EXTENDED ONLINE DIVISIVE AGGLOMERATIVE CLUSTERING

Ibrahim Musa Ishag Musa, Dong Gyu Lee, Keun Ho Ryu

Database/Bioinformatics Laboratory, Chungbuk National University, Korea
{Ibrahim, dglee, khryu}@dblab.chungbuk.ac.kr

ABSTRACT: Clustering data streams has an importance over many applications like sensor networks. Existing hierarchical methods follow a semi fuzzy clustering that yields duplicate clusters. In order to solve the problems, we propose an extended online divisive agglomerative clustering on data streams. It builds a tree-like top-down hierarchy of clusters that evolves with data streams using geometric time frame for snapshots. It is an enhancement of the Online Divisive Agglomerative Clustering (ODAC) with a pruning strategy to avoid duplicate clusters. Our main features are providing update time and memory space which is independent of the number of examples on data streams. It can be utilized for clustering sensor data and network monitoring as well as web click streams.

KEYWORDS: Data Mining, Data Stream, Time series, Hierarchical Clustering, Sensor Network

1. INTRODUCTION

In current emerging network applications, *data stream* has became of importance, as data flows in and out at high speeds from multiple sources (e.g., sensor networks, web-click streams, and etc) that cause storage problems. The process of extracting useful information from such data becomes a challenge and has been studied recently (Domingos, et al 2000), (Babu, et al 2002), Focusing on hierarchical clustering on data streams, this paper addresses the duplicate clusters and incorrect splitting problems found in ODAC. We use a geometric time frame to capture streams over time and apply EODAC thus allows for viewing clusters over time. This paper is organized as follows. Section 2 addresses the related works and in Section 3, we give more details about our enhancement clearly in the splitting criteria, and Section 4 concludes this work.

2. RELATED WORK

Clustering streaming data has been studied widely because of its importance in decision support systems. The most of studies was focused on sample clustering like STREAM (Han, J., et al), and Cluster(Han, J., et al), rather than variable clustering which has been clearly addressed in(Pereira, R et al, 2008).The main difference between sample and variable clustering is that in sample clustering we cluster samples whereas in variable clustering we group the variables (attributes) into similar groups. Furthermore, hierarchical clustering has three advantages over other clustering methods even in stationary data. Those advantages are firstly it does not require involvement of users in specifying the number of clusters, as it happens in other clustering methodologies for example in partitioning clustering. Secondly, most of hierarchical methods do not require whole data to be available at once as BIRCH(Zhang, et al, 1996). It has a linear time complexity with respect to the size of input (Mohamed, A. N., et al, 2008).

3. PROPOSED CLUSTERING METHOD

We propose Extended Online Divisive Agglomerative Clustering (EODAC). The main goal of EODAC is to build a hierarchical tree-based clustering structure of variables in which leaves are the current clusters. The union of leaves is the complete set of variables and the intersection of leaves is the empty set.

3.1 Similarity Measure

The similarity measure used in this paper is Pearson's Correlation Co-efficient on data streams as in(Pereira, R et al, 2008), because we consider the correlation based on continuous variables.

$$corr(a,b) = \frac{P-\frac{AB}{n}}{\sqrt{A_2-\frac{A_2^2}{n}}\sqrt{B_2-\frac{B_2^2}{n}}} \qquad (1)$$

$$where\ A = \sum a_i,\ B = \sum b_{i,}$$
$$A_2 = \sum a_{i,}^2\ B_2 = \sum b_{i,}^2\ P = \sum a_i b_{\cdot i}$$

Since this value happened to be of a high range, Rooted Normalized one minus correlation as a normalized version is used as a distance measure which is given by the formula shown bellow. This version gives values in the range [0-1]. The cluster diameter is the highest dissimilarity between two data streams belonging to the same cluster or the variable variance in the case of single variable clusters.

$$rnomc(a,b) = \sqrt{\frac{1-corr(a,b)}{2}} \qquad (2)$$

3.2 Growing The Hierarchy

In this paper, the tree structure is thought of as having internal nodes that are different from leaf nodes in the sense that leaf nodes constitutes the real clusters. So we always find the diameter of each node which is a pair of variables whose dissimilarity is the largest with respect to a statistical hoeffding bound (Han, J., et al), given by the formula illustrated bellow.

$$\epsilon = \sqrt{\frac{R^2 \ln\left({}^{1}/_{\delta}\right)}{2n}} \tag{3}$$

$$\textit{where } R \textit{ is variable range,}$$
$$n \textit{ is number of examples}$$

Although the data stream is not a real random variable, we assume it as a random variable just to avoid bias and take decisions based on statistical measurements. As each leaf is fed with a different number of example n , each leaf C_k has its own $\in_k$ .Let d(a,b) be the dissimilarity measure used to choose the pair of data stream representing the diameter, and $D_k = \{(x_i, y_i) | x_i, y_i \in C_k, i < j\}$ be the set of pairs of variables included in a specific leaf C_k . After seeing n samples at the leaf, let $(x_1 x_{2)} \in \{(x_1 x_2) \in D_k | d(x_1, x_2) \geq d(x_i, x_j), \forall (x_i, x_j) \in D_k\}$ be the pair of variables with maximum dissimilarity within the cluster C_k, and $(y_1 y_2)$ be the pair of variables with the second maximum dissimilarity within the same cluster thus $d_1 = d(x_1, x_2), d_2 = d(y_1, y_2)$ let $\Delta d = d_1 - d_2$ be another random variable applying hoeffding bound to this variable, if $\Delta d > \in_k$, we can confidently insist that with probability $1 - \delta$, the difference between $d_1$and $d_2$is greater than zero and select (x_1, x_2) as the pair of variables representing the diameter of the cluster as shown in the following equation.

$$d_1 - d_2 > \in_k \underset{hence}{\Longrightarrow} diameter\ (C_k) = d_1 \tag{4}$$

This rule only triggers when the leaf is fed with enough examples to assure convergence supported by hoeffding bound. Proposed technique processes each example only once and computes the dissimilarities just for leaf nodes when they are tested for splitting or aggregation thus speeds up the clustering process.

3.2 Splitting Criteria

In the original ODAC, the splitting criteria were given as follows:

$$(d_1 - d_0)|d_1 + d_0 - 2d^{-}| > \in_k \tag{5}$$

$$\textit{where } d_1 \textit{ is the maximum dissimilarity}$$
$$d_0 \textit{ is the minimum dissimilarity}$$
$$d^{-} \textit{ is the a verage of all dissimilarities}$$

Thus whenever this condition is satisfied the cluster is split into two children with cluster centers as $x_{1,}$ *and where* $d_1 = d(x_1, x_2)$. And the remaining variables are assigned to the clusters from which they are near in addition to the use of τ parameter to determine how long we check for the diameter before deciding to split. But this method results in duplicate clusters as shown in this figure:

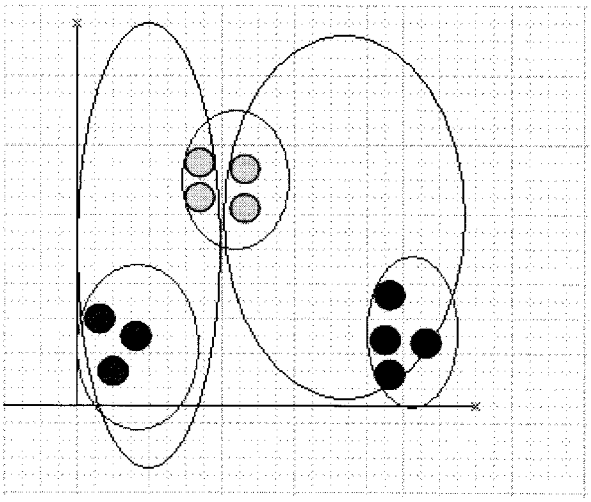

Figure 1. Strict splitting and duplicate clusters

If we consider the maximum distance as splitting criteria, we end up having duplicate clusters or a large hierarchy of clusters. Therefore, in this extended version of ODAC, the major change and difference is that the maximum dissimilarity of EODAC is considered in splitting phase. The second and third maximum dissimilarities are also considered. Thus, if $d_2 = d(x_3, x_4) and\ d_3 = d(x_5, x_6)$ are the second and third maximum distances, we look at the participating points for the second and third distances. And if it happens that any point is shared with the points of the maximum dissimilarity, the distance $d^{\sim}$ from this variable and the other end of the maximum dissimilarity is measured. And if it is satisfied the condition of being greater than the half of d_1, then its variables are also considered as pivots and hence we end up having three or four actual clusters. For instance, let $d_1 = d(x_1, x_2)$, $d_2 = d(x_2, x_3)$, since there is a variable x_2 that is shared with the first and second maximum dissimilarities, a distance $d^{\sim} = d(x_1, x_3)$ is found and test for the following condition:

$$d^{\sim} \geq d_1 \underset{select}{\Longrightarrow} x_3 \tag{6}$$

Which means x_3 is a third cluster center and it causes the algorithm to split the current cluster into three clusters with x_1, x_2, x_3 as cluster centers. As shown in figure 2 by using this proposed splitting condition, we can achieve the exact clusters in a computationally efficient manner.

```
Procedure TestSplit
Input: a cluster $C_k$
Output: Boolean value stating whether $C_k$ was split or not
1: let $d_0, d_1, d_2, d_3, d_4, d^-, d^\sim$ be the distances defined previously.
2: if $d_1 - d_2 > \epsilon_k$ or $\tau >\in_k$ then
3:   if $(d_1 - d_0)|d_1 + d_0 - 2d^-| >\in_k$ then
4:     if $d_2$ and $d_3$ share points with $d_1$ and $d^\sim$
           Satisfies $d^\sim \geq d_1$ then
             Create clusters $C_{x1,}C_{x2}, C_{x3,}C_{x4}$
5:     else if $d_2$ share points with $d_1$ and
           $d^\sim \geq d_1$ then
             Crate clusters $C_{x1}, C_s, C_{x2}$
6:         for each remaining variables $x_i \in C_x$
           Not yet assigned assign its to nearest
             Cluster
           Else
               return false
       End
 return true
```

Figure 2. Splitting criteria

This procedure tests for splitting criteria and then splits either into two, three, or four childes at once.

3.3 Aggregating Clusters

Sometimes previous splitting decisions are no longer survives so a back tracking is required. The aggregation condition is same as ODAC that we monitor the diameter of children's, and if the difference between the diameters of a cluster C_k and its children is greater than hoeffding bound. The children are aggregated to their parent and the statistics are calculated again. Figure 3 shows the aggregating procedure.

```
Procedure TestAggregate
Input: a cluster $C_k$
Output: Boolean value stating if cluster $C_k$ was aggregated or not
1: Update dissimilarities of $C_k$ if needed and
   Find $\in_{max} = \max(\in_k, \in_{ki})$
2:  if $diam(C_k) - \sum_i diam(C_{ik}) > \in_{max}$ then
      Cut  children turning     into leaf node
       And reset statistics
      return  true;
    endif
  Return false;
```

Figure 3. TestAggregate

In this procedure line 1 calculates the summary statistics, and hoeffding bound for the current node needed for the dissimilarity. And compares the resulting hoeffding bound with parent's one to decide either to aggregate or not.

```
Procedure EODAC
Input: a set of streaming time series
 $X = \langle x_1, x_2, x_3, .., x_n \rangle$
Output: a hierarchical clustering structure S with leafs as clusters $L = \langle l_1, l_2, l, .., l_n \rangle$
1: repeat
     read new example $X^t$ and update statistics
     On leafs L;
2:  for each leaf $l_k$ not yet tested do
      Update dissimilarities and Hoeffding
      Bound   for this leaf
     If TestSplit($l_k$) or TestAggregate($l_k$) then
           Announce new structure S;
      endif
until EOF
```

Figure 4. Process of EODAC

This is the whole procedure which reads the stream as snapshots over time and updates the summary statistics at leaf nodes, and then goes throw all nodes to test for splits or aggregates.

The major benefit for this extended version is execution time and memory space because it eliminates duplicate clusters and incorrect splits. When splitting every time, the space and computations required are reduced. For example in a worst case when a node with n variables splits into two Childs with (n-1) and (1) element, the space required to store dissimilarities is reduced by one. Therefore, proposed method does not need to compute an additional dissimilarity.

4. CONCLUSION

This paper proposed an enhancement to the ODAC by addressing its two major weaknesses as duplicate clusters and incorrect splits. The added splitting criteria also speeds up the clustering process as it still uses the same data structures exist in ODAC. Future work is to implement and evaluate the proposed clustering method for applying to the real application domains.

REFERENCES

References from Journals:

Mohamed, A. N., Leckie, C., and Udaya, P., 2008. An Efficient Clustering Scheme to Exploit Hierarchical Data in Network Traffic Analysis. *IEEE Transactions on Knowledge and Data Engineering, 20(6). pp.752-767.*

Pereira, R, R., Gama, J, and Pedro, J., 2007. Hierarchical Clustering of Time Series Data Stream. *IEEE Transactions on Knowledge and Data Engineering.*

References from Books:
Han, J., and Mich K,. *Data Mining Concepts and Techniques*. 2nd Edition, pp. 482-496.

References from Other Literature:
Babu, B, S., Motwani, D, R., Widom, J., 2002. Models and Issues in Data Stream Systems, *ACM Symposium on Principles of Database Conference.*

Domingos, P., Hulten, G., 2000. Mining High-Speed Data Streams. *ACM International Conference on Knowledge Discovery and Data Mining.*

Zhang,T R. Ramakrishnan , and Livny, M., 1996. Birch: An Efficient Data Clustering Method for very Large Databases, processing of *ACM International Conference on Management of Data.*, pp. 103-114.

ACKNOWLEDGEMENTS

This research was supported by a grant(#07-KLSG-C05) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Construction & Transportation of Korean government, and a grant from the Personalized Tumor Engineering Research Center(PTERC) and the Korea Science and Engineering Foundation(KOSEF).