ISSN 2277-3061

A two level approach to discretize cosmetic data using Rough set theory

P.M. Prasuna, Dr.Y. Ramadevi, Dr. A.Vinay Babu
Research Scholar JNTU, Hyderabad
prasunamanikya@yahoo.com
Dr.Y. Ramadevi, Professor CBIT, Hyderabad
yrdcse.cbit@gmail.com
Dr. A.Vinay Babu, Professor JNTUHCE, Hyderabad
avb1222@jntuh.ac.in

ABSTRACT

Discrete values play a very prominent role in extracting knowledge. Most of the machine learning algorithms use discrete values. It is also observed that the rules discovered through discrete values are shorter and precise. The predictive accuracy is more when discrete values are used. Cosmetic industry extracts the features from the face images of the customers to analyze their facial skin problems. These values are continuous in nature. A predictive model with high accuracy is required to determine the cosmetic problems of the customers and suggest suitable cosmetic. Existing traditional discretization techniques are not sufficient for deriving discretized data from continuous valued cosmetic data as it has to balance the loss of information intrinsic to process adapted and generating a reasonable number of cut points, that is, a reasonable search space. This paper proposes a two level discretization method which is a combination of traditional k means clustering technique and rough set theory to discretize continuous features of cosmetic data.

Indexing terms/Keywords

Rough set Theory, Discretization, cut points, kmeans.

Academic Discipline And Sub-Disciplines

Data Mining and Retrieval

SUBJECT CLASSIFICATION

Discretization technique

TYPE (METHOD/APPROACH)

Rough set theory

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol. 14, No. 10
www.ijctonline.com , editorijctonline@gmail.com

ISSN 2277-3061

INTRODUCTION

There are huge volumes of data in the cosmetic industry not only to analyze the problems of the customers but also to rejuvenate a new product basing on the customer problems. Data mining algorithms help us to extract necessary information for decision making from this cosmetic data. However, many mining algorithms or machine learning algorithms cannot be applied on them as they are continuous in nature. Numeric data contain large number of values when compared to discrete values, the rules discovered looks complex and gives less predictive accuracy. As discrete attributes are represented with simple interval numbers they are understandable and easier to use. The rules of discrete attributes usually are shorter and easy to understand, hence will increase the accurateness of predictions. Therefore, it is essential to have good descretization techniques [1] to transform continuous valued features into discrete valued features. This not only speeds up the mining process but also helps in developing a better model. This paper deals with a two level discretization technique for cosmetic data which firstly uses the traditional kmeans algorithm and then applies rough set theory to discrete the data at attribute level.

K means algorithm

Kmeans algorithm: Kmeans is a simple unsupervised clustering technique [2]. It follows simple and easy steps to form the clusters. Initially number of clusters to be formed is to be determined. Then it follows three steps, initialization, expectation and maximization. In initialization step k centers are created where k is the number of clusters to be formed which is predetermined. In expectation step each data point is assigned to the center closest to it and maximization step deals with computation of new center basing on the data points associated to it. These steps are carried out repeatedly until no more changes are done to centers.Finally, this clustering technique aims at minimizing an objective function, in this case a squared error function. The objective function used is

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j is an indicator of the distance of the data points from their respective cluster centres [3].

Kmeans Algorithm:

Make initial guesses for the centres c1, c2, ..., ck

- Until there are no changes in any centre
  - Use the estimated means to classify the samples into clusters
  - For i from 1 to k
    - Replace mi with the mean of all of the samples for cluster i
  - end_for
- end_until

Application of k means algorithm to cosmetic data discretization

Initially kmeans algorithm is applied on sample cosmetic data to form the clusters as it is unsupervised [4]. This completes the basic discretization step. This step discretizes the data into specified number of intervals .The results are then given to the second phase which uses Rough set theory[5]

Rough Set Theory

Rough set theory was proposed by Professor Powlak (powlak, 1982:1991 skowron, 1990) [6]. The main goal of the rough set analysis is induction of (learning) approximations of concepts. It offers mathematical tools to discover patterns hidden in data. The basic concepts of rough set theory are described below:

Approximation Space: An approximation space is a pair (U, B) where U is a nonempty finite set called the universe and B is an equivalence relation defined on U.

Information System: An information system is a pair S= (U, A), where U is a nonempty finite set called the universe and A is a nonempty finite set of attributes, i.e., a: U→Va for aєA, where Va is called the domain of a.

Decision Table (Data Table): A decision table is a special case of information system, S = (U, A= C ϵ {d}), where attributes in C are called condition attributes and d is a designated attribute called the decision attribute.

Approximations of Sets: Let S = (U, B) be an approximation space and X be a subset of U.

The lower approximation of X in S is defined as

$$\underline{B}X = \{x \in X: [x]\ B \epsilon X\}$$

The upper approximation of X in S is defined as

$$\overline{B}X = \{x \in X: [x]\ B \cap X \neq \varphi\}$$

For a given set of conditional attributes B, the B- positive region POSB(D) in the relation IND(D) is defined as, POSB(D) = ϵ{BX : X ϵ [x]D } . The positive POSB(D) region contains all the objects in U that can be classified without any error into distinct classes defined by IND (D), based only on information in the relation IND (B). Greater the cardinality POSB(D) higher the significance of the attributes in the set B with respect to D.

The rough membership function quantifies the degree of relative overlap between X and the equivalence class to which x belongs. Thus this rough membership function is also a measure of the significance of B⊆A to describe X and is defined by [7],

$$\mu_{X(X)}^{B} = \frac{card\,(X \cap [X]IND(B))}{card\,(IND(B))}$$

Application of rough set theory to refine the cut points generated by Kmeans algorithm

The traits of the clusters formed by the kmeans algorithm vary. This discretization using clustering technique is not sufficient to generate cut points with minimum information loss. Hence they are refined using Rough set theory concepts [8]. The main aim in splitting the cluster is to refine the discretized interval. The refinement is to enhance the significance of the attribute. In rough set theory the significance of an attribute is measured through rough membership function POSai(D).Hence maximizing POSai(D) leads to maximizing the significance of the attribute. To maximize POSai (D), the clusters formed through kmeans are refined further to generate new intervals or cut points. The refinement is processed in such a way that the maximum number of objects is correctly classified by each of the interval of attribute ai, just as they are classified by D [9]. This is done by a rough membership function applied to each interval of the attribute ai with respect to the clusters formed through the kmeans which are further treated as class labels.

Let us take the data set U contains objects of m clusters say {c1, c2, c3 … cm} and let the k distinct values of an attribute ai in ascending order be {vi1, vi2, vi3 … vik} i.e. the interval [vi1, vik]. The rough membership function of any interval

I= [Vi1, Vij] of the attribute ai for class cp is defined as

$$f(ai, cp, I) = \frac{card(\underline{a_{i,I}} X_{cp})}{card(X_{a_{i,I}})}$$

where $X_{a_{i,I}}$ ={x | ai(x) |ϵ I} and, $\underline{a_{i,I}}\ X_{cp}$ ={x | ai(x) ϵ I, D(x) =cp}.

Maximizing f(ai,cp,I) is maximising $card(\underline{a_{i,I}} X_{cp})$ which further maximizes POSai(D) [10].To achieve this each cluster generated by kmeans is examined carefully and if necessary a cluster may be split into two or merged with the neighbouring cluster. The splitting process uses the rough set membership function such that it maximizes the POSai (D).in this way the intervals are refined. The refinement takes place as follows. Initially three predetermined parameters are taken. Max_size determines the maximum no of values that could fall in each cluster. Min_size decides the minimum number of values to form a cluster and Range gives the length of the cluster. These parameters decide whether the cluster can be retained or still to be refined. The refinement process takes place if the cluster is large or small. The cluster is said to be large if its cardinality is greater than the Max_size or the length is greater than the Range. A cluster is treated as small if its cardinality is less than the Min_size. If the cluster is large it is split into two or else small, merged with other small clusters thereby generating new cut points or intervals. This process is refined until there is no change in the cut points or intervals.

Algorithm for the proposed method

Step1: Consider each attribute in the data set, select distinct values and sort them.

Step2: Apply kmeans algorithm to form clusters.

Step3.From the generated clusters determine the class labels as well as intervals.

Step4.Refine the intervals and add new intervals to the interval set.

Refine (I1, I2, …. Ir))

While (no change in no. of intervals) do

For each interval Ij

If SP-C (Ij, Min_size, Range) = True then

```
                    Temp= Cut Point ({vj1, vj2, vj3 … vjk})
                    Replace the interval Ij with two intervals
                    Ij1 = [vj1, Temp] and Ij2 = [Temp, vjk]
            Else if | Ij | < Min_size then
            If for Ik′ either neighbour of Ij
            MR_C (Ij, Ik′, Max_size, Min_size) = True   then Merge Ij to an interval Ik′

            End if
            End if

        End for
End while
```

Cut Point ({vi1, vi2, vi3 … vik})

```
I = [vi1, vik/2]
MAXRMV= Max ({f (Ai, cp, I)}) ϵ cp,
for each vij  , j=k/2 to 2
        I = [vi1, vij]
        Temp = Max ({ f (A i , c p , I) }) ϵ cp;

        if Temp> MAXRMV then
                MAXRMV=Temp;
         else
                break;
        End if
        If (j < k/2) then
                return vij as cut point for the cluster
        else
                for each vij, j=k/2 to k-1
                        I = [vij, vij]
                        Temp = Max ({ f (A i , cp , I) }) ϵ cp;
                        if Temp> MAXRMV then
                                MAXRMV=Temp;

                        else
                                return vij as cut point for the cluster
                        End if
                End for
        for each vij, j=k/2 to k-1
                 I = [vij, vij]
                Temp = Max ({ f (Ai , cp , I) })ϵ cp;
                if Temp> MAXRMV then
```

MAXRMV=Temp;

else

return vij as cut point for the cluster

End if

End for

Results

Cosmetic data has been collected from the customers of different age groups. The facial images of the customers are captured under sophisticated environment and then the features are extracted. The features are numeric in nature. To analyse the data collected mining tools are applied. As a preprocessing step to mining process the numeric values are discretized. To show the experimental results a dataset of 33 samples taken which consists of 17 numeric Features. After applying the proposed algorithm the results are as shown in Table -1:

Table -1

Sno	Attribute	Type	Distinct values	Intervals
1	Stype	Numeric	5	3
2	Saa	Numeric	30	7
3	S_Count	Numeric	29	6
4	A_Spots	Numeric	12	4
5	Pimples	Numeric	13	4
6	Pastules	Numeric	3	3
7	Papules	Numeric	5	4
8	Cysts	Numeric	2	2
9	B_Visi	Numeric	23	4
10	A_count	Numeric	20	4
11	p_count	Numeric	33	8
12	V_pores	Numeric	33	4
13	E_lines	Numeric	31	6
14	F_Lines	Numeric	27	7
15	D_lines	Numeric	20	7
16	E_Wrinkles	Numeric	1	1
17	h_skin	Numeric	33	5

Complexity

Before we apply kmeans algorithm first distinct values are identified then they are sorted. For carrying out this process complexity is O(N log N) where N is the number of objects in the dataset. Kmeans is known to have the complexity $O(n^{k+1} \log n)$ which may be in worst situation for the above algorithm i.e. when the attribute values for each object are distinct. The complexity of the Refine function is bounded by k * N/2, where k is the number of intervals of an attribute and the running time of the function Cut_Point is bounded by N/2. If n is the number of attribute then the total complexity of the algorithm is bounded by,

n * (N log N + N log N + k * N /2 + N)

≈ n * (N log N)

The number of attributes n is normally small in comparison to N. The preprocessing of the dataset for selecting relevant attributes further reduces the value of n to be small compared to N. Therefore, the running time of the proposed algorithm for labeled data is bounded by N logN.

Conclusion

By the proposed method the natural intervals of the values of the continuous attributes are obtained which maximized the mutual class-attribute interdependency. The method also generates the possibly minimum number of intervals.

Although the computational effort for the search algorithm for cut point has been reduced to half of N, the size of dataset, by implementing binary search for the cut point can further reduce the complexity of search step.

REFERENCES

[1] D Sotiris Kotsiantis, Dimitris Kanellopoulos “Discretization Techniques: A recent survey”GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 47-58

[2] Tapas Kanungo, , David M. Mount, , Nathan S. Netanyahu, ,Christine D. Piatko, Ruth Silverman, and Angela Y. Wu “An Efficient k-Means Clustering Algorithm: Analysis and Implementation” IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 7, JULY 2002

[3] James G. Booth, Ithaca, George Casella and James, P. Hobert “ Clustering using objective functions and stochastic search “J. R. Statist. Soc. B (2008) 70, Part 1, pp. 119–139

[4] Daniela Joiţa “ UNSUPERVISED STATIC DISCRETIZATION METHODS IN DATA MINING Titu Maiorescu University, Bucharest, Romania

[5] XU Chenggang “A Two-step Discretization Algorithm Based on Rough Set 2012 International Conference on Computer Science and Electronics Engineering

[6] Zbigniew Suraj “An Introduction to Rough Set Theory and Its Applications” ICENCO’2004, December 27-30, 2004, Cairo, Egypt.

[7] Frida Coaquira and Edgar Acuña “Applications of Rough Sets Theory in Data Preprocessing for Knowledge Discovery” Proceedings of the World Congress on Engineering and Computer Science 2007, San Francisco, USA

[8] Guan Xin, Yi Xiao, He You “Discretization Of Continuous Interval-Valued Attributes In Rough Set Theory And Its Application “ Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

[9] Nandita Sengupta “Evaluation of Rough Set Theory Based Network Traffic Data Classifier Using Different Discretization Method “ International Journal of Information and Electronics Engineering, Vol. 2, No. 3, May 2012.

[10] Girish Kumar Singh, Sonajharia Minz “Discretization Using Clustering and Rough Set Theory” Proceedings of the International Conference on Computing: Theory and Applications(ICCTA'07)0-7695-2770-1/07,2007Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[11] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[12] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender