INFO-951
Primljeno/Received 2006-03-17

UDK: 004.65
Original Scientific Paper/Izvorni znanstveni rad

SOFTWARE SOLUTIONS IN MARKETING RESEARCH FOR KNOWLEDGE DISCOVERY IN DATABASES BY FUZZY CLUSTERING

SOFTVERSKA RJEŠENJA U MARKETING ISTRAŽIVANJU ZA OTKRIVANJE ZNANJA U BAZAMA PODATAKA POMOĆU FUZZY KLASTERIRANJA

Brano Markic, Dražena Tomić
Faculty of Economics, University of Mostar, Mostar, Bosnia and Herzegovina
Ekonomski fakultet, Sveučilište u Mostraru, Mostar, Bosna i Hercegovina

Abstract

Knowledge discovery in databases is the process of identifying novel, valid, useful and ultimately understandable patterns in data stored in databases. Data mining is only a step in this process in charge to find patterns or models in data. There are many data mining algorithms for clustering. Clustering is unsupervised classification, the process of grouping the data into classes so that the data objects (examples) are similar to one another within the same cluster and dissimilar to the objects in other clusters. In the paper is developed a conceptual model and program solution for clustering data stored in subject oriented data warehouse. Data warehouse and mining algorithms are integrated and this integration has shown satisfactory implementation power.

Sažetak

Otkrivanje znanja u bazama podataka je proces identificiranja novih, validnih, korisnih i razumljivih paterna i modela iz podataka pohranjenih u bazama podataka. Data mining je samo jedan korak u tom procesu, a on ima zadatak pronaći i otkriti paterne i modele. Postoji više data mining algoritama za klasteriranje. Klasteriranje pripada nenadziranom učenju (unsupervised learning), a ono je postupak grupiranja podataka u klase tako da je sličnost najveća između podataka u jednoj klasi a razlika što veća u odnosu na podatke u drugoj klasi. U radu je razvijen konceptualni model integracije i odgovarajuće programsko rješenje, klasteriranja podataka pohranjenih u skaldištu podataka. Rješenje je u marketing funkcijskom području a integracija skladišta podataka i data mining algoritma pokazuje zadovoljavajuću implementacijsku snagu.

1. INTRODUCTION

Knowledge discovery in databases is an interdisciplinary field that relates to databases, data warehouse, machine learning, expert systems (formalisms of knowledge representation), statistics and operational research and data visualization. The common goal of integrating these different fields is extracting knowledge from data stored in large databases and data warehouse. For us is especially interesting the overlap knowledge data discovery and the area of data warehouse. The most popular approach for analysis of data warehouses is OLAP (on-line analytical processing) and is focusing on providing multi-dimensional data analysis. Knowledge discovery and OLAP are complementary and may be integrated with other technologies such as expert systems and provide a new generation of intelligent information extraction and management tools.

2. KNOWLEDGE DISCOVERY IN DATABASES FOR MARKETING RESEARCH

The process of knowledge discovery inherently consists of several steps [1]. The first step is to understand the application domain and to formulate the problem. Our application domain is clustering customers on business markets. This step determines the appropriate data mining algorithms.

The second step is to collect and preprocess the data. Corresponding data model is physically implemented by relational database. This database is known as operational. From operational database in a process of extracting, transforming and loading data is populated the data warehouse. This step is one of the most tedious and usually takes the most time needed for the whole knowledge data discovery process.

Data for customers clustering are stored in relational data warehouse that is temporarily loading from transactional data bases. The fact table in data warehouse named as Customers_Cluster includes three measures: sales, profit and days of payments.

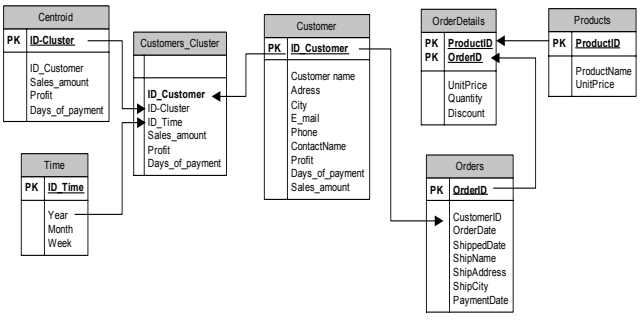

Fig.1.: Data warehouse for customers clustering

The third step is data mining that extracts patterns or models hidden in data. A model can be viewed "a global representation of a structure" and "a pattern is a local structure, perhaps relating to just a handful of variables and a few cases". In our example the third step is implementing fuzzy c-means clustering in marketing that extracts patterns hidden in data (adequate centroids of clusters).

The last step is to interpret discovered knowledge and put knowledge in practical use. All these steps are "covered" with adequate software technologies and programs. In the paper are shown the most important and critical steps in building the software solution of knowledge discovery.

3. SOFTWARE SOLUTION OF CLUSTERING IN MARKETING FUNCTIONAL FIELD

The software solution for market segmentation uses the fuzzy c-means algorithm. Clustering is the process of grouping the data into classes (clusters) so that the data objects (examples) are similar to one another within the same cluster and dissimilar to the objects in other clusters. A *good clustering* method will produce high quality clusters with high *intra-class* similarity and low *inter-class* similarity.

Hard k-means algorithm executes a sharp clustering, in which each object is either assigned to a cluster or not.

The k-means algorithm partitions a set of N vector into **c** clusters (clusters G_i, i=1,..,c). The goal is finding cluster centers (centroids) for each cluster. The algorithm minimizes a dissimilarity (or distance) function which is given in Equation 1.

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{k, x_k \in G_i} d(x_k - c_i) \qquad (1)$$

c_i is the centroid of cluster i;

$d(x_k - c_i)$ is the distance between i_{th} centroid(c_i) and k_{th} data point;

Overall dissimilarity function is expressed as in Equation 2

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, x_k \in G_i} | x_k - c_i |^2 \right) \qquad (2)$$

Partitioned groups can be defined by a **binary** membership matrix(m), where the element m_{ij} is 1 if the j_{th} data point x_j belongs to cluster i, and 0 otherwise (Equation 3)

$$m_{ij} = \begin{cases} 1 & \text{if } \| x_j - c_i \|^2 \le \| x_j - c_k \|^2, \text{ for each } k \ne i, \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Since a record can only be in a cluster, the membership matrix (m) has two conditions which are given equation 4 and equation 5.

$$\sum_{i=1}^{c} m_{ij} = 1 \qquad (4)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{c} m_{ij} = n \qquad (5)$$

Centroids are computed as the mean of all vectors in group i:

$$c_i = \frac{1}{| G_i |} \sum_{k, x_k \in G_i} x_k \qquad (6)$$

$|G_i|$ is the size of G_i.

The software solution for k-means algorithm and all necessary steps [2] no guarantee for an optimum solution steps with a recordset x_j, j=1,..n. The performance of the algorithm depends on the initial positions of centroids.

Fuzzy clustering allows that one tuple belongs at the same time to several clusters but with different degrees. This is an important feature for segmentation business markets to increase the sensitivity.

Fuzzy c-means clustering is a clustering technique which is separated from hard k-means that employs hard partitioning. The employs fuzzy partitioning such that a tuple (fact table record in data warehouse) can belong to all groups with different membership grades between 0 and 1. Fuzzy c-means is an iterative algorithm. The aim of fuzzy c-means is to find cluster centers (centroids) that minimize a dissimilarity function.

1. The first step is randomly generating the clusters. The clusters are chosen from fact table DataKlasterF and the next source code written in Visual Basic shows how is randomly selecting these seed clusters.

Random generated starting centroid

ID_Customer	Amount of sale	Profit	Days	ID_Cluster
26	647	65	45	1
68	21212	2345	45	2
11	34432	7890	45	3
54	34523	4321	55	4
1	234	24	30	5
11	34432	7890	45	6

2.The second step is calculating the membership matrix(U) according to Equation 7

$$\sum_{j=1}^{c} m_{ij} = 1 \tag{7}$$

The dissimilarity function which is used in fuzzy c-means clustering is given Equation 8

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{n} J_i \sum_{j=1}^{c} J_j = \sum_{i=1}^{n} \sum_{j=1}^{c} m_j^t d_j^2 \tag{8}$$

m_{ij} is between 0 and 1; c_i is the centroid of cluster i; d_{ij} is the Euclidian distance between i_{th} centroid(c_i) and j_{th} data point; t ϵ [1,∞] is a weighting exponent.

To reach a minimum of dissimilarity function there are two conditions. These are given in Equation 9 and Equation 10.

$$c_i = \frac{\sum_{i=1}^{n} u_j^t X_i}{\sum_{i=1}^{n} u_j^t} \tag{9}$$

$$m(i,j) = \frac{1}{\sum_{k=1}^{c} (\frac{d_{ij}}{d_k})^2} \tag{10}$$

The next code written in Visual Basic implements the equation 10 (membership matrix m(i,j)):

```
rsDataKlasterF.MoveFirst ' rsDataKlasterF is a recordset
 ReDim Preserve m(rc, Val(Trim$(Text1.Text)))
 ReDim Preserve a(Val(Trim$(Text1.Text)))
 For i = 1 To rc
X = rsDataKlasterF!promet, Y = rsDataKlasterF!ruc,
Z = rsDataKlasterF!dop
rsCentroidF.MoveFirst
                For j = 1 To Val(Trim$(Text1.Text))
                    xc = rsCentroidF!promet
                    yc = rsCentroidF!ruc
                    zc = rsCentroidF!dop
                  a(j) = Round(Round((X - xc) ^ 2, 0) +
                  Round((Y - yc) ^ 2, 0) + (Z - zc) ^ 2,
                  0)
                  rsCentroidF.MoveNext
            Next j
          Dim S As Single
          Dim k As Integer
        For j = 1 To Val(Trim$(Text1.Text))
          S = 0
                  For k = 1 To Val(Trim$(Text1.Text))
                      If a(k) = 0 Then
                                m(i, k) = 1
                                S = 0
                                Exit For
                       Else
                                S = S + Round(a(j) /
a(k), 5)
                      End If
                  Next k
                  If S <> 0 Then
                        m(i, j) = Round(1 / S, 5)
                  End If
            Next j
              rsDataKlasterF.MoveNext
 Next i
```

The membership matrix m(i,j) for the first fourteen records is presented by Fig. 2:

Fuzzy membership

Record	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
1	0	0	0	0	1	0
2	0,01012	0,85968	0,03838	0,04366	0,00977	0,03838
3	0,71655	0,00086	0,0003	0,00031	0,28167	0,0003
4	0,74004	0,00066	0,00023	0,00024	0,2586	0,00023
5	0,04343	0,00003	0,00001	0,00001	0,95652	0,00001
6	0,48983	0,05004	0,01392	0,0145	0,41779	0,01392
7	0,13511	0,55388	0,06007	0,06346	0,1274	0,06007
8	0,95781	0,00003	0,00001	0,00001	0,04214	0,00001
9	0,87521	0,00013	0,00005	0,00005	0,12452	0,00005
10	0,42055	0,11761	0,02827	0,02958	0,37572	0,02827
11	0	0	1	0	0	0
12	0,01647	0,71461	0,08159	0,08983	0,01592	0,08159
13	0,83852	0,00021	0,00007	0,00008	0,16105	0,00007
14	0,73323	0,00071	0,00025	0,00026	0,26531	0,00025
15	0,49961	0,04237	0,01206	0,01252	0,42138	0,01206
16	0,08218	0,14368	0,23142	0,22992	0,08138	0,23142
17	0,46721	0,06979	0,01854	0,01931	0,4066	0,01854
18	0,46776	0,06922	0,01844	0,01918	0,40695	0,01844

Fig 2.: Membership matrix m(i,j)

The membership matrix calculating for six randomly chosen clusters as centroids satisfy constrains

$$\sum_{j=1}^{n} m(i,j) = 1 \quad \forall j = 1,2,...n. \tag{11}$$

3. The third step is calculating the center of centroids (new centroids of clusters) using membership matrix m(i,j) and values for profits, sales and days of payments:

$$c_j = \frac{\sum_{i=1}^{n} m(i,j)^2 * X_i}{\sum_{i=1}^{n} m(i,j)^2} \tag{12}$$

m(i,j) is between 0 and 1; c_j is the centroid of cluster j;

After calculating the new centroid it is necessary to calculate the distance between the cluster center and each one records. Stop if its improvement over previous iteration is below a threshold. Stop condition in this example is defined by statement:

If Abs(pC(j, 1) - rsCentroidF!promet) < 60 And Abs(rucC(j, 2) –

rsCentroidF!ruc) < 6 And Abs(dopC(j, 3) - rsCentroidF!dop) < 1 Then

```
                nastavi = False
            Else
                nastavi = True
            End If
```

or generally do while $\sum_{j=1}^{c} \left\| c_j^{Previous} - c_j \right\| > \varepsilon$.

The next code implements calculating new centroids, test the conditions and if the stop condition is not satisfied calculate the new membership matrix.

```
   Dim Brojnik() As Single, Nazivnik()
   Dim nastavi As Boolean
   Dim Xr() As Single
   ReDim Xr(Val(Trim(Text1.Text)), 3)
   ReDim Brojnik(Val(Trim(Text1.Text)), 3)
   ReDim Nazivnik(Val(Trim(Text1.Text)), 3)
    Dim p As Single, ruc, dop
    Dim pC() As Single
    Dim rucC() As Single
    Dim dopC() As Single
   nastavi = True
   Do While nastavi
   rsCentroidF.MoveFirst
   For j = 1 To Val(Trim(Text1.Text))
   Brojnik(j, 1) = 0,  Nazivnik(j, 1) = 0, Brojnik(j, 2) = 0, Nazivnik(j, 2) = 0
   Brojnik(j, 3) = 0,  Nazivnik(j, 3) = 0
   rsDataKlasterF.MoveFirst
         For i = 1 To rc
                  Xr(j, 1) = rsDataKlasterF!promet, Xr(j, 2) = rsDataKlasterF!ruc
                  Xr(j, 3) = rsDataKlasterF!dop
                  Brojnik(j, 1) = Brojnik(j, 1) + Round(m(i, j) ^ 2 * Xr(j, 1), 2)
                   Nazivnik(j, 1) = Round(Nazivnik(j, 1) + m(i, j) ^ 2, 3)
                   Brojnik(j, 2) = Round(Brojnik(j, 2) + m(i, j) ^ 2 * Xr(j, 2), 3)
                   Nazivnik(j, 2) = Round(Nazivnik(j, 2) + m(i, j) ^ 2, 3)
                   Brojnik(j, 3) = Round(Brojnik(j, 3) + m(i, j) ^ 2 * Xr(j, 3), 3)
                   Nazivnik(j, 3) = Round(Nazivnik(j, 3) + m(i, j) ^ 2, 3)
                   rsDataKlasterF.MoveNext
          Next i
          ReDim Preserve pC(Val(Trim(Text1.Text)), 3)
           ReDim Preserve rucC(Val(Trim(Text1.Text)), 3)
          ReDim Preserve dopC(Val(Trim(Text1.Text)), 3)
           pC(j, 1) = rsCentroidF!promet,  rucC(j, 2) = rsCentroidF!ruc
          dopC(j, 3) = rsCentroidF!dop
          p = Round(Brojnik(j, 1) / Nazivnik(j, 1), 3)
          ruc = Round(Brojnik(j, 2) / Nazivnik(j, 2), 3)
          dop = Round(Brojnik(j, 3) / Nazivnik(j, 3), 2)
           rsCentroidF!promet = p, rsCentroidF!ruc = ruc,  rsCentroidF!dop = dop
           rsCentroidF.MoveNext
      Next j
      rsCentroidF.MoveFirst
          For j = 1 To Val(Trim(Text1.Text))

   If Abs(pC(j, 1) - rsCentroidF!promet) < 60 And Abs(rucC(j, 2) –
   rsCentroidF!ruc) < 6 And  Abs(dopC(j, 3) - rsCentroidF!dop) < 1 Then
                  nastavi = False
            Else
                  nastavi = True
            End If
            rsCentroidF.MoveNext
      Next j

    rsDataKlasterF.MoveFirst
   For i = 1 To rc
      X = rsDataKlasterF!promet
      Y = rsDataKlasterF!ruc
      Z = rsDataKlasterF!dop
      rsCentroidF.MoveFirst
         For j = 1 To Val(Trim$(Text1.Text))
               xc = rsCentroidF!promet,  yc = rsCentroidF!ruc,  zc = rsCentroidF!dop
            a(j) = Round(Round((X - xc) ^ 2, 0) + Round((Y - yc) ^ 2, 0) + (Z - zc) ^ 2, 0)
            rsCentroidF.MoveNext
      Next j
       For j = 1 To Val(Trim$(Text1.Text))
            S = 0
            For k = 1 To Val(Trim$(Text1.Text))
               If a(k) = 0 Then
                  m(i, k) = 1
                   S = 0
                  Exit For
               Else
                  S = S + Round(a(j) / a(k), 5)
                End If
            Next k
               If S <> 0 Then
                  m(i, j) = Round(1 / S, 5)
               End If
       Next j
   rsDataKlasterF.MoveNext
   Next i
  Loop
```

By iteratively updating the cluster centers and the membership matrix [3] for each record, fuzzy c-means iteratively moves the cluster centers to the "right center" within data records.

Cluster centers (centroids) are initializing using randomly selecting records and fuzzy c-means does not ensure that it converges to an optimal solution. Performance depends on initial centroids and may be improved on two ways:

1) Using an algorithm to determine all of the centroids. (for example: arithmetic means of all records)

2) Run fuzzy c-means several times each starting with different initial centroids.

Software agents for market segmentation preferred the second approach.

4. EXPERIMENTAL RESULTS

Software agent for segmentation business markets assigns the customers to different clusters, different market segments. After final segmentation of business markets to six segments and assign each one customer two segments follows six clusters with centroids and corresponding membership grades:

New centroid

	Revenues	Profit	Days	ID_Cluster
	2446,555	324,259	50	1
	31110,71	4248,482	49	2
	74844,66	8067,709	48	3
	51084,19	5725,954	48	4
	9588,406	981,328	54	5
	99882,76	11108,45	51	6

Fuzzy membership

Record	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
1	0,93913	0,00482	0,00083	0,00179	0,05296	0,00047
2	0,08529	0,6497	0,01446	0,05007	0,19395	0,00652
3	0,97924	0,00141	0,00023	0,00051	0,01848	0,00013
4	0,97659	0,00161	0,00027	0,00058	0,0208	0,00015
5	0,9347	0,00525	0,00091	0,00195	0,05669	0,00051
6	0,59863	0,00925	0,00127	0,00293	0,38724	0,00068
7	0,1312	0,06722	0,00518	0,01406	0,77975	0,00259
8	0,95952	0,00299	0,00051	0,00109	0,0356	0,00028
9	0,96475	0,00255	0,00043	0,00093	0,0311	0,00024
10	0,10449	0,00536	0,00066	0,00157	0,88758	0,00035
11	0,01929	0,85785	0,01276	0,0739	0,03134	0,00485
12	0,06861	0,71669	0,01311	0,04673	0,14902	0,00585
13	0,96743	0,00233	0,00039	0,00085	0,02878	0,00022
14	0,97764	0,00153	0,00025	0,00055	0,01988	0,00014
15	0,68871	0,00822	0,00114	0,00263	0,29867	0,00062
16	0,00937	0,02206	0,61285	0,05552	0,01123	0,28896
17	0,39372	0,00984	0,0013	0,00304	0,59141	0,00069
18	0,40029	0,0099	0,0013	0,00305	0,58475	0,0007

Fig 3. Clusters centroids and corresponding membership grades

These software solution shows that the first customer (record 1) 93.913% belongs to market segment where the average amount of sale is 2446.555; average number of payments day is 50 and realized profit per customer 324.259.

One customer can belong to several market segments at the same time but with different degrees. This is an important feature for market segmentation to increase the sensitivity.

For the membership degrees close to 0.5 are the suspicious cases. Assigning the sample to one cluster could be wrong. Therefore software gives very reliable results.

Suspicious records

ID_Customer	Cluster	Membership
6	1	0,59863
17	5	0,59141
18	1	0,40029
18	5	0,58475
50	1	0,561
50	5	0,42444
67	2	0,45649
86	5	0,53937
88	2	0,46036
89	3	0,47268
89	6	0,42825
91	3	0,52268

Fig. 4.: Membership degrees of suspicious samples

Now is necessary the expert judgment to assign the customers 6, 17, 83, 18, 50, 67, 86, 88, 89 and 91 to adequate market segment. Namely, the software solution extracts the customers for which the membership grade is between 0.4 and 0.6. This interval may be closer.

5. CONCLUSION

These paper theoretically and practically presented the integration knowledge discovery in databases, data mining and marketing research. The data mining component of the knowledge discovery process is mainly concerned with algorithms by which patterns are extracted from the data (fuzzy c-means clustering). Data for customers clustering are stored in relational data warehouse that is temporarily loading from transactional data bases. After running the program for fuzzy c-means clustering written in Visual Basic development environment follow the results that is easy understand and explain. Finally step is finding out suspicious records, customers for which the membership degree is close to 0.5. Expert judgment is necessary to analyze this tuples. On the business market the firm may very easy to define the required number of clusters (five, ten, twenty etc.) and the software will assign the customers to adequate cluster with membership degree. So has been assured the sensitivity and broad applicability the software and the concept of knowledge discovery.

Literature

1. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, S., and Uthurusamy, R.: (1996), *Advances in Knowledge Discovery and Data Mining*, M.I.T. Press.
2. G. Berks, D.G. Keyserlingk, J. Jantzen, M. Dotoli, H. Axer, *Fuzzy Clustering- A Versatile Mean to Explore Medical Database*, ESIT2000, Aachen, Germany
3. J.-S. R. Jang, C.-T. Sun, E.Mizutani, *Neuro-Fuzzy and Soft Computing*, p (426-427) Prentice Hall, 1997.
4. Markić, B., Tomić, D. (2005),*Executive information system for customers clustering*, Društvo i tehnologija, Međunarodni simpozij, Hrvatska, Zadar.
5. Markić, B., Tomić, D. (2006), *Building software agents for market segementation*, Baden Baden.