Journal of Theoretical and Applied Information Technology
10th October 2014. Vol. 68 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OCCURRENCE BASED CATEGORICAL DATA CLUSTERING USING COSINE AND BINARY MATCHING SIMILARITY MEASURE

[1]S. ANITHA ELAVARASI1, [2] J. AKILANDESWARI

[1] Department of Computer Science and Engineering, Sona College of Technology, Salem, T N, India

[2] Department of Information Technology, Sona College of Technology, Salem, Tamil Nadu, India

E-mail: [1]anishaer@gmail.com , [2]akilandeswari@sonatech.ac.in

ABSTRACT

Clustering is the process of grouping a set of physical objects into classes of similar object. Objects in real world consist of both numerical and categorical data. Categorical data are not analyzed as numerical data because of the absence of inherit ordering. This paper describes about occurrence based categorical data clustering (OBCDC) technique based on cosine similarity measure and simple binary matching similarity measure. The OBCDC system consists of four modules, such as data pre-processing, similarity matrix generation, cluster formation and validation. Similarity matrix generation uses three functions, namely FrequencyComputation, OccurranceBasedCosine and OccurranceBasedSBMS. The time complexity of various algorithms are discussed and its performance on real world data are measured using accuracy and error rate

Keywords: *Clustering, Unsupervised Learning, Categorical Data, Cosine Similarity, Simple Binary Matching Similarity*

1. INTRODUCTION

Data mining is the process of extracting useful information from the given data set. Clustering is a process of grouping objects with similar properties [1]. Data in real world are either numerical or categorical in nature. Numerical data is continuous data with some ordering and Categorical data consist of a set of categories with no ordering exist between them. Categorical data are divided into Dichotomous and multi categorical data [2]. Dichotomous can have only two values. Multi categorical data can be in three ways, (1) an ordinal variable (ordered nature, eg. high low medium), (2) nominal variable (unordered in nature, eg. mode of transport preferred by persons) and (3) quantitative variable.

Categorical data are distributed in three ways: (1) binomial, (2) multinomial and (3) Poisson [3]. Applications of categorical data are: health care, education, biomedical, genetics and marketing fields. Categorical data are analyzed in four manners [4, 18], such as

1. Simple matching approach
2. Co-occurrence approach
3. Probabilistic approach
4. Distance hierarchy approach

Categorical data exhibit various similarity measures. Cosine similarity is a popular method for text mining. It is used for comparing the document and finds the closeness among the data points. Its range lies between 0 and 1. The simple binary matching similarity measure counts the number of attribute that matches the two data instance. The range of per attribute value is 0 to 1. Similarity measure for any clustering algorithm should exhibit 3 main properties such as,

1. Symmetry
2. Non Negative
3. Triangular Inequality

In this paper frequency based cosine similarity and simple binary matching similarity is used as an objective measure for clustering categorical data. The system architecture of OBCDC (Occurrence Based Categorical Data Clustering) algorithm and its performance is measured on real word dataset.

This paper is organized as related work on section 2. Section 3 describes the proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm, its system architecture, basic definitions and pseudo code. Section 4 describes the experimental result and finally section 5 concludes the work.

2. RELATED WORK

K-means algorithm is a well known partition clustering algorithm. It is efficient for processing larger data set, suitable for numerical data set and sensitive to outliers. The author extends the k means by using simple matching dissimilarity function suitable for categorical data [4]. Mode value is used instead of mean value and finally a frequency based method for updating the clustering process which reduces the cost function.

K modes algorithm produce only local optima. Time complexity is O(tkn) where 't' describes number of iteration, 'k' describes number of cluster and 'n' describes number of object. K-prototype [5] algorithm combines both K-means and K-modes for mixed numerical and categorical data. The author compares the performance and scalability of K-modes with K-prototype algorithm for soya bean dataset and motor insurance dataset.

Squeezer is a clustering algorithm for categorical data [6]. The data structures involved are Cluster Summary and Cluster Structure. Summary holds pair of attribute value and their corresponding support. Cluster Structure (CS) holds the cluster and summary information. The advantages of Squeezer algorithm are (1) It produces high quality clustering result (2) It deserves good scalability (3) It makes only one scan over the dataset, so it is highly efficient when considering I/O cost turn into bottleneck. The disadvantages of Squeezer algorithm is, quality of the cluster depends on the threshold value's'. Space complexity is O(n+k*p*m) and Time complexity is O(n*k*p*m). The value 'n' represent the size of the data set, 'm' represents number of attribute, 'k' represents the final number of cluster and 'p' represents the distinct attribute value.

ROCK stands for RObust Clustering using linKs [7]. It is a Agglomerative hierarchy clustering. It uses links to measure similarity between data point. Initially each tuple is assigned as a separate cluster. Clusters are merged based on the closeness between clusters. Closeness is measured as the sum of the number of links between all pair of tuple. It is suitable for boolean and categorical data. In traditional approach, categorical data are treated as boolean value. Scalability of the algorithm depends on the sample size. Time complexity of Rock is O(n2+nmmma+n2log n) where n represents no. of input data point , mm represents maximum no of neighbor, ma represents average no of neighbor.

QROCK [8] is a Quicker version of ROCK algorithm. Two major variation of QROCK from ROCK are: (1) Final clusters are formed as the connected components of certain graph. (2) Similarity threshold is specified instead of desired number of cluster. It uses three main ADT (1) merge (A, B): finds the union of A and B, (2) Initial(X): creates the component with element X, (3) find(X) : returns the component with element X. The overall complexity is QROCK is O(n2) where n is the number of data point.

CAtegorical ClusTering Using Summaries (CACTUS) is a fast summarization based algorithm [9]. It consist of three phase (1) Summarization: computes summary information, (2) Clustering: candidate cluster formation and (3) Validation: deals with identification of false candidate. It scans data set only twice and performs subspace clustering.

Fuzzy K-modes[10] is an extension to fuzzy K-means algorithm designed for categorical data. It generates fuzzy partition matrix which improves confidence degree of data in different cluster.

Data Intensive Similarity Measure for Categorical Data analysis (DISC) [11]. It makes use of a data structure called categorical information table (CI Table). CI table stores the co-occurrence statistics for categorical data. Similarity matrix construction using DISC has 4 steps. (1)Initially similarity matrix is defined using overlap measure, (2) formation of CI table, (3) Computation of similarity value (similarity between two attribute is measured using the cosine similarity measure), (4) Updating similarity matrix. DISC satisfies non negative, symmetric, commutative and triangle inequality property.

DILCA - DIstance Learning in Categorical Attribute is the measure used by the author [12,13]. Co-occurrence table is formed for all the features using symmetric uncertainty, a matrix is generated and conditional probability is applied, the results are given to the Euclidean measure to find the similarity between the attributes.

The author presents an improved squeezer algorithm called NabSqueezer [14]. It attempt to improve the clustering accuracies of existing categorical data clustering algorithms by giving greater weight to uncommon attribute value matches in similarity computations. Two stages (scan) of the algorithm are: (1) Frequency of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

occurrence is calculated (MSFVS) (2) similarity computation and cluster formation. Advantage: (1) Outlier can be handled; (2) Number of cluster is not needed. The results of the system depends on tuning parameter's'.

The author develops a TOP DATA algorithm [15], exploit a diagonal traversal strategy and TOP-MATA algorithm, a max-first traversal strategy for the search of top-K pairs. TOPMATA has the advantages of saving the computations for false-positive item pairs and can significantly reduce I/O costs. Limitation of the system is diagonal traversal strategy usually leads to more I/O costs.

Cosine similarity is a popular method for text mining. It is used for comparing the document (word frequency) and finds the closeness among the data points in clustering. Its range lies between 0 and 1. One desirable property of Cosine similarity [16] is independent of document length. Limitation is the terms are assumed to be orthogonal in space. If the value is zero no similarity exist between the data element and if the vale is 1 similarity exist between two elements.

3. PROPOSED WORK

3.1 Proposed System Architecture

The system architecture for OBCDC (Occurrence Based Categorical Data Clustering) is represented in Figure1. The OBCDC system consists of four modules, such as data preprocessing, similarity matrix generation, cluster formation and validation.

Preprocessing: Data from different data source (UCI repository) are collected and preprocessed. Data in real world are incomplete, inconsistent and noisy. Quality of resultant cluster depends on the quality of input data given to the system. Preprocessing deals with ensuring quality input data. Data cleaning deals with filling missing values, handling noisy data and resolving inconsistencies in data. (Missing value are ignored, noise data are removed).

Similarity matrix generation: It uses three functions for similarity computation. They are, (1) FrequencyComputation : Frequency computation deals with calculating the frequency of occurrence for each attributes available in the dataset. (2) OccurranceBasedCosine: Occurrence information are generated and stored in a multi dimensional array. Occurrence Based Cosine function deals with computing the similarity matrix using cosine similarity defined in equation 1. (3) OccurranceBasedSBMS: Occurrence Based SBMS deals with computing the similarity matrix using simple binary matching similarity defined in equation 2.

Cluster formation: Similarity matrix is constructed and maximum similarity for each attribute is found. ClusterFormation deals with grouping data with maximum similarity (greater the similarity closer the data points) together into a cluster.

Cluster validation: It is the process of evaluating the cluster results in a quantitative and objective manner. Generated clustered are validated using accuracy and error rate.

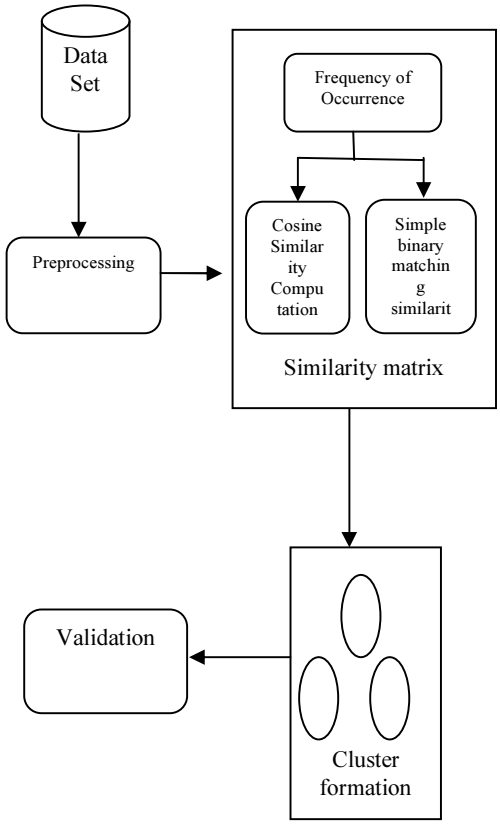

Figure1. System Architecture

3.2 Definitions

Definition 1: Occurrence information: An event that happens in connection with another

Definition2: Let D be data set with n tuple. Each tuple consist of A1, A2,..Am categorical attribute with domain D_1,D_2...D_m. Occurrence of each attribute is O_w (Ai). Occurrence based cosine

Journal of Theoretical and Applied Information Technology
10th October 2014. Vol. 68 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

similarity (OBCS) between X and Y is measured as:

$$OBCS(X,Y) = \frac{\sum_{i=1}^{n} O_w(X_i) * O_w(Y_i)}{\sqrt{\sum_{i=1}^{n}(O_w(X_i))^2} * \sqrt{\sum_{i=1}^{n}(O_w(Y_i))^2}} \quad (1)$$

Definition 3: Let D be data set with n tuple. Each tuple consist of A1, A2,..Am categorical attribute with domain D1,D2...Dm. Occurrence of each attribute is O_w(Ai). Occurrence based simple binary matching similarity (OSBMS) between X and Y is measured as:

$$OSBMS(X,Y) = \sum_{i=1}^{m} f(x,y) \quad (2)$$

$$f(x,y) = \begin{cases} O_w(\text{Ai}) & if\ X_k = Y_k \\ 0 & otherwise \end{cases} \quad (3)$$

Definition 4: Non Negative: The range of occurrence based cosine similarity always lies between 0 and 1. The range of occurrence based simple binary matching similarity always lies between 0 and α

Definition 5: Symmetric: OBCS(X,Y) = OBCS(Y,X) and OSBMS(X,Y) = OSBMS(X,Y). Due to the diagonal symmetric nature of the matrix (axy = ayx) similarity between items X and Y is same as the similarity between items Y and X. Considered the example X = { yellow, small, stretch } and Y = { yellow, small, dip}. OBCS (X,Y) = OBCS(Y,X) = 0 .8326 and OSBMS(X,Y) = OSBMS(X,Y) = 6

Definition 6: Triangular Inequality: The triangle inequality for cosine is same as for occurrence based cosine. To rotate from x to y is to rotate to z and hence to y. The sum of those two rotations cannot be less than the rotation directly from x to y.

3.3 Sample Walkthrough

Considered the balloon dataset shown in table 1,

Table 1: Balloon data set

Color	Size	Act
Yellow	Small	Stretch
Yellow	Small	Dip
Yellow	Large	Stretch
Yellow	Large	Dip

The frequencies of occurrence array O are:
O [Yellow] = 4, O [Small] = 2, O [Large] = 2, O [Stretch] = 2, O [Dip] = 2. Assume X = { yellow, small, stretch } , Y = {yellow, small, dip} and Z = { yellow, small, stretch }. Similarity between X,Y and X,Z are calculated using equation 1 & 2 and results are shown in Table 2.

Table 2: SIMILARITY COMPARISON FOR BALLOON DATA SET

Similarity between data point	Cosine Similarity	Occurrence based Cosine Similarity	Simple Binary Matching	Occurrence based Simple Binary Matching
X,Y	1	0.836	2	6
X,Z	1	1	3	8

3.4 Procedure

Input: Given Dataset with 'N' categorical attribute
Output: 'k' clusters

Algorithm : OBCDC

```
begin
  Initialize n as total number of tuple
  read the dataset one by one
//Computer frequency
  call FrequencyComputation();
//Similarity Matrix formation
  for i=1 to n do
    for j=1 to n do
      Sim(X,Y) = OccurrenceBasedCosine(X,Y);
      Sim(X,Y)= OccurrenceBasedSMS(X,Y)/N;
    end for //j
  end for //i
//Cluster the data with max similarity
MaxSim = Select Sim(X,Y) with maximum value
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Check for Similarity value from Sim matrix > MaxSim
assign to the respective cluster Cluster[k++]
return final cluster formed
end

Function :OccurrenceBasedCosine(X,Y)
begin
Vector representation of string (X,Y)
multiplying each vector by its frequency of occurrence
for i=1 to vectorlength do
Numerator = Ow(X_i) * Ow (Y_i)
Xvector = Ow (X_i) * Ow (X_i)
Yvector = Ow (Y_i) * Ow (Y_i)
end for
compute result using eqution (1)
return result
end

Function : FrequencyComputation()
begin
Assign initial occurrence of an attribute (a_i) be zero
while not EOF do
for each attribute a_i on A
calculate the occurrence of each attribute as Ow (a_i)
end for
end while
return occurrence
end

Function : OccurrenceBasedSBMS()
begin
Conversion of string into vector
Multiplying each vector by its occurrence of (X,Y)
compute result using equation (2)
end

4. RESULT DISCUSSION

The cluster validation is the process of evaluating the cluster results in a quantitative and objective manner. Cluster evaluation is done either internal or external. The internal evaluation determines the quality of the cluster. The external evaluation determines the partitioning among the cluster. In this paper OBCS and OSBMS is validated using two such measure (1) Cluster Accuracy 'r' is defined a

$$r = \frac{\sum_{i=1}^{k} a_i}{n} \quad (4)$$

where
'n' refers number of instance in the dataset,
'a_i' refers to number of instance occurring in both cluster i and its corresponding class and
'k' refers to final number of cluster.
(2) Error rate 'E' is defined as

$$E = 1 - r, \quad (5)$$

where 'r' refers to the cluster accuracy.

Real world dataset: Two real life dataset, such as Mushroom and balloon are obtained from UCI machine learning repository [17]. Mushroom: Each tuple represent the physical characteristic of mushroom. Number of instance is 8124 and number of attribute is 22. It can be classified into edible (4028) and poisonous mushroom (3916). Accuracy and error rate for the Rock, K-modes, K-modes with simple matching (KmodesSM), Fuzzy K means and weighted fuzzy K-means (WeightedFM) algorithm are compared with the proposed algorithm (OBCS and OSBMS) using mushroom dataset.

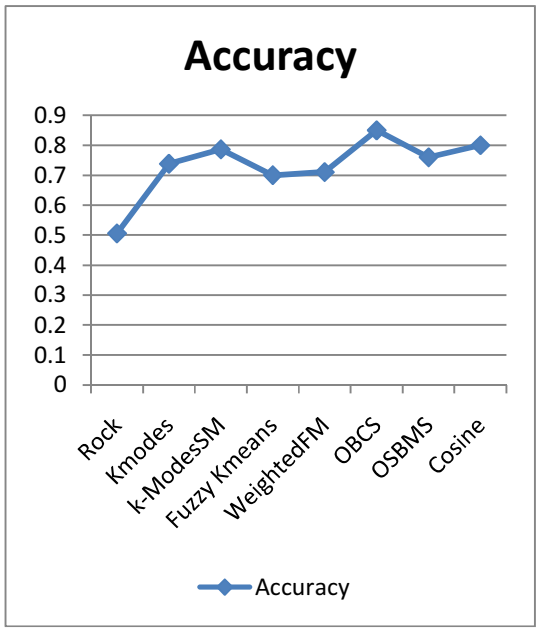

Figure 2. Comparison of Accuracy for different clustering Algorithms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

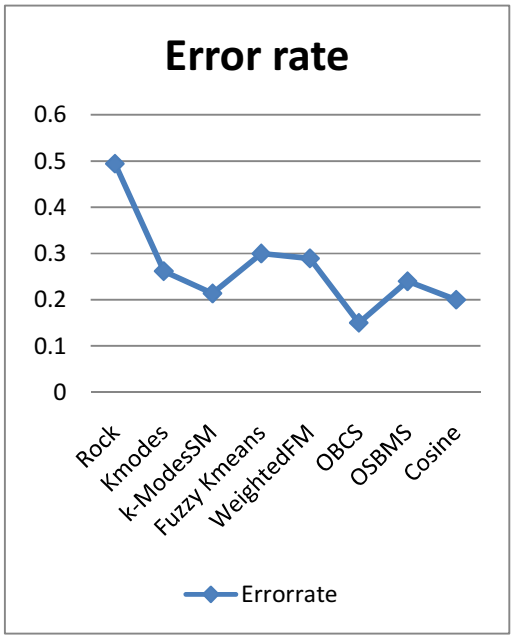

Figure 3. Comparison of error rate for different clustering Algorithms

Balloon dataset: Each tuple represent the physical characteristic of balloon. Number of instance is 20 and number of attribute is 4. This can be classified as Inflated is true (8) if the attribute age = adult and attribute act = stretch and Inflated is false (12) otherwise. Accuracy and error rate for balloon is shown in Table 3. It is clear that occurrence based cosine and occurrence based simple binary matching similarity outperform simple cosine similarity and simple binary matching similarity.

Table 3: Accuracy and Error rate for Balloon data set

Algorithm	Dataset	Accuracy	Error rate
Cosine Similarity	Balloon	0.804	0.196
OBCS		0.97	0.03
Simple binary matching similarity		0.8	0.2
OSBMS		0.9	0.1

5. CONCLUSION

This paper describes frequency based cosine similarity and simple binary matching similarity as an objective measure for categorical data clustering. The system architecture of OBCDC (Occurrence Based Categorical Data Clustering) algorithm is illustrated. The system uses four routine to compute similarity and cluster formation. Performance of OBCDC mainly depends on similarity value (MaxSim). Cluster validation is performed using accuracy and error rate for real world data set using different categorical clustering algorithm. Both for mushroom and balloon dataset occurrence based cosine similarity and simple binary matching similarity outperforms traditional cosine similarity and other clustering algorithm. The system can be further enhanced in future to achieve pure cluster formation and to reduce the execution time.

REFRENCES:

[1] Jiawei Han, Micheline Kamber.; *Data Mining Concepts and Techniques.* The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2000.

[2] Hana Rezankova,*Cluster Analysis and categorical data statistika,2009,* pp 216-232.

[3] Alan Agresti, *An Introduction to categorical data analysis, second edition*, A John Wiley & sons publication

[4] Z.Haung and Michael K.Ng.; A Fuzzy k-Modes Algorithm for Clustering Categorical Data,*IEEE Transaction On Fuzzy systems*, Vol. 7, No-4,1999.

[5] Zhexue Huang; Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery 2,1998,* pp283–304.

[6] HE Zengyou, XU Xiaofei.; SQUEEZER: An Efficient Algorithm for Clustering Categorical Data. Journal *on Computer Science & Technology,* Vol.17 No.5, 2002.

[7] S. Guha, R. Rastogi, and K. Shim.; ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, vol. 25, no. 5, 2000, pp 345-366.

[8] Dutta, M., A. Mahanta, and A. Pujari.; QROCK: A quick version of the ROCK algorithm for clustering of categorical data. *Pattern Recognition Letters*, vol. 26, 2005, pp2364-2373.

[9] V. Ganti, J. Gehrke, and R. Ramakrishnan.; CACTUS-clustering categorical data using summaries. *In KDD '99: Proceedings of the*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA, 1999 pp 73-83.

[10] Ng, M.K. and Jing, L.; A new fuzzy k-modes clustering algorithm for categorical data. *Int. J. Granular Computing, Rough Sets and Intelligent Systems,* Vol. 1, No. 1, 2009, pp105–119.

[11] Aditya Desai, Himanshu Singh, Vikram Pudi, DISC: Data Intensive Similarity for Categorical Attributes. *The 15thPacific-Asia Conference on Knowledge Discovery and Data Mining,Shenzen,* China, 2011.

[12] Dino Ienco, Ruggero G. Pensa And Rosa Meo.; From Context to Distance: Learning Dissimilarity for Categorical Data Clustering , *ACM Transactions on Knowledge Discovery from Data ,* 2011.

[13] D Ienco, R Pensa, R Meo.; Context-based distance learning for categorical data clustering. *Advances in Intelligent Data Analysis VIII* , 2009, pp. 83-94.

[14] Zengyou He, Xiaofei Xu, Shenchun Deng.; Improving Categorical DataClusterinq Algorithm by Weighting Uncommon Attribute Value Matches. *ComSIS* Vol. 3, No.1, 2006.

[15] Shiwei Zhu, Junjie Wu, Hui Xiong , Guoping Xia.; Scaling up top-K cosine similarity search. Data & Knowledge Engineering 70, 2011 ,pp 60–83.

[16] Anand Rajaraman, Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2011.

[17] UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/ MLRepository.html.

[18] Fuyuan Cao, Jiye Lianga, Deyu Li, Liang Bai,Chuangyin Dang," A dissimilarity measure for the k-Modes clustering algorithm"Knowledge-Based Systems 26, 2012, pp 120–127.