Efficient Algorithms for Clustering Data and Text Streams

D

Panagiotis Antonellis
University of Patras, Greece

Christos Makris
University of Patras, Greece

Yannis Plegas
University of Patras, Greece

Nikos Tsirakis
University of Patras, Greece

INTRODUCTION

Clustering data may differ depending on the variety of aspects such as the dimensionality, cluster size and noise. Also the criterion for clustering may be based on the context in which the data is given. For instance web click-stream data are characterized by the vast amount of data and the fast evolvement they have. For such large-scale data, the clustering process becomes essential for reducing the problem size and time to analyze them.

For many recent applications, the concept of the data stream is more appropriate than that of the dataset. By nature, a stored dataset is an appropriate model when significant portions of the data are queried repeatedly, and update operations on the data are relatively infrequent. In contrast, a data stream is an appropriate model when a large volume of data is arriving continuously. It is either unnecessary or impractical to store all arriving data in some forms. For example, web logs in a web server belong to data streams. In these applications, decisions have to be made at times when important events occur. In the data stream model, data points can only be accessed in the order of their arrivals and random access is not allowed. The space available to store data streams is often not enough because of the volume of unbounded streaming data points. How to effectively organize and efficiently extract useful information from a data stream is a problem that has attracted researchers from many fields. Although there are many algorithms for data mining, they are not designed for data streams (Zhou, Cai, Wei, & Qian, 2003).

In this article we describe the most recent algorithms for clustering data and text streams, focusing mainly on identifying the main categories of clustering techniques, the advantages and drawbacks of each approach and how they adapt into the modern text stream environments. Finally, we introduce a novel technique based on multi-level clustering in order to analyze in real time, user's behavior, based on distributed web click streams.

BACKGROUND

There already exists a plethora of interesting works that have been published to address data streams in the data mining community. These proposals have tried to adapt traditional data mining technologies to the data stream model. Clustering, as one of the most important parts of data mining process, has also gained research attention, regarding its usefulness when processing data streams. *Data stream clustering* is usually studied with the objective of minimizing the memory space, and processing time required. For such algorithmic designs use of single-pass algorithms that consume a small amount of memory is critical (Antonellis, Makris, & Tsirakis, 2009). Moreover, the real-time requirements and the evolving nature of data streams makes effective

DOI: 10.4018/978-1-4666-5888-2.ch170

clustering a challenging research problem. There are many approaches in the literature, some of them trying to extend the classical clustering algorithms, such as *k-median* and *k-means* to data stream applications and some others trying to process the data streams using novel techniques, tailored to the challenges of data streams

An interesting variant of the problem of data stream clustering is when dealing with *text streams*. This version of the problem has a number of interesting applications such as topic detection and tracking, user characterized recommendation, trend analysis etc. Text stream analysis has attracted great interest and is of great practical value. Clustering text data streams is an important part of the data mining process and appears to have several practical applications such as news group filtering, text crawling, document organization and topic detection. There are many approaches in the literature trying to produce better results from the existing ones like adaptive feature selection (Gong, Zeng, & Zhang, 2011), efficient streaming text clustering (Zhong, 2005), topic models over text streams (Banerjee & Basu, 2007), categorical data stream clustering (Aggarwal & Yu, 2010) and semantic smoothing models (Liu, Cai, Yin, & Wai-Chee Fu, 2007; Xiaodan, Zhou, & Hu, 2006; Zhou, Hu, Zhang, Lin, & Song, 2006).

DATA STREAM CLUSTERING

Distance Based

The models in this category are mainly extensions based on the classical clustering algorithms and utilize a distance function between the incoming data points for grouping them into different clusters. One of the earliest and best known practical clustering algorithms for data streams is BIRCH (Zhang, Ramakrishnan, & Livny, 1997). BIRCH is based on a novel heuristic that computes a pre-clustering of the data into so-called clustering features and then clusters this pre-clustering using an agglomerative (bottom-up) clustering algorithm. Another well-known algorithm in this category is STREAMLS (O' Callaghan, Meyerson, Motwani, Mishra, & Guha, 2002), which partitions the input stream into chunks and computes for each chunk a clustering using a local search algorithm from (Guha, Koudas, & Shim, 2001). STREAMLS is slower than BIRCH but provides a clustering with much better quality (with respect to the sum of squared errors). The authors in (Guha, Meyerson, Mishra, & Motwani, 2003) propose an agglomerative hierarchical clustering approach that requires very small space and efficiently clusters the input data stream in one pass, based on the *k-median* approach. More precisely, the proposed algorithm uses the divide-and-conquer approach to divide the data into pieces, clusters each of pieces and then again clusters the centers obtained (where each center is weighted by the number of points assigned to it). Finally STREAMKM++ (Ackermann et al., 2012) is a *k-means* based algorithm that maintains a small sketch of the input using the merge-and-reduce technique, i.e. the data is organized in a small number of samples, each representing $2^i m$ input data points (for some integer i and a fixed value m). Every time when two samples representing the same number of input data points exist, the algorithm takes the union (merge) and creates a new sample (reduce).

Density and Grid Based

Density-based clustering has been long proposed as another major clustering algorithm. Due to its nature, it fits on the context of the data stream clustering, because it can find arbitrarily shaped clusters, it can handle noises and is an one-scan approach that needs to examine the raw data only once. Further, it does not demand a prior knowledge of the number of clusters k, as the *k-means* algorithm does. The main proposed models of this category are the D-Stream (Chen, 2007) and the MR-Stream (Wan, Keong, Dang, Yu, & Zhang, 2009). D-Stream is a grid-based stream clustering algorithm. The synopsis information is contained in the grid cells, which can be considered to play the role of microclusters. For each incoming data point, a hash table determines which grid an incoming data point belongs to. As the D-Stream algorithm is also based on the fading model of stream data, the importance of each grid cell, which is an aggregate of the weights of data points in the cell, diminishes over time if there are no incoming data points. For offline generation of clustering results, D-Stream periodically removes sparse grid cells in order to compute clusters. In parallel MR-Stream provides a hierarchical, multi-resolution view of clusters at any time. Thus, MR-Stream is able to cluster at different resolutions in the offline com-