УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ” - ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ISSN 1857- 8691

ГОДИШЕН ЗБОРНИК 2012

YEARBOOK 2012

ГОДИНА 1 VOLUME I

GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ГОДИШЕН ЗБОРНИК 2012

YEARBOOK 2012

ГОДИНА 1 МАРТ, 2013 VOLUME I

GOCE DELCEV UNIVERSITY – STIP
FACULTY OF COMPUTER SCIENCE

ГОДИШЕН ЗБОРНИК
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
YEARBOOK
FACULTY OF COMPUTER SCIENCE

За издавачот:
Проф д-р Владо Гичев

Редакција и администрација	Address of the editorial office
Универзитет „Гоце Делчев“-Штип	Goce Delcev University – Stip
Факултет за информатика	Faculty of Computer Science
ул. „Крсте Мисирков“ 10-А	Krste Misirkov 10-A
п. фах 201, 2000 Штип	PO box 201, 2000 Štip,
Р. Македонија	R. of Macedonia

СОДРЖИНА
CONTENT

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

Done Stojanov[1,*], Cveta Martinovska[2]

[1]*Faculty of Computer Science, University ,,Goce Delcev”-Stip*
done.stojanov@ugd.edu.mk
[2]*Faculty of Computer Science, University ,,Goce Delcev”-Stip*
cveta.martinovska@ugd.edu.mk
** Done Stojanov, e - 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 $O(nk)<O(n^2)$, where $O(n^2)$ 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.

2. Preliminaries

If a set of n objects $O=\{o_1,o_2,...o_{n-1},o_n\}$ is given, where each object is represented with m attributes (properties), $o_i=(p_{i,1},p_{i,2},...,p_{i,m-1},p_{i,m})$, objects should be properly partitioned in $k, k<n$ clusters, where similar objects share a common cluster. There is no empty cluster.

3. Methodology

For each object o_i, a radius-vector $R_i = \sqrt{\sum_{k=1}^{m} p_{i,k}^2}, 1 \le i \le n$ is calculated. Memory keeps n data pairs $(i, R_i), 1 \le i \le n$, tracking object's position i in the object set O, where R_i is the radius-vector corresponding to the object at position i.

From the set of radius-vectors $R = \{R_1, R_2, \ldots, R_{n-1}, R_n\}$, the smallest and the largest radius-vector are chosen, $R_{\min} = \min\{R_1, R_2, \ldots, R_{n-1}, R_n\}$, $R_{\max} = \max\{R_1, R_2, \ldots, R_{n-1}, R_n\}$. The interval $[R_{\min}, R_{\max}]$ is divided into k equal subintervals, starting from s_1 up to s_k. A radius-vector R_i, such as $R_i \in s_j, 1 \le i \le n, 1 \le j \le k$ is satisfied, initially is partitioned in cluster c_j.

$$s_1 : [R_{\min}, R_{\min} + \frac{1}{k}(R_{\max} - R_{\min}))$$

$$s_2 : [R_{\min} + \frac{1}{k}(R_{\max} - R_{\min}), R_{\min} + \frac{2}{k}(R_{\max} - R_{\min}))$$

.....

$$s_{k-1} : [R_{\min} + \frac{k-2}{k}(R_{\max} - R_{\min}), R_{\min} + \frac{k-1}{k}(R_{\max} - R_{\min}))$$

$$s_k : [R_{\min} + \frac{k-1}{k}(R_{\max} - R_{\min}), R_{\min} + \frac{k}{k}(R_{\max} - R_{\min})]$$

Since the data distribution is initiall, some of the radius-vectors might be inappropriately partitioned. The mean values for each two neighboring clusters c_j and $c_{j+1}, 1 \le j \le k-1$, are calculated according (1) , where $|c_j|$ is the number of elements in cluster c_j. A radius-vector $R_i \in c_j$, for which $|R_i - mc_{j+1}| < |R_i - mc_j|$ is satisfied, is moved from cluster c_j in cluster c_{j+1}. Thus radius-vector $R_i \in c_{j+1}$, for which $|R_i - mc_j| < |R_i - mc_{j+1}|$ is satisfied, is moved from cluster c_{j+1} in cluster c_j. When a radius-vector is moved from one cluster into another, clusters' structure and clusters' mean values are changed, recalculating clusters' new mean values mc_j and mc_{j+1}. 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 (i, R_i) information, each radius-vector R_i is transformed into object o_i, $1 \le i \le n$. Thus clusters of radius-vectors $c_j, 1 \le j \le k$, are transformed into object clusters $oc_j, 1 \le j \le k$, having each object $o_i, 1 \le i \le n$ from the object set O properly partitioned in object cluster $oc_j, 1 \le j \le k$.

$$mc_j = \frac{\sum R_i \in c_j}{|c_j|}, 1 \le j \le k \quad (1)$$

4. Algorithm

Algorithm 1 Improved IC: Intelligent Clustering

Input: set of objects $O=\{o_1,o_2,\ldots,o_{n-1},o_n\}$
Output: k clusters of objects oc_j, 1<=j<=k

for each object o_i which belongs to the object set O{
calculate its radius-vector R_i;
store data pair (i,R_i) in the memory;
}
find the smallest radius-vector $R_{min}=\min\{R_1,R_2,\ldots,R_{n-1},R_n\}$;
find the largest radius-vector $R_{max}=\max\{R_1,R_2,\ldots,R_{n-1},R_n\}$;
determine sub-intervals s_j, 1<=j<=k;
i=1;
j=1;
while(i<=n){
while(j<=k){
if(R_i belongs to sub-interval s_j){
add R_i in cluster c_j;
break *while(j<=k)* loop;
}
j++;
}
i++;
}
calculate centers of clusters mc_j, 1<=j<=k*;*
LOOP: j=1;
while(j<=k-1){
for each R_i which belongs to cluster c_j
if ($|R_i-mc_{j+1}|<|R_i-mc_j|$){
move R_i from cluster c_j in cluster c_{j+1};
calculate clusters' new mean values mc_j and mc_{j+1};
}
for each R_i which belongs to cluster c_{j+1}
if ($|R_i-mc_j|<|R_i-mc_{j+1}|$){

```
    move Ri from cluster cj+1 in cluster cj;
    calculate clusters' new mean values mcj and mcj+1;
    }
    j++;
    }
go to LOOP while at least one mcj is changing;
transform radius-vector clusters cj into object clusters ocj, 1<=j<=k;
```

5. An Example

Set of objects $O=\{(3,4),(5.7,5.9),(6,5.7),(6.1,5.8),(5.8,5.9),(4.5,4.9),(4.6,5),(7,7),(4,4),(8,6)\}$ should be partitioned in three clusters. According to the methodology being presented, for each object at position i a radius-vector $R_i, 1\le i\le 10$ is calculated, Table 1. Memory keeps ten data pairs $(i,R_i), 1\le i\le 10$, 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 (i,R_i)

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, $R_{\min} = 5, R_{\max} = 10$ intervals s_1, s_2 and s_3 can be determined.

$$s_1 : \left[5, 5 + 1 \times \frac{(10-5)}{3}\right) = \left[5, \frac{20}{3}\right) = [5, 6.667)$$

$$s_2 : \left[\frac{20}{3}, 5 + 2 \times \frac{(10-5)}{3}\right) = \left[\frac{20}{2}, \frac{25}{3}\right) = [6.667, 8.333)$$

$$s_3 : \left[\frac{25}{3}, 5 + 3 \times \frac{(10-5)}{3}\right] = \left[\frac{25}{3}, \frac{30}{3}\right] = [8.333, 10]$$

Distributing radius-vector $R_i, 1 \le i \le 10$ in cluster $c_j, 1 \le j \le 3$ is permitted, only if R_i belongs to the interval $s_j, 1 \le j \le 3$.

Cluster c_1: {5,6.653,5.657}, mean value $mc_1 = \frac{17.31}{3} = 5.77$

Cluster c_2: {8.204,8.276,8.273,6.794}, mean value $mc_2 = \frac{31.547}{4} = 7.887$

Cluster c_3: {8.417,9.899,10}, mean value $mc_3 = \frac{28{,}316}{3} = 9.439$

A check for radius-vectors $R_i \in c_1$, being cluster c_2 less distanced than cluster c_1, is conducted, Table 3.

Table 3 Calculating the distances between cluster c_1 radius-vectors and cluster c_1 and c_2 mean values

Radius-vector	Distance from cluster c_1	Distance from cluster c_2
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 c_1 radius-vector, being closer to cluster c_2 than cluster c_1, what indicates appropriate radius-vector distribution in cluster c_1.

A check for radius-vectors $R_i \in c_2$, being closer to cluster c_1 than cluster c_2, has also to be conducted, Table 4.

Table 4 Calculating the distances between cluster c_2 radius-vectors and cluster c_1 and c_2 mean values

Radius-vector	Distance from cluster c_2	Distance from cluster c_1
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 c_1 less distanced than cluster c_2, where was initially distributed. In this case, radius-vector 6.794 is moved from cluster c_2 in cluster c_1. Since cluster c_1 and cluster c_2 structure has been changed, cluster c_1 and cluster c_2 new mean values are calculated.

Cluster c_1 : {5,6.653,5.657,6.794}, mean value $mc_1 = \frac{24.104}{4} = 6.026$

Cluster c_2 : {8.204,8.276,8.273}, mean value $mc_2 = \frac{24.753}{3} = 8.251$

Cluster c_3 : {8.417,9.899,10}, mean value $mc_3 = \frac{28,316}{3} = 9.439$

Distance results between cluster c_2 radius-vectors and cluster c_3 and cluster c_2 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 c_2 radius-vector being closer to cluster c_3 than cluster c_2, where from can be concluded that cluster c_2 radius-vectors are properly partitioned.

At the end has to be checked whether exist cluster c_3 radius-vectors being cluster c_2 less distanced than cluster c_3, Table 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 c_3 radius-vector 8.417 is cluster c_2 less distanced than cluster c_3 , resulting with rearrangement of radius-vector 8.417, being moved from cluster c_3 in cluster c_2. Since cluster c_2 and cluster c_3 structure is changed, clusters' new mean values mc_2 and mc_3 are calculated.

Cluster c_1 : {5,6.653,5.657,6.794}, mean value $mc_1 = \frac{24.104}{4} = 6.026$

Cluster c_2 : {8.204,8.276,8.273,8.417}, mean value $mc_2 = \frac{33.17}{4} = 8.293$

Cluster c_3 : {9.899,10}, mean value $mc_3 = \frac{19,899}{2} = 9.950$

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 $(i, R_i), 1 \le i \le 10$, each radius-vector is transformed into object from the object set O. Thus radius-vector clusters are transformed into object clusters, having all objects properly partitioned.

Object cluster oc_1 : {(3,4),(4.5,4.9),(4,4),(4.6,5)}
Object cluster oc_2 : {(5.7,5.9),(6,5.7),(5.8,5.9),(6.1,5.8)}
Object cluster oc_3 : {(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.