TIME COMPLEXITY IMPROVEMENT OF THE FIRST PROCESSING STAGE OF THE INTELLIGENT CLUSTERING

Done Stojanov1,\*, Cveta Martinovska2

*1Faculty of Computer Science, University ,,Goce Delcev”-Stip*

*done.stojanov@ugd.edu.mk*

*2Faculty of Computer Science, University ,,Goce Delcev”-Stip*

*cveta.martinovska@ugd.edu.mk*

*\* Done Stojanov, е - mail: (done.stojanov@ugd.edu.mk)*

**Abstract.** A new approach for data clustering is presented. IC clustering [1] initial processing stage is changed, so that the interval between the smallest and the largest radius-vector is divided into k equal sub-intervals. Each sub-interval is associated to a cluster. Depending on which sub-interval a radius-vector belongs, it is initially distributed within a cluster, associated with that sub-interval.

**Key words:** data clustering, radius-vectors, IC clustering, intervals.

1. **Introduction**

Since the second half of the 20th century, several techniques for data clustering have been proposed. The oldest one, but commonly used technique for data clustering is the k-means [2] algorithm, based on initial selection of k,k<n random objects (centroids) of object set of size n. The remaining n-k objects, which are not selected as centroids, are distributed within the closest clusters. Initially, each centroid represents a cluster. When a cluster is changed, cluster’s center is also changed. Centers no further change implies appropriate data distribution.

PAM (Partitioning Around Medoids) [4] as opposed to the k-means algorithm, effectively handles extreme values ​​(data outliers), which can easily disrupt the overall data distribution. Central objects within clusters (medoids) are used. Medoids are swapped only if that would result with a better data clustering.

CLARA [3] is basically PAM clustering, applied to a part (set of samples) of the object set. The result is not always the optimal one. CLARANS [5] searches graph data structure. Nodes medoids are replaced by nodes non-medoids, if that would reduce the clustering cost.

IC clustering [1] calculates the radius-vector for each object of object set of size n. During the first processing stage, the set of radius-vectors is sorted in ascending order, and then divided into k subsets of approximately equal size, where each subset initially represents a cluster. Next, radius-vectors being closer to the neighboring clusters are moved from one cluster into another. This is repeated until clusters no further change, when all objects are properly partitioned. Finally radius-vector clusters are transformed into object clusters, with properly partitioned objects.

In this paper, IC clustering is changed. Each radius-vector initially is partitioned within a cluster, determined by a sub-interval to which the radius-vector belongs, what in the worst case takes *O(nk)* processing time, where n is the size of the object set, k is the number of clusters, k<n. Certainly *O(nk)<O(n2)*, where *O (n2)* is the time required to sort a set of size n, what implies improved time complexity of the first processing stage of the IC clustering.

1. **Preliminaries**

If a set of  objects  is given, where each object is represented with  attributes (properties), , objects should be properly partitioned in  clusters, where similar objects share a common cluster. There is no empty cluster.

1. **Methodology**

For each object , a radius-vector is calculated. Memory keeps  data pairs, tracking object’s position  in the object set , where  is the radius-vector corresponding to the object at position .

From the set of radius-vectors , the smallest and the largest radius-vector are chosen, , . The interval  is divided into  equal subintervals, starting from  up to . A radius-vector , such as  is satisfied, initially is partitioned in cluster .





.....





Since the data distribution is initiall, some of the radius-vectors might be inappropriately partitioned. The mean values for each two neighboring clusters  and , are calculated according (1) , where  is the number of elements in cluster . A radius-vector , for which  is satisfied, is moved from cluster  in cluster . Thus radius-vector  , for which  is satisfied, is moved from cluster  in cluster . When a radius-vector is moved from one cluster into another, clusters’ structure and clusters’ mean values are changed, recalculating clusters’ new mean values  and . Objects are moved from one cluster into another neighboring cluster, until clusters’ structure no further change, when all radius-vectors will be properly partitioned. Using data pairs  information, each radius-vector  is transformed into object ,. Thus clusters of radius-vectors , are transformed into object clusters , having each object  from the object set  properly partitioned in object cluster .

 (1)

1. **Algorithm**

 **Algorithm 1** Improved IC: Intelligent Clustering

**Input:** set of objects О={o1,o2,…,on-1,on}

**Output:** k clusters of objects ocj, 1<=j<=k

*for each object оi which belongs to the object set О{*

calculate its radius-vector Ri;

store data pair (i,Ri) in the memory;

*}*

find the smallest radius-vector Rmin=min{R1,R2,…,Rn-1,Rn};

find the largest radius-vector Rmax=max{R1,R2,…,Rn-1,Rn};

determine sub-intervals sj, 1<=j<=k;

i=1;

j=1;

*while(i<=n){*

*while(j<=k){*

*if(Ri belongs to sub-interval sj){*

add Ri in cluster cј;

break *while(ј<=k)* loop;

*}*

j++;

*}*

i++;

*}*

calculate centers of clusters mcј, 1<=j<=k*;*

**LOOP**: j=1;

 *while(ј<=k-1){*

 *for each Ri which belongs to cluster cj*

  *if (|Ri-mcj+1|<|Ri-mcj|){*

 move Ri from cluster cј in cluster cј+1;

 calculate clusters’ new mean values mcј and mcј+1;

  *}*

 *for each Ri which belongs to cluster cj+1*

 *if (|Ri-mcj|<|Ri-mcj+1|){*

 move Ri from cluster cј+1 in cluster cј;

 calculate clusters’ new mean values mcј and mcј+1;

 *}*

 j++;

 *}*

go to **LOOP** while at least one mcј is changing;

transform radius-vector clusters cј into object clusters ocј, 1<=j<=k;

1. **An Example**

Set of objects should be partitioned in three clusters. According to the methodology being presented, for each object at position  a radius-vector is calculated, Table 1. Memory keeps ten data pairs, Table 2.

**Table 1** Objects’ radius-vectors

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Object** | (3,4) | (5.7,5.9) | (6,5.7) | (6.1,5.8) | (5.8,5.9) | (4.5,4.9) | (4.6,5) | (7,7) | (4,4) | (8,6) |
| **Radius-vector** | 5 | 8.204 | 8.276 | 8.417 | 8.273 | 6.653 | 6.794 | 9.899 | 5.657 | 10 |

**Table 2** Data pairs 

|  |
| --- |
| **Data pairs** |
| (1,5) |
| (2,8.204) |
| (3,8.276) |
| (4,8.417) |
| (5,8.273) |
| (6,6.653) |
| (7,6.794) |
| (8,9.899) |
| (9,5.657) |
| (10,10) |

Once the smallest and the largest radius-vector have been found, intervals  and  can be determined.







Distributing radius-vector  in cluster  is permitted, only if  belongs to the interval .

Cluster  {5,6.653,5.657}, mean value 

Cluster  {8.204,8.276,8.273,6.794}, mean value 

Cluster  {8.417,9.899,10}, mean value 

A check for radius-vectors, being cluster  less distanced than cluster , is conducted, Table 3.

**Table 3** Calculating the distances between cluster c1 radius-vectors and cluster c1 and c2 mean values

|  |  |  |
| --- | --- | --- |
| **Radius-vector** | **Distance from cluster c1** | **Distance from cluster c2** |
| 5 | |5-5.77|=0.77 | |5-7.887|=2.887 |
| 6.653 | |6.653-5.77|=0.883 | |6.653-7.887|=1.234 |
| 5.657 | |5.657-5.77|=0.113 | |5.657-7.887|=2.23 |

According Table 3, there is no cluster  radius-vector, being closer to cluster  than cluster , what indicates appropriate radius-vector distribution in cluster .

A check for radius-vectors , being closer to cluster  than cluster , has also to be conducted, Table 4.

**Table 4** Calculating the distances between cluster c2 radius-vectors and cluster c1 and c2 mean values

|  |  |  |
| --- | --- | --- |
| **Radius-vector** | **Distance from cluster c2** | **Distance from cluster c1** |
| 8.204 | |8.204-7.887|=0.317 | |8.204-5.77|=2.434 |
| 8.276 | |8.276-7.887|=0.389 | |8.276-5.77|=2.506 |
| 8.273 | |8.273-7.887|=0.386 | |8.273-5.77|=2.503 |
| **6.794** | **|6.794-7.887|=1.093** | **|6.794-5.77|=1.024** |

Considering Table 4 distance results, it can be denoted that radius-vector 6.794 is cluster  less distanced than cluster , where was initially distributed. In this case, radius-vector 6.794 is moved from cluster  in cluster . Since cluster  and cluster  structure has been changed, cluster  and cluster  new mean values are calculated.

Cluster  {5,6.653,5.657,6.794}, mean value 

Cluster  {8.204,8.276,8.273}, mean value 

Cluster  {8.417,9.899,10}, mean value 

Distance results between cluster  radius-vectors and cluster  and cluster  mean values are given in Table 5.

**Table 5** Calculating the distances between cluster c2 radius-vectors and cluster c2 and c3 mean values

|  |  |  |
| --- | --- | --- |
| **Radius-vector** | **Distance from cluster c2** | **Distance from cluster c3** |
| 8.204 | |8.204-8.251|=0.047 | |8.204-9.439|=1.235 |
| 8.276 | |8.276-8.251|=0.025 | |8.276-9.439|=1.163 |
| 8.273 | |8.273-8.251|=0.022 | |8.273-9.439|=1.166 |

Table 5 distance results clearly show that there is no cluster  radius-vector being closer to cluster  than cluster , where from can be concluded that cluster  radius-vectors are properly partitioned.

At the end has to be checked whether exist cluster  radius-vectors being cluster  less distanced than cluster , Тable 6.

**Table 6** Calculating the distances between cluster c3 radius-vectors and cluster c2 and c3 mean values

|  |  |  |
| --- | --- | --- |
| **Radius-vector** | **Distance from cluster c3** | **Distance from cluster c2** |
| **8.417** | **|8.417-9.439|=1.022** | **|8.417-8.251|=0.166** |
| 9.899 | |9.899-9.439|=0.46 | |9.899-8.251|=1.648 |
| 10 | |10-9.439|=0.561 | |10-8.251|=1.749 |

Once again, radius-vector being partitioned in one cluster is closer to the neighboring cluster. Cluster  radius-vector 8.417 is cluster  less distanced than cluster , resulting with rearrangement of radius-vector 8.417, being moved from cluster  in cluster . Since cluster  and cluster  structure is changed, clusters’ new mean values  and  are calculated.

Cluster  {5,6.653,5.657,6.794}, mean value 

Cluster  {8.204,8.276,8.273,8.417}, mean value 

Cluster  {9.899,10}, mean value 

Repeating this procedure from the beginning, no structure change of a cluster is recorded, where from a conclusion for clusters’ no further structure change can be deduced.

Using data pairs , each radius-vector is transformed into object from the object set . Thus radius-vector clusters are transformed into object clusters, having all objects properly partitioned.

Object cluster  {(3,4),(4.5,4.9),(4,4),(4.6,5)}

Object cluster  {(5.7,5.9),(6,5.7),(5.8,5.9),(6.1,5.8)}

Object cluster  {(7,7),(8,6)}

**Conclusion**

A new data clustering technique is presented. Each object is represented with a radius-vector. Instead of sorting a set of radius-vectors of size n (Intelligent Clustering initial processing stage [1]), the interval between the smallest and the largest radius-vector is divided in k equal sub-intervals. Depending on which sub-interval a radius-vector belongs, it is distributed within a particular cluster. Radius-vectors being less distanced to the neighboring clusters are rearranged, moving them from one cluster into another. That is repeated until clusters’ structure no further change, when all radius-vectors are properly partitioned. Finally clusters of radius-vectors are transformed into clusters of objects, having all objects appropriately partitioned.

**References**

[1] D. Stojanov (2012): *IC: Intelligent Clustering, a new time efficient data partitioning methodology*. International Journal of Computer Science and Information Technologies 3(5), pp. 5065-5067.

[2] J. MacQueen (1967): *Some Methods for classification and Analysis of Multivariate Observations*. In Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297.

[3] L. Kaufman and P. Rousseeuw (1990): *Finding Groups in Data, An Introduction to Cluster Analysis*, 99th Edition. Willey-Interscience.

[4] L. Kaufman and P. Rousseeuw (1987): *Clustering by means of medoids*. In Statistical Data Analysis Based on the L1 Norm, pp. 405-416.

[5] R. Ng and J. Han (1994): *Efficient and effective clustering methods for spatial data mining*. In Proc. of the 20th VLDB Conference, pp. 144–155.