American Journal of Applied Sciences 9 (4): 531-534, 2012
ISSN 1546-9239

An Efficient Way for Clustering Using Alternative Decision Tree

Gothai, E. and P. Balasubramanie
Department of Computer Science and Engineering,
Kongu Engineering College, Perundurai, Erode, Tamilnadu, India

Abstract: Problem statement: To Improve the quality of clustering; a Multi-Level Clustering (MLC) algorithm which produces a most accurate cluster with most closely related object using Alternative Decision Tree (ADT) technique is proposed. **Approach:** Our proposed method combines tree projection and condition for clustering formation and also is capable to produce a customizable cluster for varying kind of data along with varying number of cluster. **Results:** The experimental results shows that the proposed system has lower computational complexity, reduce time consumption; most optimize way for cluster formulation and clustering quality compared is compared effectively. **Conclusion:** The new method offers more accuracy of cluster data without manual intervention at the time of cluster formation. Compared to existing clustering algorithms either partition or hierarchical, our new method is more robust and easy to reach the solution of real world complex business problem.

Key words: Multi-level clustering, clustering quality, decision tree algorithm, most optimize, cluster data, clustering algorithm, cluster formation, either partition, without manual

INTRODUCTION

Clustering is an important method in data warehousing and data mining. It groups similar object together in a cluster (or clusters) and dissimilar object in other cluster (or clusters) or remove from the clustering process. That is, it is an unsupervised classification in data analysis that arises in many applications in different fields such as data mining, image processing, machine learning and bioinformatics. Since, it is an unsupervised learning method, it does not need train datasets and pre-defined taxonomies. But there are some special requirements for search results clustering algorithms, two of which most important is, clustering performance and meaningful cluster description. Lots of clustering method is available, among those hierarchical clustering and Partition Clustering is the widely used clustering methods. A Partition-clustering algorithm in their outputs produce one clustering set that consists of disjoint clusters, i.e., the data description is flat. In other words, partitioned clustering is nothing but pre-defined number of partition range. Where the total number of partition (k) range should be less than number of object (n) in the dataset. Partition clustering always should satisfy the condition $k < n$.

A Hierarchical clustering is a nested of partitions technique depend on the business requirements. It produces not just one clustering set in their outputs but a hierarchy of clusters. This method work for both kind of approach either bottom up and top down approach. In this method all record object arranged with in a big cluster, then big cluster are continuously divided into small clusters.

This study proposes a Multi-Level Clustering mechanism using alternative decision tree algorithm that combines the advantage of partition clustering (Elavarasi *et al*., 2011), hierarchical clustering (Lancichinetti and Fortunato, 2009; Mirzaei and Rahmati, 2010) and incremental clustering technique for rearranging the most closely related object. The clustering initiation should happen based on the short name value, each short name pointing to the appropriate whole record object. At each step during the clustering (Jiang *et al*., 2009), we assure the quality of clustering (Kumaran and Rangarajan, 2011), should be more accurate as well as the content of each clustering should be closely related object. If the quality of cluster still needs to improve (which mean again want to split a data) at any step then again we need to perform clustering technique depend on the business requirement otherwise, the clustering will terminated and current clustering is a result. The proposed MLC

Corresponding Author: Gothai, E., Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamilnadu, India

algorithm has been experimentally tested on a set of data to find a cluster with closely related object. This method is used to overcome the existing system problem, such as manual intervention, misclassification and difficulties of finding a partition range and so on.

Background of alternative decision tree: The wide use of the search engine and the problem remained in it have motivated research in search results clustering. Currently various industrial systems implement search results clustering in their metasearch engines or desktop search software. Examples of these systems include Vivisimo, Mooter, Copernic, iBoogie, Kartoo, Groxis, Dogpile, Carrot. The most commonly used clustering algorithms can be typically classified into four types: Hierarchical Method (BIRCH, CURE), Partitional Clustering (K-means, K-prototypes, K-mode), Density-based Clustering and Grid-based Clustering.

Some researchers have focused on clustering information between words to improve quality of the clustering. Our idea in this study is different we utilize tree projection and condition for clustering formation and also the proposed algorithm is capable to produce a customizable cluster for varying kind of data along with varying number of cluster.

An Alternative Decision Tree (ADTree) is a learning (Mahmoodian *et al*., 2010) method for record classification. It generalizes decision tree for performing the supervised learning. The goal is to create a model that predicts the value of a target variable based on several input variables. In our scenario prediction should be short name (Xi) with respect to the full name (Yi). It consists of two nodes decision nodes and prediction node. Decision nodes specify the prediction condition, if condition is satisfied return true else false.

Consider the Training dataset {(X1, Y1) ... (Xi, Yi) ..., (Xn, Yn)}, each X is an example with a label Y. A set of weights Wi corresponding to each instance node:

```
if (is_valid[short_name])
then
    if (contains_cluster[short_name])
    then
        return
    existing_cluster[short_name];
    else
        return new_cluster[short_name];
    endif
else
    return 0;
endif
```

ADTree algorithm consists of precondition, condition and two score values. A pre-condition is simply a logical condition, here we are checking whether short_name is valid or not, if it is an valid then proceed with appropriate record or return a FALSE(0) value and exit from the process. A condition predicates the attribute (Chen *et al*., 2009) comparison value, here we are checking whether the clustering index value contains the short_name value or not. If already contains then return existing clustering index else create a new cluster index with value of short_name.

This algorithm looks to be similar to others in a way like it adds a decision rule at a time by finding the best split to expand the current tree. The basic difference between this and the rest are:

- Decision nodes are added at any location in the tree and not just at the leaves
- The slitting criteria are different

From our description of the alternating decision trees it is clear that they are generalization of decision trees.

MATERIALS AND METHODS

MLC forms a tree for the clustering process (Fig 1). In the tree structure, the height of each level of nodes represents the similar degree between clusters. MLC incorporate the futures of ADTree features and overcome the existing hierarchical clustering problem.

Here we did not use any split algorithm for splitting data into a cluster; alternatively we are using ADTree technique for splitting a whole data into cluster.

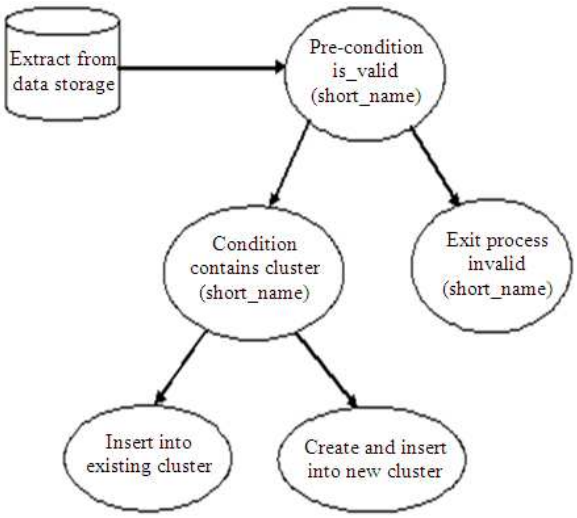

Fig. 1: Multi-level clustering formation

ADTree divide the data based on short name, If cluster is already available with the short name then insert a record into the same cluster else create a new cluster with the new name of short name then insert into a new cluster. In each cluster, sub-set short name points to the whole record.

The cluster formation method mainly focus on form a similarity (Arasu *et al*., 2006) value in single group, for this purpose we are using different method and result of each method is different cluster based on data and spread condition. ADTree can be defined as a simple basic rules, it is nothing but pre-condition. If pre-condition return a TRUE value then starts the process else exit from the process. In condition we are making a decision based on condition value, here we are checking already clustering is available or not.

Proposed algorithm implementation:

Initialize: Parent_List[n] ← 0,
Child_List[n] ← 0,
Grant_child_list[n] ←0;
Loop L1: While !endOfRecord[Record]
C1 ← Level_1_cluster_attr_value;
Position ← Size[Parent_List]+1;
If(is_valid[C1])
then
C2 ← Level_2_cluster_attr_value;
C3 ← Level_3_cluster_attr_value;
Child_Position ← Size[Child_List]+1;

Loop L2: while !endOfRecord[Record]
If(!contains[parent_list, C2])
Then

Parent_List[Position] ← Create new Cluster C1, Parent_Position;
Child_List[n] ← Create new Cluster C2, Child_Position;
Grant_child_list[n] ← Create new Cluster C3, Child_Position;
Parent_List[Position]←Insert into new Cluster {Parent_Position, (C1,vector[Record_Information])};
Child_List[n] ← Insert into new Cluster {Child_Position, (C2,vector[Record_Information])};
Grant_child_list[n] ← Insert into new Cluster {Child_Position, (C2,vector[Record_Information])};
return new_cluster;
else
Parent_List[Position] ← Insert into existing Cluster {Parent_Position, (C1,vector[Record_Information])};
Child_List[n] ← Insert into existing Cluster {Child_Position, (C2,vector[Record_Information])};
Grant_child_list[n] ← Insert into existing Cluster {Child_Position, (C2,vector[Record_Information])};
return existing_Cluster;
endif

Goto L2
else
return 0;
endif
Goto L1

RESULTS

The experimental results shows the proposed system has lower computational complexity, reduce time consumption, most optimize way for cluster formulation and better clustering quality compared with the existing hierarchical clustering algorithm. We ran extensive experiments with them to find time consumption and compared them with various versions of existing algorithms in order to show this new system reduces the time consumption.

DISCUSSION

We simulate and study the constrained MLC algorithm without alignment and without manual intervention using the ADTree model as that has been discussed.

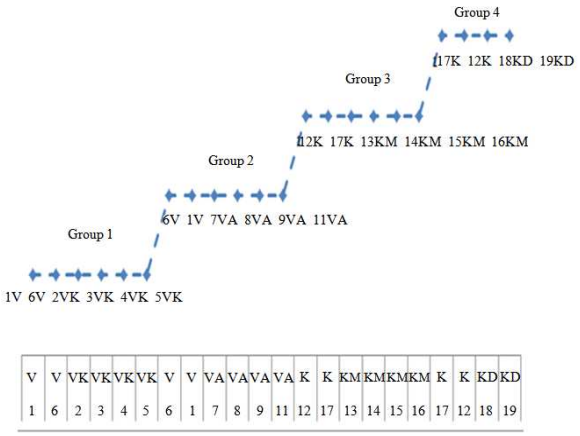

Fig. 2: Simulation results of group formation

Fig 1 shows the simulation of the MLC construction and Fig. 2 shows the simulation result of the algorithm and performance effectiveness. As can be seen, the MLC algorithm has proven in terms of performance compared with the existing clustering system.

CONCLUSION

In this study, a new efficient method is introduced for clustering formation using ADTree algorithm. The new method offers more accuracy of cluster data without manual intervention at the time of cluster formation. Compare to existing clustering algorithm either partition or hierarchical, our new method is more robust and easy to reach the solution of real world complex business problem.

REFERENCES

Arasu, A., S. Babu and J. Widom, 2006. The CQL continuous query language: semantic foundations and query execution. Int. J. Large Data Bases. 15: 121-142. DOI: 10.1007/s00778-004-0147-z

Chen, W., W. Fan and S. Ma, 2009. Incorporating cardinality constraints and synonym rules into conditional functional dependencies. Inform. Process. Lett., 109: 783-789. DOI: 10.1016/j.ipl.2009.03.021

Elavarasi, S.A., J. Akilandeswari and B. Sathiyabhama, 2011. A survey on partition clustering algorithms. Int. J. Enterprise Comput. Bus. Syst., 1: 1-14.

Jiang, P., C. Zhang, G. Guo, Z. Niu and D. Gao, 2009. A K-means approach based on concept hierarchical tree for search results clustering. Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, Aug. 14-16, IEEE Xplore Press, Tianjin, pp: 380-386. DOI: 10.1109/FSKD.2009.658

Kumaran, M.S. and R. Rangarajan, 2011. Ordering Points to Identify the Clustering Structure (OPTICS) with ant colony optimization for wireless sensor networks. Eur. J. Sci. Res., 59: 571-582.

Lancichinetti, A. and S. Fortunato, 2009. Community detection algorithms: A comparative analysis. Cornel University Library.

Mahmoodian, N., R. Abdullah and M.A.A. Murad, 2010. Classifying maintenance request in bug tracking system. Int. J. Comput. Sci. Inform. Secu., 8: 32-38.

Mirzaei, A. and M. Rahmati, 2010. A novel hierarchical-clustering-combination scheme based on fuzzy-similarity relations. IEEE Trans. Fuzzy Syst., 18: 27-39. DOI: 10.1109/TFUZZ.2009.2034531