Fast Dimension-based Partitioning and Merging clustering algorithm

Tamer F. Ghanem [a,*], Wail S. Elkilani [b], Hatem M. Abdelkader [c], Mohiy M. Hadhoud [a,1]

[a] Department of Information Technology, Faculty of Computers and Information, Menofiya University, Shebin El Kom, Menofiya, Egypt
[b] Department of Computer Systems, Faculty of Computers and Information, Ain Shams University, Cairo, Egypt
[c] Department of Information Systems, Faculty of Computers and Information, Menofiya University, Shebin El Kom, Menofiya, Egypt

ARTICLE INFO

ABSTRACT

Clustering multi-dense large scale high dimensional numeric datasets is a challenging task duo to high time complexity of most clustering algorithms. Nowadays, data collection tools produce a large amount of data. So, fast algorithms are vital requirement for clustering such data. In this paper, a fast clustering algorithm, called Dimension-based Partitioning and Merging (DPM), is proposed. In DPM, first, data is partitioned into small dense volumes during the successive processing of dataset dimensions. Then, noise is filtered out using dimensional densities of the generated partitions. Finally, merging process is invoked to construct clusters based on partition boundary data samples. DPM algorithm automatically detects the number of data clusters based on three insensitive tuning parameters which decrease the burden of its usage. Performance evaluation of the proposed algorithm using different datasets shows its fastness and accuracy compared to other clustering competitors.

1. Introduction

Clustering is the process of finding groups of similar objects based on some similarity measures. Clustering techniques are successfully applied in many applications in biology, finance and marketing domains [1,2]. Traditional clustering algorithms are not suitable for processing huge amounts of data with a large number of attributes which are collected in many industrial fields. This is reasoned by their high processing time overhead and quality limits [3].

Curse of dimensionality is a well-known problem in clustering high dimensional datasets [4–7]. As the number of dataset dimensions increases, measuring distance between dataset objects becomes more meaningless. This is reasoned by increasing the number of dimensions causes points to spread out until they are nearly equidistant from each other which completely masks clusters. One of the problems that arises due to this curse is the probability that a cluster may only exist in some subset of data attributes. Another problem is that these subset of attributes may differ from one cluster to another [2].

Clustering algorithms require tuning parameters to be set. Clustering results should be insensitive to these parameters values.

If not, clustering process will be more worsen because of the increasing of the burden on the end user. Parameter is considered as insensitive if results could be obtained under a range of parameter settings. This ensures that clustering results are not changed under slight variations of these parameter values. Furthermore, the number of clustering parameters should be small as possible [3].

The main contribution of this work is to introduce a fast clustering algorithm called Dimension-based Partitioning and Merging (DPM). This algorithm is used for clustering numeric, multi-dense, large scale, and high dimensional datasets along with automatic clusters number detection using three insensitive tuning parameters. The motivation of this work is that most popular clustering algorithms used for such datasets are time consuming due to the high time complexity of the used processing operations. The proposed algorithm starts with partitioning the data space into small dense partitions by visiting each dimension values only once using dimension histogram. In the next stage, noise is filtered out based on dimensional densities of the obtained partitions. At last, merging stage based on a small number of samples called boundary samples is applied to construct data clusters. Observations demonstrate that the proposed algorithm is extremely fast with the capability of detecting number of clusters compared to other clustering algorithms.

The rest of this paper is organized as the following: Section 2 presents some literature review on clustering algorithms. Section 3 discusses the proposed clustering algorithm methodology. Experimental results along with a comparison with other clustering

* Corresponding author. Tel.: +20 1004867003.
E-mail addresses: tamer.ghanem@ci.menofia.edu.eg (T.F. Ghanem),
wail.elkilani@gmail.com (W.S. Elkilani), hatem6803@yahoo.com
(H.M. Abdelkader), mmhadhoud@yahoo.com (M.M. Hadhoud).
[1] On leave as Dean of the Canadian Higher Institute for Engineering and Business New Cairo, Egypt.

algorithms are presented in Section 4 followed by some conclusions in Section 5.

2. Related work

Traditional Clustering algorithms can be categorized into four groups: partition clustering, hierarchical clustering, density-based clustering and grid-based clustering [8]. In partitioning clustering category, data is split into k partitions (clusters) using an iterative relocation process in order to enhance the similarity of each partition. Hierarchical clustering is the next category where data is clustered in a hierarchical fashion using bottom-up approach or top-down approach. Density-based clustering is another category where clusters are identified using a growing scheme based on a density threshold that neighborhood objects must exceed. Although arbitrary shapes and noise can be detected using these clustering methods, they are not scale well with the size of dataset. Finally, Grid-based clustering algorithms partition data space into grid of cells which are combined to form clusters based on neighborhood relations. It is distinguished by fastness but it does not work efficiently in high dimensional space.

Unlike Traditional clustering algorithms, subspace clustering has been proposed to overcome problems arisen from curse of dimensionality phenomena by constructing clusters based on similarities on a subset of attributes (subspace). As a result, some samples may be assigned to multiple clusters. If each sample is assigned to only one cluster based on some subset of attributes, subspace clustering will be called projected clustering [3].

Most of projected and subspace algorithms have the capability to detect clusters and noise which are closely related [9]. Also, most of these clustering algorithms are controlled by many sensitive parameters which are difficult to set by users (e.g., projected clusters number or the average number of relevant attributes of projected clusters). Moreover, they are less effective for detecting clusters with few relevant attributes found in high-dimensional spaces [10].

A grid-based clustering algorithm for high dimensional data is presented in [11]. Its basic concept depends on mapping data onto a multi-dimensional space followed by a linear transformation to the feature space. Finally, a grid-clustering method is invoked to identify clusters.

GGCA [12] is another grid-based clustering algorithm where divisible and the agglomerative clustering approaches are combined into single framework. Data is divided into a number of grid cells with optimal sizes followed by merging process to aggregate all data objects to form final clusters. In addition, GGCA does not require users to input any control parameters.

AGRID [13] is a grid-density based algorithm for clustering high dimensional datasets. It partitions space into hyper-rectangular cells based on interval division of dataset dimensions. Due to its computational overhead, enhancements are added to improve its efficiency in [8] by reducing the processed neighborhood cells number and using a novel density compensation method.

Random Projection subspace clustering is proposed in [14]. Data is projected to randomly generated subspaces then EM algorithm is used to generate several groups of clusters. These groups are combined later using a similarity matrix to form the final clusters.

P3C [15] is another projected clustering algorithm which effectively discovers clusters depending on the computation of what is known as cluster cores using few number of parameters. Moreover, it has the capability of clustering both numerical and categorical datasets.

Another algorithm is proposed in [16] with the aim of detecting parallel clusters in databases. This algorithm is based on density detection in the neighborhood in order to construct clusters. This algorithm targets clustering sequence data instead of generic data.

STATPC [17] proposes a problem formulation that aims at extracting axis-parallel regions that stand out in data in a statistical sense. All axis-parallel regions are represented by a reduced statistical set of significant regions R. The task of representing R is done using an approximation algorithm called STATPC to reduce computational complexity. STATPC uses error probabilities that are accepted by the user as parameters. This means that no prior knowledge about the data is required for parameters to be set.

PROCLUS [18] is one of projected clustering techniques which based on partitional approaches to form clusters. Each cluster is initially represented by one of its points called "medoid" together with a subset of dataset attributes. The algorithm iteratively tends to minimize the average within-cluster spreading. PROCLUS has two parameters which are hard to guess and its results depend on its initialization.

DOC [19] is density-based projected clustering algorithm which constructs one cluster at a time. An arbitrary pivot point is selected with some random number of points to form tentative cluster. An attribute is considered relevant if these points are within threshold distance from pivot point on this attribute. This process is repeated until a cluster is formed using points that fall within distance threshold from a pivot point on all attributes considered as relevant. DOC results is sensitive to its parameter values which leads to some difficulties in using it in real life datasets. To reduce processing time overhead, FASTDOC [19] is proposed to reduce search time using three heuristics, but clustering accuracy is no longer guaranteed.

In [20], a density-based clustering algorithm is introduced. It begins by identifying the different densities in the dataset and ranking its objects. Next, data is project into a space with one more dimension. Finally, a density based clustering is invoked using the reverse-nearest-neighbor of the objects.

DBSCAN [21] is another density-based projected clustering algorithm. This algorithm forms clusters by combining samples in neighborhood together if their density is high enough above some threshold. This density threshold is specified by the user. To improve searching process, R*-tree structure is used for more efficient queries. DBSCAN time complexity is $O(n \log n)$ (where n is the number of samples in dataset) [22].

Another density based clustering algorithm is presented in [23]. This work extends DBSCAN algorithm for clustering datasets with deferent clusters densities. This algorithm attempts to find density based clusters that may not be separated by any sparse region depending on the detection of significant change in adjacent regions densities.

A hybrid clustering algorithm called GRPDBSCAN is presented in [24]. This work combines grid-based and density-based approaches to enhance DBSCAN algorithm in detecting noise and arbitrary shape clusters, in addition to select its parameters automatically.

Clustering numeric large scale high dimensional datasets is a time consuming process. Moreover, sensitive parameters are another issue that increases the clustering difficulties. This work proposes a fast clustering algorithm with three insensitive parameters. Time complexity is greatly reduced by simplify the processing operations in different stages of the proposed algorithm. First, small dense partitions are identified on the fly during successive processing of each dimension. Then, noise reduction stage is invoked followed by partitions merging process. This merging process depends on a small number of boundary samples generated in the initial stage. No grid cells or heavily neighborhood processing is invoked. As a result, processing time overhead is greatly reduced for such clustering. Performance evaluation of DPM compared to other clustering algorithms shows its extreme fastness.