tree and

ii) Applies clustering algorithm to cluster the leaf nodes. Birch handles the task in a very novel manner. It maintains a set of Cluster Features (CF) of the sub- cluster. The criteria for merging two sub-clusters are taken from the information provided solely by the set of CFs of the respective sub-cluster. Each entry in the CF tree represents a cluster of objects and is characterized by a triple feature as (N, LS, SS), where N is the number of data objects in the cluster and LS is the linear sum of the data object and SS is the square sum of the data object in the cluster.

C. *Chameleon Algorithm*

Chameleon's sparse-graph representation of the items is based on the commonly used *k*-nearest-neighbour graph approach. Each vertex of the *k*-nearest-neighbour graph represents a data item. An edge exists between two vertices *v* and *u* if *u* is among the *k* most similar points of *v*, or *v* is among the *k* most similar points of *u*. Data items that are far apart are completely disconnected, and the weights on the edges capture the underlying population density of the space. Items in denser and sparser regions are modelled uniformly, and the sparsity of the representation leads to computationally efficient algorithms. Because Chameleon operates on a sparse graph, each cluster is nothing more than a sub graph of the data set's original sparse-graph representation.

Relative interconnectivity between clusters is their absolute interconnectivity normalized with respect to their internal interconnectivities. By looking at the relative closeness, Chameleon correctly merges clusters so that the resulting cluster has a uniform degree of closeness between its items.

D. *Proposed Work: Enhanced K-Means Algorithm*

There are three phases to construct the functional group of chain details of required compound atom in the pharmaceutical structure similarly as chameleon algorithm. Enhanced K means clustering algorithm can be used based on the initial calculated Centroid distance instead of Agglomerative hierarchical clustering.

In first phase, k-nearest-neighbor graph approach to construct a sparse graph, there exist edges between two vertices, if one object is among the *k*-most-similar objects to other. The edges are weighted to reflect the similarity between objects where each vertex of the graph represents a data object.

In second phase, it can be proposed that the similarity between each pair of clusters such as C_i and C_j by their relative interconnectivity, RI (C_i, C_j), and their relative closeness, RC (C_i, C_j) based on chameleon algorithm.

$$\mathrm{RI}(C_i, C_j) = \frac{\left| EC_{\{C_i, C_j\}} \right|}{\frac{1}{2}\left(\left| EC_{c_i} \right| \left| EC_{c_j} \right| \right)} \qquad \text{eq. (1)}$$

where $EC_{\{C_i,C_j\}}$ is the edge cut, defined as above, for a cluster containing both C_i and C_j. Similarly, EC_{Ci} (or EC_{Cj}) is the minimum sum of the cut edges that partition C_i (or C_j) into two equal parts.

The relative closeness, RC (C_i, C_j), between a pair of cluster is the absolute closeness between the two cluster has normalized with respect to the internal closeness of the two clusters. It is defined as

$$RC(C_i, C_j) = \frac{\bar{S}EC_{\{C_i, C_j\}}}{\frac{|C_i|}{|C_i| + |C_j|}\bar{S}EC_{C_i} + \frac{|C_j|}{|C_i| + |C_j|}\bar{S}EC_{C_j}} \qquad \text{eq. (2)}$$

where $\bar{S}EC_{\{C_i, C_j\}}$ is the average weight of the edges that connect vertices in C_i to vertices in C_j, and $\bar{S}EC_{C_i}$ or ($\bar{S}EC_{C_j}$) is the average weight of the edges that belong to the min bisector of cluster C_i (or C_j).

To apply Enhanced K-means clustering algorithm instead of Agglomerative clustering technique for origin of clustering process of atoms to merging the functional group of related atom for analysing the atom in different stage. Pseudocode of the enhanced Kmeans algorithm is given below,

```
Input:
D = {$a_1$, $a_2$,......$a_n$} // set of n number of atoms data items
K = { $x_1$, $x_2$,$x_3$,… $x_m$}          // Number of desired
      clusters of connectivity atom with bond details
Output:
S     set of k clusters.
Steps:
Phase 1: Determine the initial centroids of the clusters by using Algorithm 1.
Phase 2: Assign each data point to the appropriate clusters by using Algorithm 2.
```

In the first phase, the initial centroids are determined systematically so as to produce clusters with better accuracy [12]. The second phase makes use of a variant of the clustering method discussed in [4]. It starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency. The two phases of the enhanced method are described below as Algorithm 1 and Algorithm 2.

```
Algorithm 1: Finding the initial centroids
Input:
D = {$d_1$, $d_2$... $d_n$} // set of n data items
k // Number of desired clusters
Output: A set of k initial centroids.
Steps:
1. Set n = 1;
2. Compute the distance between each data point and all other data- points in the set D;
3. Find the closest pair of data points from the set D and form a data-point set $a_i$ (1<= i <= k) which contains these  \
   two data- points, Delete these two data points from the set D;
4. Find the data point in D that is closest to the data point set add it to $a_i$ and delete it from D;
5. Repeat step 4 until the number of data points in Sn reaches 0.75*(n/k);
6. If i<k, then i = i+1, find another pair of data points from D between which the distance is the shortest, form
   another data-point set Sn and delete them from D, Go to step 4;
7. For each data-point set $a_i$ (1<=i<=k) find the arithmetic mean of the vectors of data points in Sn, these means will
   be the initial centroids.
```

Algorithm 1 describes the method for finding initial centroids of the clusters [12]. Initially, compute the distances between each data point and all other data points in the set of data points. Then find out the closest pair of data points and form a set S_1 consisting of these two data points, and delete them from the data point set D. Then determine the data point which is closest to the set S_1, add it to S_1 and delete it from D. Repeat this procedure until the number of elements in the set S1 reaches a threshold. At that point go back to the second step and form another data-point set S_2. Repeat this till 'k' such sets of data points are obtained. Finally the initial centroids are obtained by averaging all the vectors in each data-point set. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids. The distance between one vector X = (x_1, x_2, ...x_n) and another vector Y = (y_1, y_2 ….y_n) is obtained as

$$\sqrt{d(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \qquad \text{eq. 3}$$

The distance between a data point X and a data-point set D is defined as d(X, D) = min (d (X, Y), where Y ∈ D). The initial centroids of the clusters are given as input to the second phase, for assigning data-points to appropriate clusters. The steps involved in this phase are outlined as Algorithm 2.

```
Algorithm 2: Assigning data-points to clusters
Input:
D = {$d_1$, $d_2$,......,$d_n$} // set of $n$ data-points.
C = {$c_1$, $c_2$,.......,$c_k$} // set of $k$ centroids
Output:
A set of $k$ clusters
Steps:
1.    Compute the distance of each data-point $di$ (1<=i<=n) to all the centroids $c_j$ (1<=j<=k) as $d(d_i, c_j)$;
2.    For each data-point $di$, find the closest centroid $c_j$ and assign $d_i$ to cluster $j$.
3.  Set ClusterId[i]=j; // j:Id of the closest cluster
4.  Set Nearest_Dist[i]= $d(d_i, c_j)$;
5.  For each cluster $j$ (1<=j<=k), recalculate the centroids;
6.  Repeat
7. For each data-point $d_i$,
7.1 Compute its distance from the centroid of the present nearest cluster;
7.2 If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;
 Else
7.2.1 For every centroid $cj$ (1<=j<=k) Compute the distance $d(d_i, c_j)$;
    Endfor;
7.2.2 Assign the data-point $di$ to the cluster with the nearest centroid $c_j$
7.2.3 Set ClusterId[i]=j;
7.2.4 Set Nearest_Dist[i]= $d(d_i, c_j)$;
        Endfor;
8. For each cluster $j$ (1<=j<=k), recalculate the centroids;
Until the convergence criteria is met.
```

IV. ANALYSIS OF RESULT AND EVALUATION

From fig.1, represents the cluster formation of given number of atom, after merge the inter connectivity of atom functionally form a group which can be measured based on centroid point initially and get the result efficiently, the performance can be experimented in MAT LAB as shown in fig 2. Finally, the proposed algorithm can functioned by the value of initial centroid distance with respect to the number of desired clusters as an input and form the functional group of cluster for chain details analyse efficient by the improvement of K- Means algorithm. From Fig.3 shows the accuracy of each algorithm for time taken for analyse the process to form the group of cluster. Here Chameleon Algorithm takes 55 seconds for analysing the process, Birch algorithm takes 51 seconds and Enhanced K means Clustering algorithm takes 49 seconds for analysing the cluster the atoms.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

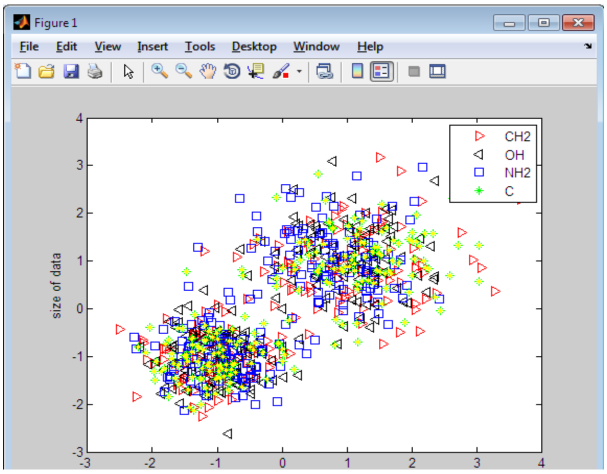

Fig.1. Cluster formation for given atoms

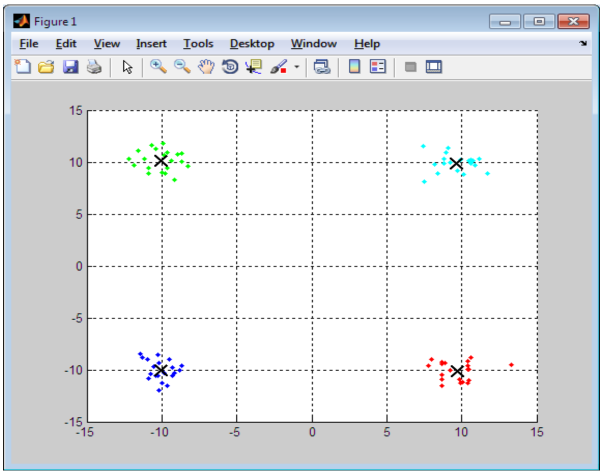

Fig. 2. Functional group of Inter connectivity of atoms as in chain detail

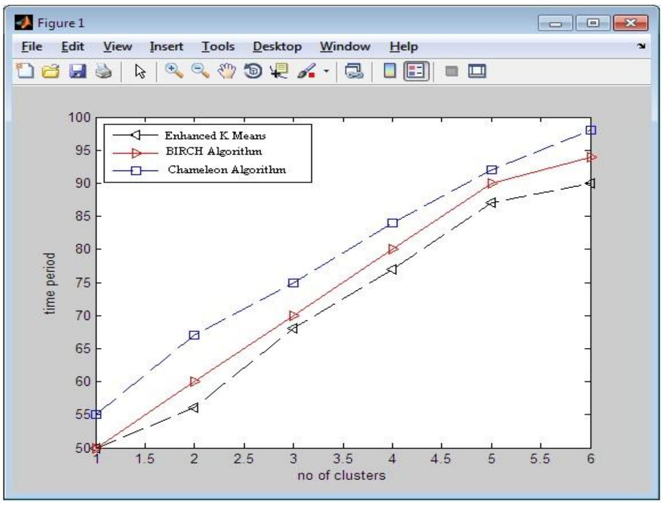

Fig. 3. Accuracy for comparing Chameleon, Birch and Enhanced Kmeans Algorithm

V. Conclusion and Future Work

The clustering results showed that the proposed algorithm can be formed the functional group of interconnected atom details and grouped as a chain detail. Instead of Agglomerative Hierarchical clustering technique Enhanced K Means algorithm can be performed very well to experiment by other than Birch and chameleon algorithm. As the performance, Enhanced Kmeans clustering algorithm took 49 seconds for executing cluster efficiently.

References.

1. Fahim A.M, Salem A. M, Torkey A and Ramadan M. A, "An Efficient enhanced k-means clustering algorithm," Journal of Zhejiang University, 10(7):1626–1633, 2006.
2. Yuan F, Meng Z. H, Zhang H. X and Dong C. R, "A New Algorithm to get the Initial Centroids," Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.
3. Daxin Jiang, Chum Tong and Aidong Zhang, "Cluster Analysis for Gene Expression Data," IEEE Transactions on Data and Knowledge Engineering, 16(11): 1370-1386, 2004.

4. PALANISAMY, A. KUMARKOMBAIYA, "ANALYSING PHARMACEUTICAL COMPOUNDS BASED ON CLUSTER TECHNIQUES", INTERNATIONAL JOURNAL OF COMPUTER SCIENCE RESEARCH & TECHNOLOGY, ISSN: 2321-8827, VOL. 1(03).
5. Jiawei Han and Micheline Kamber," Data Mining: Concepts and Techniques". Publication: ISBN-10: 0123814790 | ISBN-13: 978 0123814791, Edition: 3
6. Zhang, R. Ramakrishnan and M. Livny: BIRCH : "An Efficient Data Clustering Method for Very Large Databases". SIGMOD '96 6/96 Montreal, CanadaIQ1996ACM0-89791-794-4/96/0006.
7. Daniel T. Larose , Data Mining Methods and Models, Copyright © 2006 John Wiley and Sons, Inc.
8. Marjan Kuchaki Rafsanjani, Zahra Asghari Varzaneh, Nasibeh Emami Chukanlo, "A survey of hierarchical clustering algorithms", The Journal of Mathematics and Computer Science Vol .5 No.3 (2012) 229-240.
9. M. R. Anderberg; Cluster Analysis for Applications: Academic Press, New York, 1973.
10. D. Pelleg and A. Moore; X-means: Extending k-means with efficient estimation of the number of clusters: In Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, pp. 727-734, 2000.
11. Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and AngelaY. Wu. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell., 24(7):881–892, 2002.
12. Yuan F, Meng Z. H, Zhang H. X and Dong C. R, "A New Algorithm to Get the Initial Centroids," Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.