ISSN 2321-6905

International Journal of Science Engineering and Advance Technology

A Case Study of Clustering Algorithms for Categorical Data sets

Ms.V.Sujitha
Anurag Group of Institutions
Hyderabad, India

Mr.B.Venkateshwar Reddy
Anurag Group of Institutions
Hyderabad, India

Mr.G.Vishnu Murthy
Anurag Group of Institutions
Hyderabad, India

***Abstract*- The data clustering, an unsupervised pattern recognition process is the task of assigning a set of objects into groups called clusters so that the objects in the same cluster are more similar to each other than to those in other clusters. Most traditional clustering algorithms are limited to handling numerical data. However, these cannot be directly applied for clustering of nominal data, where domain values are discrete and have no ordering. In this paper various categorical data clustering algorithms are going to be addressed in detail. A detailed survey on existing algorithms will be made and the scalability of some of the existing algorithms will be examined**.

Keywords—data clustering, categorical data, data mining, scalability

I. INTRODUCTION

Data clustering or unsupervised learning is an important but an extremely difficult problem. The Objective of clustering is to partition a set of unlabelled objects into homogeneous groups or clusters. A number of application areas use clustering techniques for organizing or discovering structure in data, such as data mining, information retrieval, image segmentation and machine learning. In real world problems, clusters can appear with different shapes, sizes, data sparseness, and degree of separation. Further, noise in the data can mask the true underlying structure present in the data. Clustering techniques require the definition of a similarity measure between patterns, which is not easy to specify in the absence of any prior knowledge about cluster shapes. Additionally, quantitative evaluation of the quality of clustering results is difficult due to the subjective notion of clustering. A large Number of clustering algorithms exists, yet no single algorithm is able to identify all sorts of cluster shapes and structures that are encountered in practice. Each algorithm has its own approach for estimating the number of clusters, imposing a structure on the data and validating the resulting clusters. Model-based techniques assume particular cluster shapes that can be given a simple and compact description. Examples of model-based techniques include: parametric density approaches, such as mixture decomposition techniques; prototype-based methods, such as central clustering, square-error clustering, K-means or K-medoids clustering and shape fitting approaches. Model order selection is sometimes left as a design parameter to be specified by the user, or it is incorporated in the clustering procedure. Most of the above

Techniques utilize an optimization procedure tuned to a particular cluster shape, or emphasize cluster compactness. Fisher et al. proposed an optimization-based clustering algorithm, based on a pair wise clustering cost function, emphasizing cluster connectedness. Non-parametric density based clustering methods attempt to identify high density clusters separated by low density regions, Graph-theoretical approaches have mostly been explored in hierarchical methods, that can be represented graphically as a tree or dendrogram. Both agglomerative and divisive approaches (such as those based on the minimum spanning tree - MST) have been proposed; different algorithms are obtained depending on the definition of similarity measures between patterns and between clusters. The single-link (SL) and the complete-link (CL) hierarchical methods are the best known techniques in this class, emphasizing, respectively, connectedness and compactness of patterns in a cluster. Prototype-based hierarchical methods, which define similarity between clusters based on cluster representatives, such as the centroid, emphasize compactness. Variations of the prototype-based hierarchical clustering include the use of multiple prototypes per cluster, as in the CURE algorithm. Other hierarchical agglomerative clustering algorithms follow a split and merge technique, the data being initially split into a large number of small clusters, merging being based on inter-cluster similarity; a final partition is selected among the clustering hierarchy by thresholding techniques or based on measures of cluster validity. Treating the clustering problem as a graph partitioning problem, a recent approach, known as spectral clustering, applies spectral graph theory for clustering. Among the various clustering methods, the K-means algorithm, which minimizes the squared-error criteria, is one of the simplest clustering algorithms. It is computationally efficient and does not require the user to specify many parameters. Its major limitation,

however, is the inability to identify clusters with arbitrary shapes, ultimately imposing hyper-spherical shaped clusters on the data. Extensions of the basic K-means algorithm include: use of Mahalanobis distance to identify hyper-ellipsoidal clusters, introducing fuzzy set theory to obtain non-exclusive partitions; and adaptations to straight line fitting .While hundreds of clustering algorithms exist, it is difficult to find a single clustering algorithm that can handle all types of cluster shapes and sizes, or even decide which algorithm would be the best one for a particular data set.

II. CLUSTERING CATEGORICAL DATA

These cannot be directly applied for clustering of categorical data, where domain values are discrete and have no ordering defined.

An example of categorical attribute is

Sex = {male, female} or shape= {circle, rectangle,. . .}

Many categorical data clustering algorithms have been introduced in recent years, with applications to interesting domains such as protein interaction data.

A. Categorical data clustering algorithms

1) K-modes

Objective:

- Extending K-means to categorical domains
- Using a simple matching dissimilarity measure for categorical objects
- Replacing means of clusters by modes
- Using a frequency-based method to find the modes

Algorithm:

1. Select K initial modes, one for each cluster
2. Allocate an object to the cluster whose mode is the nearest to it. Update the mode of the cluster
3. After all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes if an object is found its nearest mode belongs to another cluster, reallocate the object to that cluster and update the modes of both clusters
4. Repeat 3 until no objects has changed clusters

Advantage:

The k-modes algorithm is faster than the k-means because it needs less iteration to converge

2) Squeezer

- Squeezer, a one-pass algorithm is proposed.
- Squeezer repeatedly read tuples from dataset one by one.
- When the first tuple arrives, it forms a cluster alone. The consequent tuples are either put into an existing cluster or rejected by all existing clusters to form a new cluster according to the given similarity function.

3) LIMBO

- LIMBO, a scalable hierarchical categorical clustering algorithm built on the Information Bottleneck (IB) framework.
- As a hierarchical algorithm, LIMBO has the advantage that it produces clustering's of different size in a single execution.
- LIMBO can also control the size of the model it builds to summarize the data. We use LIMBO to cluster both tuples (in relational and market-basket data sets) and attribute values.
- We define a novel distance between attribute values that allows us to quantify the degree of interchangeability of attribute values within a single attribute.

Algorithm:

- The LIMBO algorithm proceeds in three phases.
- In the first phase, the DCF tree is constructed to summarize the data.
- In the second phase, the DCFs of the tree leaves are merged to produce a chosen number of clusters.
- In the third phase, we associate each tuple with the DCF to which the tuple is closest.

4) STIRR

- STIRR, an iterative algorithm based on non-linear dynamical systems
- The approach used can be mapped to a certain type of non-linear systems. If the dynamical system converges, the categorical databases can be clustered.
- Another recent research shows that the known dynamical systems cannot guarantee

convergence, and proposes a revised dynamical system in which convergence can be guaranteed.

- STIRR also provides a natural framework for effectively visualizing the underlying relational data
- The STIRR algorithm not only takes into consideration items that appear together in a tuple, but also identifies relationships amongst items occurring in different tuples.

Overview of the algorithm

- Iterative method – The STIRR algorithm is an iterative method, the number of iterations depending upon the dataset in consideration. The algorithm keeps on performing the same steps a number of times until a result is obtained which does not change on further iterations.
- Assigning and propagating weights on categorical values – A relational table is taken as input to the algorithm. This relational table has fields (attributes) that can take values in a particular domain. The STIRR algorithm takes each distinct value in the table and performs a series of steps to assign numerical values (weights) to it.
- Similarity measure obtained from co-occurrence of values in dataset – Each distinct value in the database is assigned a weight. In the first iteration of the STIRR algorithm, the weight of each distinct value is calculated depending on with what values this distinct value appears in the database.

5) ROCK

Objective:

- ROCK: Robust Clustering using links,
- Use links to measure similarity/proximity
- Not distance based

Algorithm:

- Draw random sample
- Cluster with links
- Label data in disk

procedure cluster(S, k)
begin
1. $link := \text{compute\_links}(S)$
2. **for each** $s \in S$ **do**
3. $\quad q[s] := \text{build\_local\_heap}(link, s)$
4. $Q := \text{build\_global\_heap}(S, q)$
5. **while** $\text{size}(Q) > k$ **do** {
6. $\quad u := \text{extract\_max}(Q)$
7. $\quad v := \max(q[u])$
8. $\quad \text{delete}(Q, v)$
9. $\quad w := \text{merge}(u, v)$
10. $\quad$ **for each** $x \in q[u] \cup q[v]$ **do** {
11. $\quad\quad link[x, w] := link[x, u] + link[x, v]$
12. $\quad\quad \text{delete}(q[x], u); \text{delete}(q[x], v)$
13. $\quad\quad \text{insert}(q[x], w, g(x, w)); \text{insert}(q[w], x, g(x, w))$
14. $\quad\quad \text{update}(Q, x, q[x])$
15. $\quad$ }
16. $\quad \text{insert}(Q, w, q[w])$
17. $\quad \text{deallocate}(q[u]); \text{deallocate}(q[v])$
18. }
end

6) CLICK

CLICK a novel algorithm for mining categorical (subspace) clusters.

Objective:

i) A novel formalization of categorical datasets as k-partite graphs, where clusters correspond to k-partite cliques after post-processing.

ii) A selective vertical expansion approach to guarantee a complete search; overlapping cliques are merged to report more meaningful clusters.

iii) CLICKS outperform existing approaches by over an order of magnitude. It can mine subspace clusters and scales extremely well for high dimensions.

7) CACTUS

CACTUS is a fast summarization-based algorithm for clustering categorical data. CACTUS exploits the small domain sizes of categorical attributes.

Objective: The central idea in CACTUS is that summary information constructed from the dataset is sufficient for discovering well-defined clusters.

The properties that the summary information typically fits into main memory and that it can be constructed efficiently

Algorithm:

- CACTUS consists of three phases: summarization, clustering, and validation.
- In the summarization phase, we compute the summary information from the dataset.
- In the clustering phase, we use the summary information to discover a set of candidate clusters.
- In the validation phase, we determine the actual set of clusters from the set of candidate clusters.

8) COOLCAT

Algorithm:

1. Initialization:

The initialization step "bootstraps" the algorithm, finding a suitable set of clusters out of a sample S, taken from the data set ($|S| << N$), where N is the size of the entire data set. We first find the k most "dissimilar" records from the sample set by maximizing the minimum pair wise entropy of the chosen points.

2. Incremental Step:

After the initialization, we process the remaining records of the data set (the rest of the sample and points outside the sample) incrementally, finding a suitable cluster for each record. This is done by computing the expected entropy that results of placing the point in each of the clusters and selecting the cluster for which that expected entropy is the minimum. We proceed in the incremental step by bringing a buffer of points to main memory and clustering them one by one.

Advantage:

- Given a set of clusters, COOLCAT will place the next point in the cluster where it minimizes the overall expected entropy.
- COOLCAT acts incrementally, and it is capable to cluster every new point without having to re-process the entire set.
- Therefore, COOLCAT is suited to cluster data streams.
- This makes COOLCAT applicable in a large variety of emerging applications such as intrusion detection, and e-commerce data.

9) CLOPE

CLOPE algorithm introduce a distance measure between partitions based on the notion of generalized conditional entropy

Objective:

- A genetic algorithm approach is utilized for discovering the median partition

Problem definition Given D and r, find a clustering C that maximize $Profit_r(C)$.

```
/* Phrase 1 - Initialization */
1: while not end of the database file
2:    read the next transaction ⟨t, unknown⟩;
3:    put t in an existing cluster or a new cluster C_i
        that maximize profit;
4:    write ⟨t, i⟩ back to database;

/* Phrase 2 - Iteration */
5: repeat
6:    rewind the database file;
7:    moved = false;
8:    while not end of the database file
9:       read ⟨t, i⟩;
10:      move t to an existing cluster or new cluster C_j
           that maximize profit;
11:      if C_i ≠ C_j then
12:         write ⟨t, j⟩;
13:         moved = true;
14: until not moved;
```

III. CONCLUSION

In this paper, a detailed analysis of various categorical data clustering techniques and its advantages and disadvantages were conducted wherein the drawbacks of each technique was considered. Although, a large number of algorithms have been introduced for clustering categorical data, there is no single clustering algorithm that performs best for all data sets and can discover all types of cluster shapes and structures presented in data. Each algorithm has its own strengths and weaknesses. For a particular data set, different algorithms, or even the same algorithm with different parameters, usually provide distinct solutions. Therefore, it is difficult for users to decide which algorithm would be the proper alternative for a given set of data.

IV. REFERENCES

[1] L. Kaufman and P.J. Rousseeuw, Finding Groups Data: An Introduction to Cluster Analysis. Wiley Publishers, 1990.

[2] A.K. Jain and R.C. Dubes, Algorithms for Clustering. Prentice-Hall, 1998.

[3] P. Zhang, X. Wang, and P.X. Song, "Clustering Categorical Data Based on Distance Vectors," The J. Am. Statistical Assoc., vol. 101, no. 473, pp. 355-367, 2006.

[4] J. Grambeier and A. Rudolph, "Techniques of Cluster Algorithms in Data Mining," Data Mining and Knowledge Discovery, vol. 6, pp. 303-360, 2002.

[5] K.C. Gowda and E. Diday, "Symbolic Clustering Using a New Dissimilarity Measure," Pattern Recognition, vol. 24, no. 6, pp. 567- 578, 1991.

[6] Z. Huang, "Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values," Data Mining and Knowledge Discovery, vol. 2, pp. 283-304, 1998.

[7] Z. He, X. Xu, and S. Deng, "Squeezer: An Efficient Algorithm for Clustering Categorical Data," J. Computer Science and Technology, vol. 17, no. 5, pp. 611-624, 2002.

[8] D. Gibson, J. Kleinberg, and P. Raghavan, "Clustering Categorical Data: An Approach Based on Dynamical Systems," VLDB J., vol. 8, nos. 3-4, pp. 222-236, 2000.

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

[10] M.J. Zaki and M. Peters, "Clicks: Mining Subspace Clusters in Categorical Data via Kpartite Maximal Cliques," Proc. Int'l Conf.Data Eng. (ICDE), pp. 355-356, 2005.

[11] V. Ganti, J. Gehrke, and R. Ramakrishnan, "CACTUS: Clustering Categorical Data Using Summaries," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 73-83.

[12] Barbara, Y. Li, and J. Couto, "COOLCAT: An Entropy-Based Algorithm for Categorical Clustering," Proc. Int'l Conf. Information and Knowledge Management (CIKM), pp. 582-589, 2002.

[13] Y. Yang, S. Guan, and J. You, "CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp

V. AUTHORS BIOGRAPHY

Ms.V.Sujitha received B.Tech Computer Science and Engineering from JNTUH. Pursuing M.Tech Computer Science and Engineering from Anurag group of Institutions, Hyderabad, India. Her area of interest includes Data Mining, Machine Learning and Pattern Recognition.

Mr.B.Venkateshwar Reddy Received M.Sc Mathematics from Osmania University and M.E Computer Science And Engineering from Sathyabama University, Chennai. Presently working as a Assistant Professor in school of Engineering, Anurag Group of Institutions, Hyderabad, India. Published ten papers in various National and International Conferences, Journals.

Mr.G.Vishnu Murthy received his B.E and M.Tech degrees in Computer Science and Engineering. He has 15 years of teaching experience and pursuing Ph.D. from JNTU, Hyderabad.. He is the Life Member of ISTE, Member of IEEE computer society, Member of ACM, Member of CRSI and Member of CSI. He has guided more than 12 projects at P.G. Level and more than 80 projects at graduation level. He has 5 publications in international journals and presented 2papers.