ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

An Efficient Hierarchical Clustering Algorithms Approach Based on Various-Widths Algometric Clustering

Prof. K.Jeyalakshmi, S.Shanmugapriya

Associate Professor, PG & Research Department of Computer Science, Hindustan College of Arts and Science, Coimbatore, Tamilnadu, India

Research Scholar, PG & Research Department of Computer science, Hindustan College of Arts and Science, Coimbatore, Tamilnadu, India

ABSTRACT: Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster.In this paper proposed an experimentally evaluate the performance of different global criterion functions in the context of hierarchical agglomerative clustering algorithms and compare the clustering results of partitional algorithms for each one of thecriterion functions The proposed method builds the solution by initially assigning each points to its own cluster and then repeatedly selecting and merging pairs of clusters, to obtain a single allinclusive cluster.The key parameter in agglomerative algorithms is the method used to determine the pair of clusters to be merged at each step. Experimental results obtained on synthetic and real datasets demonstrate the effectiveness of the proposed various width cluster method.

KEYWORDS: Change detection (CD), hierarchical clustering, hyperspectral (HS) images, multiple changes, multi-temporal analysis, remote sensing.

I. Introduction

Cluster analysis divides data into groups (clusters) that are meaningful, useful, or both. If meaningful groups are the goal, then the clusters should capture the natural structure of the data. In some cases, however, cluster analysis is only a useful starting point for other purposes, such as data summarization. Whether for understanding or utility, cluster analysis has long played an important role in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.Cluster analysis provides an abstraction from individual data objects to the clusters in which those data objects reside. Additionally, some clustering techniques characterize each cluster in terms of a cluster prototype; i.e., a data object that is representative of the other objects in the cluster. These cluster prototypes can be used as the basis for number of data analysis or data processing techniques.

Given a set of objects *O* and a query object *q*, a k-nearest neighbor (*k*-NN) query returns from *O* the *k* closest (most similar) objects to *q*. For example, in an image database, a user might be interested in finding the images most similar to a given query image. k-NN is a classical problem and has applications in a wide range of domains such as pattern recognition [1], outlier detection [2],intrusion detection [6], classification [8] and spatial databases, to name a few.

The traditional approach to compute exact results, the called Exhaustive k-NN (*Ek*-NN) approach requires scanning the whole data set and finds k-NNs by computing the distance between q and every object in *O*. This results in high computational cost [10]. To address this, alarge body of research has focused on pre-processing the data set (i.e., constructing an index) with an aim to compute k-NNs by accessing only a part of the data set. The techniques can be loosely classified into two categories: i) tree-based indexes; ii) flat indexes.

II. Related work

In [2] authors proposed a novel distance-based outlier detection algorithm, named DOLPHIN, working on disk-resident datasets and whose I/O cost corresponds to the cost of sequentially reading the input dataset file twice, is presented. It is both theoretically and empirically shown that the main memory usage of DOLPHIN amounts to a small fraction of the dataset and that DOLPHIN has linear time performance with respect to the dataset size. DOLPHIN gains efficiency by naturally merging together in a unified schema three strategies, namely the selection policy of objects to be maintained in main memory, usage of pruning rules, and similarity search techniques. In [3] authors discussed a distance-based outlier detection method that finds the top outliers in an unlabeled data set and provides a subset of it, called outlier detection solving set, that can be used to predict the outlierness of new unseen objects, is proposed. The solving set includes a sufficient number of points that permits the detection of the top outliers by considering only a subset of all the pair-wise distances from the data set. In [4] authors defining outliers by their distance to neighboring data points has been shown to be an effective non-parametric approach to outlier detection. In recent years, many research efforts have looked at developing fast distance-based outlier detection algorithms. Several of the existing distance-based outlier detection algorithms report log-linear time performance as a function of the number of data points on many real low-dimensional datasets. However, these algorithms are unable to deliver the same level of performance on high-dimensional datasets, since their scaling behavior is exponential in the number of dimensions. The authors presenteda RBRP fast algorithm for mining distance-based outliers, particularly targeted at high-dimensional datasets. RBRP scales log-linearly as a function of the number of data points and linearly as a function of the number of dimensions. In [5] authors proposeda simple nested loop algorithm that in the worst case is quadratic can give near linear time performance when the data is in random order and a simple pruning rule is used. To test the algorithm on real high-dimensional data sets with millions of examples and show that the near linear scaling holds over several orders of magnitude.In [6] authors had considered thetechniques for Nearest Neighbour classification focusing on; mechanisms for assessing similarity (distance), computational issues in identifying nearest neighbours and mechanisms for reducing the dimension of the data.In [7] authors proposed a two novel techniques: (i) an automatic identification of consistent and inconsistent states of SCADA data for any given system, and (ii) an automatic extraction of proximity detection rules from identified states. During the identification phase, the density factor for the k-nearest neighbors' of an observation is adapted to compute its inconsistency score. Then, an optimal inconsistency threshold is calculated to separate inconsistent from consistent observations. During the extraction phase, the well-known fixed-width clustering technique is extended to extract proximity-detection rules, which forms a small and most-representative data set for both inconsistent and consistent behaviors in the training data set.

III. Proposed algorithm

Hierarchical clustering method with each point being considered a cluster and recursively combine pairs of clusters (subsequently updating the inter-cluster distances) until all points are part of one hierarchically constructed cluster. In this work includes three tasks such as i) Data Cleaning; ii) Cluster-Width Learning; iii) Agglomerative Hierarchical clustering. The proposed phase architecture shows in below Fig. 1.

A. *Data Cleaning:*

The data cleaninglarge amount of irrelevant entries, which are required to be removed from the web log for preparation prior to data mining. To incomplete the lacking attribute values, lacking certain attributes of interest, or containing only aggregate data. The pre-processing method follows the data conversion approach that facilitates of data clustering. The unsupervised raw dataset is first partitioned into three groups: (1) a finite set of objects, (2) the set of attributes (features, variables) and (3) the domain of attribute. For each groups in the dataset, a decision system is constructed. Each decision system is subsequently split into two parts: the training dataset and the testing dataset. Each training dataset uses the corresponding input features and fall into two classes: normal (+1) and abnormal (−1).

DOI: 10.15680/IJIRCCE.2016. 0407023

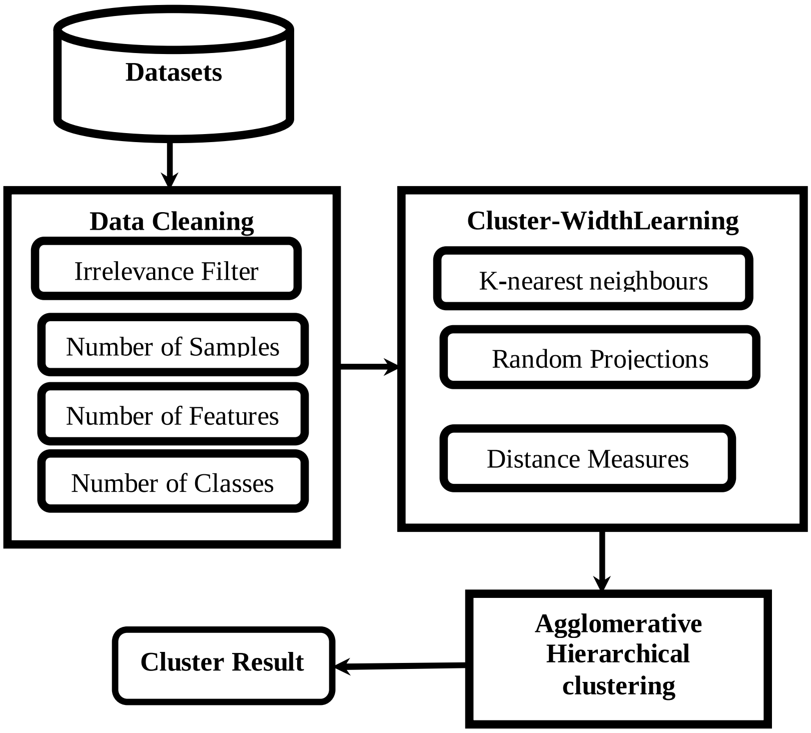

Fig: 1 Proposed Architecture Flow diagram

B. *Cluster-Width Learning*

The pre-processed data be the function of *k*-nearest neighbours for the object H_i, *clsWidth* be the function computing the width (radius) of nearest neighbours value, where the width is the distance between the object H_i and the farthest object among its neighbours.To find the appropriate global width approach randomly draw a few objects from *D*, $H=\{H_1,H_2,\ldots,H_r\}$ where r <<|*D*|, and for each randomly selected object, the radius of its k-nearest neighbours is computed, and the average is used as a global width for *D* as follows:

$$W = \frac{1}{r}\sum_{i=1}^{r} clsWidth(NN_k(H_i), H_i) \quad (1)$$

The learning process partitions a data set into a number of clusters using a large width to resolve the issue of clustering the sparsely distributed objects in n-dimensional space. However,large clusters from dense areas will be created such as clusters C_2 and C_3. Therefore, each large cluster whose size exceeds a user-defined threshold (maximumcluster size) will be divided into a number of clusters using a width that suits the density of that cluster. This process continues until the sizes of all clusters are less than or equalto the user-defined threshold (Mean and Trail Error method).

C. *Agglomerative Hierarchical clustering*

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

The hierarchy within the final cluster has the following properties:

- Clusters generated in early stages are nested in those generated in later stages.
- Clusters with different sizes in the tree can be valuable for discovery.

IV. Pseudo code

```
Step 1: Assign each object to a separate cluster.
Step 2: Calculate the various width cluster for each nearest neighbor using eq. (1).
Step 3: Construct a distance matrix using the distance values.
Step 4: Look for the pair of clusters with the shortest distance.
Step 5: Remove the pair from the matrix and merge them.
Step 6: Evaluate all distances from this new cluster to all other clusters, and update the matrix.
Step 8: End.
```

V. Results

For evaluating the proposed work, in Fig. 2, the cluster result of original DARPA data set [9]. This sample contains 52,488 objects. 241 objects are labelled as attacks while the rest are labelled as normal.

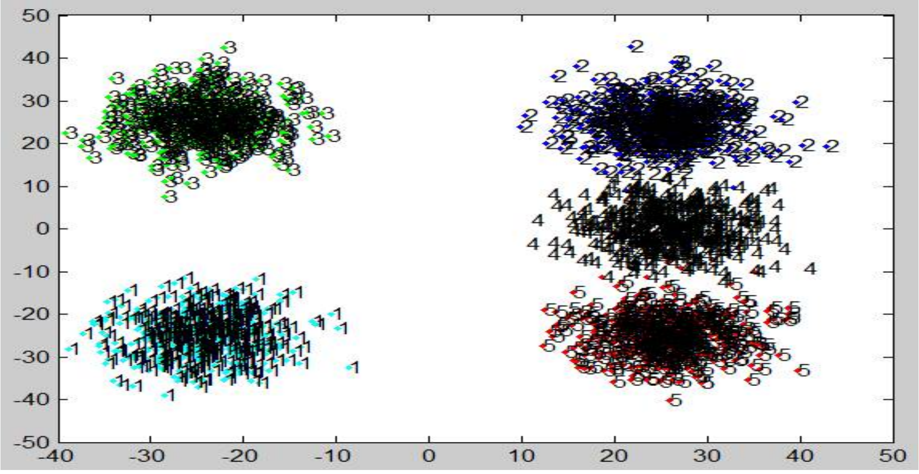

Fig 2: Agglomerative Hierarchical Cluster Result

VI. Conclusion and Future Work

In this paper, proposed anoptimalAgglomerative Hierarchical Cluster approach based on various-widths clustering called *AHWC*. This approach is able to produce compact and well-separated clusters from high dimensional data of various distributions. The proposed method is possible to discover the difference among overlapping clusters linking among the data. Moreover, the proposed approach is designed in an unsupervised way with difference constraints.

In future work, we intend to enhance the Clustering algorithm to develop the experimental methods for non-linear optimization to control the growth of tree attributes of the result unsuperviseddata.

References

1. G. Shakhnarovich, T. Darrell, and P. Indyk, “Nearest-neighbor methods in learning and vision,” IEEE Trans. Neural Netw.,vol. 19, no. 2, p. 377, Feb. 2008.
2. F. Angiulli and F. Fassetti, “DOLPHIN: An efficient algorithm for mining distance-based outliers in very large datasets,” ACMTrans. Knowl. Discovery Data, vol. 3, no. 1, p. 4, 2009.
3. F. Angiulli, S. Basta, and C. Pizzuti, “Distance-based detection and prediction of outliers,” IEEE Trans. Knowl. Data Eng., vol. 18, no. 2, pp. 145–160, Feb. 2006.
4. A. Ghoting, S. Parthasarathy, and M. E. Otey, “Fast mining of distance-based outliers in high-dimensional datasets,” DataMining Knowl. Discovery, vol. 16, no. 3, pp. 349–364, 2008.
5. S. D. Bay and M. Schwabacher, “Mining distance-based outliers in near linear time with randomization and a simple pruning rule,”in Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2003, pp. 29–38.
6. P. Cunningham and S. J. Delany, “k-nearest neighbour classifiers,” Multiple Classifier Systems, pp. 1–17, 2007.
7. A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, “An unsupervised anomaly-based detection approach for integrity attacks onscada systems,” Comput. Security, vol. 46, pp. 94–110, 2014.
8. A. Shintemirov,W. Tang, and Q. H. Wu, “Power transformer fault classification based on dissolved gas analysis by implementingbootstrap and genetic programming,” IEEE Trans. Syst., Man Cybern. C, Appl. Rev., vol. 39, no. 1, pp. 69–79, 2009.
9. M. V. Mahoney and P. K. Chan, “An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection,” in Proc. 6th Int. Symp. Recent Adv. Intrusion Detection, 2003, pp. 220–237
10. B. S. Kim and S. B. Park, “A fast k nearest neighbor finding algorithm based on the ordered partition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. TPAMI-8, no. 6, pp. 761–766, Jun. 1986.