ENHANCED ONLINE-OFFLINE VARYING DENSITY METHOD FOR DATA STREAM CLUSTERING

MARYAM MOUSAVI

THESIS SUBMITTED IN FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

KAEDAH PELBAGAI KETUMPATAN ATAS TALIAN LUAR TALIAN DAN DI TAMBAH BAIK UNTUK PENGELOMPOKAN ARUS DATA

MARYAM MOUSAVI

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH IJAZAH
DOKTOR FALSAFAH

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

29 October 2018

MARYAM MOUSAVI
P65600

ACKNOWLEDGEMENT

I am most grateful and indebted to ALLAH, for all his blessings that made everything possible for me and gave me power to overcome all the obstacles I encountered during my studies.

I would like to express my gratitude and special appreciation to my research supervisor, Prof. Dr. Azuraliza Abu Bakar, whose guidance, patience, and encouragement during my research project were valuable.

I would like to thank all my faculty staffs for their help and support throughout the research phase. Thanks also to my colleagues and friends for encouraging me during my research with their comments and suggestions.

I also wish to express my special appreciation to my beloved parents. No words can express how grateful I am to them for all of the sacrifices that they have made on my behalf. Their prayers for me mean everything.

Last but not least, I would like to convey my love and appreciation to my beloved husband, Mohammadmahdi Vakilian who has always been supportive of me, especially during my studies.

ABSTRACT

The term data stream refers to a potentially bulky, continuous and fast sequence of information. As opposed to traditional data forms which are unchanging and static, a data stream has its own unique characteristics; it is massive, even potentially infinite, and is, moreover, continuous, requires a single scan, and dynamically changes over the time, thus requiring a rapid response usually in real time. The process of data stream clustering involves extracting valuable patterns in real time from dynamic streaming data in only a single scan, which can be very challenging. However, due to the nature and characteristics of data stream, traditional clustering techniques cannot be applied. Thus, it has become crucial to develop new and improved clustering techniques. The existing clustering techniques are generally categorized into five main categories: hierarchical, partitioning, grid-based, density-based and model-based. Density-based techniques are the remarkable category in clustering data streams. These techniques consider the dense areas of objects as clusters where they are separated with low density sparse areas in data set. They can detect the clusters with arbitrary shapes and can handle noises. In addition, these methods do not require a priori knowledge of the numbers of clusters. The main objective of this research is to propose a new online-offline density-based clustering method for data stream with varying density. In the online phase, the summary of data is created (often known as micro-clusters) and in the offline phase, this synopsis of data is used to form the final clusters. Finding the accurate micro-clusters is the goal of online phase. When a new data point arrives, the procedure of finding the nearest and best fit micro-cluster is the time consuming process. This procedure can lead to increase the execution time. To address this problem, a new merging algorithm is proposed that can lead to decrease the execution time. For maintaining a limited number of micro-clusters, a pruning process is applied along with the summarization process. In the existing methods, this pruning process takes too long time to remove micro-clusters whose do not receive objects frequently that cause to increase the memory usage. In this thesis, to solve this problem, a new pruning algorithm is introduced to reduce the memory usage. Another problem with density-based methods is that they use global parameters in the data sets with varying density that can lead to dramatic decrease in the clustering quality. In our work, to create final clusters, a new density-based algorithm that works based on only MinPts parameter is proposed for increasing the clustering quality of data sets with varying density. The performance evaluation on both synthetic and real data sets illustrates the efficiency and effectiveness of the proposed method. The experimental results show that our method can increase the clustering quality in data sets with varying density along with limited time and memory usage.

ABSTRAK

Aliran data merujuk kepada urutan maklumat yang berpotensi besar, berterusan dan pantas. Berbeza dengan bentuk data tradisional yang tidak berubah dan statik, suata aliran data mempunyai ciri-cirinya yang tersendiri; ia adalah besar-besaran, and juga berpotensi tidak terhingga, dan lebih-lebih lagi, berterusan, memerlukan imbasan tunggal, and berubah secara dinamik dari masa ke masa, maka memerlukan tindak balas pesat biasanya dalam masa nyata. Proses pengelompokan aliran data melibatkan pengekstrakan corak bernilai dalam masa nyata daripada pengaliran dinamik data dalam hanya imbasan tunggal, yang mungkin menjadi sangat mencabar. Walau bagaimanapun, disebabkan oleh tabii dan ciri-ciri aliran data, teknik pengelompokan tradisional tidak boleh digunakan. Maka, adalah penting untuk membangunkan teknik pengelompokan yang baru dan lebih baik. Teknik-teknik pengelompokan yang sedia ada biasanya dikategorikan kepada lima kategori utama: hierarki, pembahagian, berasaskan grid, berasaskan ketumpatan dan berasaskan model. Teknik-teknik berasaskan ketumpatan adalah kategori yang luar biasa dalam pengelompokan aliran data. Teknik-teknik ini mengambil kira kawasan yang padat dengan objek sebagai kelompok, di mana mereka dipisahkan oleh kawasan ketumpatan rendah yang jarang kepadatan dalam set data. Ia boleh mengesan kelompok dengan bentuk rambang dan boleh mengendalikan bunyi. Di samping itu, algoritma ini tidak memerlukan pengetahuan *a priori* mengenai bilangan kelompok. Objektif utama kajian ini adalah untuk mencadangkan suatu kaedah pengelompokan 'online-offline' baru yang berasaskan ketumpatan untuk aliran data dengan pelbagai ketumpatan. Dalam fasa dalam talian, ringkasan data dicipta (sering dikenali sebagai kelompok mikro), dan dalam fasa offline, sinopsis data ini digunakan untuk membentuk kelompok akhir. pencarian cluster mikro yang tepat adalah tujuan fasa dalam talian.apabila titik data yang baru sampai, prosedur pencarian untuk cluster mikro yang paling dekat dan berpadanan memakan masa. prosedur ini akan menambah masa penggiraan. Untuk menyelesaikan masalah inin, algoritma penggabungan yang baru dicadangkan untuk mengurangkan masa penggiraan. Untuk mengekalkan bilangan kelompok mikro yang terhad, proses pemangkasan digunakan bersama-sama dengan proses rumusan. Dalam kaedah yang sedia ada, proses pemangkasan ini mengambil masa yang terlalu lama untuk mengeluarkan kelompok mikro yang tidak menerima objek kerap kali yang menyebabkan peningkatan dalam penggunaan memori. Dalam tesis ini, untuk menyelesaikan masalah ini, kaedah pemangkasan baru diperkenalkan untuk mengurangkan penggunaan memori. Masalah kedua dengan kaedah berasaskan ketumpatan adalah bahawa ia menggunakan parameter global dalam set data dengan pelbagai ketumpatan yang boleh membawa kepada penurunan dramatik dalam kualiti kelompok. Dalam kajian ini, untuk mewujudkan kelompok akhir, suatu kaedah baru berasaskan ketumpatan yang berfungsi berdasarkan hanya atas parameter MinPts dicadangkan untuk meningkatkan kualiti pengelompokan set data dengan pelbagai ketumpatan. Penilaian prestasi di kedua-dua set data sintetik dan sebenar menggambarkan kecekapan dan keberkesanan kaedah yang dicadangkan. Keputusan ujikaji menunjukkan bahawa kaedah ini boleh meningkatkan kualiti pengelompokan dalam set data dengan pelbagai ketumpatan bersama-sama dengan masa dan penggunaan memori terhad.

TABLE OF CONTENTS

CHAPTER IV METHODOLOGY

CHAPTER V PROPOSED METHOD

CHAPTER VI EXPERIMENTAL RESULTS AND DISCUSSION

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

β	Outlier threshold
μ	Integer weight of a potential micro-cluster
λ	Decay factor
r_t	Radius threshold (neighbourhood around micro-clusters)
t	Current time
h	Horizon (Range of the window)
T	Timestamp
v	Stream speed (number of incoming points per time unit)

LIST OF ABBREVIATIONS

MC	Micro-cluster
PMC	Potential micro-cluster
OMC	Outlier micro-cluster
MinPts	Minimum number of points
InitN	Initial *N* objects of the data stream

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Latest technological advances in computer hardware, software, applications, networks and data warehouses, resulted in the generation of large data sets. With ever increasing data sets, precise methods for extracting knowledge are highly demanded. Data analysis plays an indispensable role for the better understanding of various phenomena. Cluster analysis as a primitive exploration in data analysis with limited knowledge is still in research mode and being developed across a wide variety of communities. Applications of clustering algorithms in research are ubiquitous with typical examples including intrusion detection, web document mining, wireless sensor network and image analysis. However, generation of immense data sets by these applications, cluster analysis can be daunting. For this reason, novel methods to manage enormous data for clustering and extracting useful knowledge have become particularly important. Data stream clustering has been proposed as an efficient method to process and analyze enormous data sets.

Data stream clustering has recently attracted attention for emerging applications that involve large amounts of data (Aggarwal et al. 2004a; Aggarwal et al. 2004b; Liu et al. 2008; Antonellis, Makris & Tsirakis 2009; Chen & Liu 2009; Gama et al. 2009; Lühr & Lazarescu 2009; Tu & Chen 2009; Aggarwal 2009a; Aggarwal & Yu 2009b; Aggarwal 2009c; Tu et al. 2012; Singh & Meshram 2017). It is an effective strategy to precisely treat and minimize problems associated with huge data sets. Data stream clustering is defined as the clustering of data that arrive continuously such as session records in network log file, multimedia data, financial transactions, etc. Data stream clustering is usually studied under the data stream

mining and the objective is given as a sequence of points; construct a good clustering of the stream with regard to memory and time constraints. In other words, high scaled data set is converted to the stream of data whereby input is a sequence of n points with an integer number k. Similarly, output is k centers in the set of the n points to minimize the sum of distances from data points to their closest cluster centers assuming that clusters are spherical shaped. One of the most important solutions in the high scaled data sets clustering is data stream clustering with certain time and memory limitations. On the other hand, processing data in the limited time and memory causes some difficulties such as concept drift and visiting data once. The main goal of data stream clustering is to achieve the same results if the clustering method can be extended to the entire data sets without time and memory constraints. It means that in the absence of these limitations, it is still possible for data stream clustering method to pass several times over the data and improve accuracy as traditional data clustering.

Intrusion Detection System (IDS) is an instance of typical application for data stream clustering, whereby its log files are high scale and high dimensions. Hence, considering the nature of data, concept drift and IDS should be managed efficiently. On the other hand, monotonous data segments which are records with the same value for a big segment of data may arise some difficulties during the data processing. Accuracy is one of the most critical measurements which are generally defined by ratio of false positive and false negative alarms. Therefore, we need to design efficient algorithms to scan the data once and extract hidden patterns inside it. Evolving, visiting data once, accuracy in intrusion detections and space limitations are major related issues in intrusion detection systems. In addition, monotonous data segments evolve some difficulties during the data processing as well. In this context devising new technique and framework such as data stream clustering to overcome the mentioned drawbacks is required.

All data stream clustering methods attempt to find framework and algorithm to improve the quality of clustering results, but still there are certain requirements that need to be improved in terms of the quality and computation time. In addition, the treatment on different data density is also another critical problem in this area.

Therefore, there is a necessary need for studying and developing effective and efficient ways of clustering data stream, which is the main motivation of this study.

1.2 BACKGROUND

A constantly moving, massive in volume and high-velocity sequence of data information is simply known as data stream (Yadav & Nair 2015; Ding et al. 2016). Common examples of data stream encompass engineering data, scientific data, time series data, as well as data extracted from a multi-disciplinary dynamic field such as sensor network monitoring, telecommunication archives, website analysis, stock exchange, meteorological research, credit card, and e-business (Han, Kamber & Pei 2006; Yogita & Toshniwal 2013; Yang, Yi & Yu 2014; Desai & Gaikawad 2017). Mining the data stream in one pass to excerpt high-quality mining outcomes in a real-time environment of continuous incoming data may prove to be a highly difficult task (Augustine 2017). Nevertheless, the effectiveness of data stream clustering process in data mining has garnered huge interest (Ding et al. 2013; Silva et al. 2013). Data are processed during clustering and information or objects within the data are segregated into clusters. The course aims to group corresponding items in a cluster while different items are band together in the other various clusters (Madraky, Othman & Hamdan 2014). The clustering procedures used to process massive data are essentially basic methods, which are applicable in data mining, pattern identification, and machine learning. Streaming methods are required to cluster massive data due to the better performance of streaming access compared to random access to the vast amount of data kept in the hard disks or in data stream form (Ackermann et al. 2012). However, due to the data stream characteristics such as massive in size and expand gradually, the conventional clustering methods are not applicable. Thus, developing new and improved clustering techniques are becoming more crucial. Of late, there have been numerous discussions on numerous views as well as facets of streaming data clustering process and a number of procedures and techniques were recommended. Generally, there are five categories of the clustering techniques, namely hierarchical, partitioning, grid-based, density-based and model-based (Han, Kamber & Pei 2006; Nguyen, Woon & Ng 2015).

Density-based techniques are the notable category in clustering data streams that possess quite a few significant advantages for data clustering such as i) the capability to identify arbitrary-shaped clusters, ii) ability to handle noise and iii) they require just the one time to scan raw data. Apart from that, such algorithms do not require prior knowledge of the number of clusters (*k*) unlike *k*-means algorithms that need to be given the number of clusters in advance (Loh & Park 2014; Nguyen, Woon & Ng 2015).

For static data sets, the OPTICS algorithm is the solution for density-based clustering algorithms that are dependent on parameters (Ankerst et al. 1999). It contains two concepts for organizing points: i) the core distance and ii) the reachability distance. In the clustering process, the reachability distance and spatial positioning order the organized points to be added to the clustering structure list. It includes a comprehensive parameter setting for a single clustering structure. Unfortunately, the OPTICS is not suitable for use in data streams although it is perfect for parameter-dependent problems and is capable of detecting overlapping clusters and arbitrary shaped clusters.

A two-phase online-offline scheme density-based method known as DenStream (Cao et al. 2006) has been developed to cluster evolving data streams. In the first phase, this method uses the fading window model to create a synopsis of the data. Then, in the second phase, the synopsis of the data stored from first phase is utilized to provide the clustering result. This method can handle arbitrary shaped clusters but it is not capable of handling the datasets with different densities.

An improvement of the DBSCAN algorithm (Ester et al. 1996)known as LDBSCAN (Duan et al. 2007) has also been proposed. This algorithm uses the concept of local density-based clustering. It is able to detect density-based local outliers and noise. However, this algorithm does not work well in data streams.

The D-Stream method (Tu & Chen 2009) has also been proposed, which is able to make automatic and dynamic adjustments to the data clusters without user specification with regard to the target time horizon and quantity of clusters. This

technique makes separated grids to map recent arriving data. A decay factor is utilized with the density of each data object in order to specify which data are new and which are less important (old). The D-Stream method can handle noisy data in limited time. However, it is incapable of processing the datasets with different densities.

Similar to D-Stream, MR-Stream (Wan et al. 2009) creates cell partitions in the data space. Whenever a dimension is divided in half, a single cell goes through another division to form 2^d subcells, where d is the dimension of the dataset. The division process can be set to a maximum limit by a user-defined parameter. The divided cells are stored on a quad tree structure that allows for data clusters to be created at different resolution levels. The MR-Stream method allocates all new data into the appropriate cells at every time stamp interval during the online phase and also updates the summarized data. This method is able to discover clusters at multiple resolutions whenever there is change in the underlying clustering scheme. However, MR-Stream cannot process in limited time and memory.

An improvement of the DenStream algorithm is rDenStream (Li-xiong et al. 2009), which is a three-phase clustering method. In this algorithm, previously discarded unimportant clusters are stored in a transitory memory. This approach ensures that this data has the chance to form clusters and increase the clustering accuracy. rDenStream can handle a huge number of outliers and its first two phases are comparable to those of DenStream but it has an additional phase known as the retrospect. This phase allows the algorithm to learn from the discarded data to increase its accuracy. From an experimental comparison, rDenStream outperforms DenStream in the initial phase. However, this algorithm requires more time and memory as compared with DenStream because it processes and saves the historical buffer.

Density Variation Based Spatial Clustering of Applications with Noise—DVBSCAN—is DBSCAN algorithm of various density that could manage the dissimilarity of local density inside a cluster (Ram et al. 2010). This algorithm computes the growing cluster density mean; next, it computes the core object's cluster density variance that ought to be expanded, taking into consideration the density of its ε-neighbourhood in regards to the cluster density mean. In the event that the core

object's cluster density variance is lower than or identical with the value of the threshold as well as meeting the cluster similarity index, it shall permit the spreading out of the core object. However, DVBSCAN algorithm has high execution time.

OPCluStream (Wang et al. 2012) is another density-based method to cluster data stream. This method utilizes a tree topology for organizing points and directional pointers to link all related points together. This technique is able to detect arbitrary shaped and overlapping clusters. However, this method does not work well in varying density environments.

DBSCAN-DLP (DBSCAN based on Density Levels Partitioning) (Xiong et al. 2012) uses the partitioning of the density level to define the respective clusters' parameters to mechanically discern the clusters of different density. The first phase of this approach divides the dataset into a various level of density according to the statistics of its density difference. Next, the ε of every density stage is specified. The algorithm final phase implemented DBSCAN to execute clustering at every stage of density according to its respective ε in order to obtain the outcome of the clustering. The density difference data is computed according to closest distance of k-nearest neighbors, namely the space from data object *p* and its k-nearest neighbours. The k-nearest neighbors' space will decides the k-neighborhood density; hence, the shorter the space of k the higher the density of the cluster. The points that have the same or nearly the same densities belong to the Density Level Set (DLS). DBSCAN-DLP method generalizes the conventional DBSCAN to discover clusters of dissimilar densities by means of segregating the level of density. DBSCAN-DLP is a data clustering method employing a two-pass clustering with high execution time.

SOStream (Isaksson, Dunham & Hahsler 2012) spontaneously acclimatizing the boundary for density-based clustering to distinguish a composition inside high-speed evolving data streams. There is only online phase in this procedure where the merges and updates are conducted. SOStream operates based on the competitive learning, which is dedicated for SOMs (Self Organizing Maps) (Kohonen 1982) where a winner impacts its adjoining neighbourhood. A winner cluster of a newly arrived data point is delineated according to Euclidean spacing of the current micro-clusters.

In the case that the measured spacing is a lesser amount compared to the dynamically delineated edge, the micro-cluster is deemed a winner micro-cluster, thus fresh data object is included to it. At the same time, the micro-cluster neighbours of the winner group is also affected. Neighbours are delineated according to DBSCAN procedure MinPts factor. The procedure involves finding the clusters and the winner that are overlying each other. The spacing of the overlying clusters to the winning cluster are measured. If the cluster spacing is a lesser amount compare to the merge-threshold, the cluster is combined with the winner. A fresh micro-cluster will be established in the case that the new object is not included to any prevailing micro-cluster. SOStream in a dynamic manner generates, combines, and eradicates clusters by online means. This density-based grouping procedure is capable to adjust its edge according to the data stream. However, SOM technique requires a lot of time thus unsuitable for data stream clustering.

Another density-based clustering technique for streaming data is the DSCLU (Namadchian & Esfandani 2012). DSCLU uses micro-clusters to detect suitable clusters, focusing on localizing dominant micro-clusters on the basis of their neighbours' weight. It is able to detect clusters in varying density environments but it works with same radius to form micro-clusters.

A different method was presented by Khani et al. (Khani et al. 2013) who suggested Algorithm for Clustering Spatial Data (ACSD) with dissimilar densities. Generally, the notion of ACSD is to create data chart whereby boundaries between objects are added, thus, objects within a cluster will be located in a linked part that parallel to the cluster; whilst objects of other clusters are nearly separated. At first, ACSD forms an initial chart and gradually expands the chart through responses from every single point to its neighborhood points. These neighborhoods and responses are decided upon through an investigation of the received responses. Following the creation of the stable chart, post-processing of the chart subsequently form the clusters. The object's central and boundary are decided by computation of angles between edges. Next, DBSCAN clustering procedure is conducted in order to group the data accordingly. ACSD performance on high dimensional data might be unsatisfactory because of the objects' central and boundary computation structure.

Furthermore, there are three parameters to compare with the two parameters of DBSCAN.

FlockStream (Forestiero, Pizzuti & Spezzano 2013) is a density-based clustering method on the basis of a bio-inspired model. This method is based on the flocking model (Kennedy et al. 2001) in which agents are micro-clusters and they work independently but form clusters together. This method combines the online and offline levels because agents create clusters at whatever time. As a matter of fact, this technique does not require to conduct the offline clustering to obtain the clustering outcomes. Even though, an outlier agent is formed by the procedure to manage the noise, a distinct plan on how and when to eradicate the outliers from the agents list is still lacking. Therefore, this method has high memory usage.

An effective variant of DBSCAN algorithm, which is known as ISDBSCAN was suggested by (Cassisi et al. 2013). This alternative procedure offers a novel density task according to the kNN-stratification (k-nearest neighbors) and Influence function. The dataset stimulant is ranked according to density task and segregated into strata of declining density following the average k-adjusted influence function. Finally, the outliers are heuristically identified. In the course of the outliers' identification stage, any information concluded is implanted into a fresh zone of remnant through the addition of a new dimension. Every value that resides in this dimension signifies the distances summation of the influence zone, thus the segregation of clusters with dissimilar densities is alleviated. A revised algorithm is then employed to the current remnant dataset. However, detecting border points is rather difficult for the ISDBSCAN algorithm.

In (Louhichi, Gzara & Abdallah 2014) the authors proposed a new extension of DBSCAN algorithm for detecting of multi-density of data. This algorithm considers the different value for radius of clusters based on *k*-nearest neighbours curve. It has some problems to manage the variety of density within the same cluster and has high execution time.

ExDBSCAN (Ghanbarpour & Minaei 2014) is a multi-density clustering algorithm that is based on greedy technique. This algorithm gets just one parameter, MinPts, and the value of radius increases gradually to take the real neighbourhood. This algorithm has high execution time that makes it unsuitable for data stream.

M-DBScan (Vallim et al. 2014) has been proposed for change detection in data stream. This method includes a density-based clustering step followed by a novelty detection process on the basis of the entropy level measures. The method utilizes two various kinds of entropy measures, where one considers the spatial distribution of data while the other models temporal relations between observations in the stream. However, it is not able to detect clusters in datasets with different densities.

ExCC (Bhatnagar, Kaur & Chakravarthy 2014) is an algorithm of a select and comprehensive clustering for varied data stream. Similar to most procedure, ExCC is an online-offline type. Online part retains the simplified within the grids while offline part generates the ultimate requested clusters. The arithmetical traits will be charted to the grid by the algorithm and the categorical traits are allotted granularities based on the precise values in the individual field sets. The algorithm of ExCC is comprehensive due to it utilization of pruning according to the data stream promptness instead of a window template, for example fading one. The stream presented by ExCC is either fast or slow according to the mean of the data points' arrival time in the stream of data. In addition, the clustering procedure is privileged because it employs the grid in order to distribute data. The algorithm also distinguishes noise in the offline part by means of hold and observe policy. When distinguishing actual outliers, data points are retain within the hold line that is reserved independently for every dimension. ExCC utilizes a user stipulated edge for identifying condensed and scarce grids. Noise was strains by this procedure by means of cell density and cluster density edge as indicated by user. The edge was approximated by the procedure according to the grid granularity, data dimension, and the mean number of objects in every grid. To produce clusters, a pool for dense and latest grids is considered by this procedure. From this pool, the dense adjoining grids are selected taking into consideration eight adjoining of every grid. Attributes equivalence are deliberated for categorical data. Nevertheless, such strategy of holding the line requires additional memory and extra

handling time because every dimension is delineated. Furthermore, a higher volume of memory need to be retained as well as additional time is required to handle those memory when the pool is utilized to preserve dense grids.

HDSDen (Jin-yin & Hui-hao 2015) is an online-offline clustering technique for heterogeneous data streams. This technique is based on density-based algorithm that uses mixed distance measurement method. In the online phase, two different methods are proposed to calculate the distances among the continuous and categorical objects and then the core points calculated. In offline step to create the final clusters, all the density-reachable points from the core points identified. Although HDSDen is applicable for continuous and categorical attributes but still is not suitable for data stream with varying density.

C_UStream algorithm (Tu 2015) has been proposed for indeterminate data stream clustering in sliding window of the grid area. Sample windows of every grid are registered by the algorithm. It then employs the hash table and related list of line for storing and querying the grid rundown statistic information, which enhance the procedure competency to a great extent. During the course of clustering, groups are made and attuned based on the mechanics of the grid integration. Outdated data will be removed on a regular basis while outlier data are eliminated by means of the removal mechanics of dynamic sporadic grid in order to manage the complication of time and space. However, due to the increasing amount of empty grids, C_UStream algorithm is incompatible for high-dimensional data which requires longer processing time.

A hybrid grid-based multi-density clustering procedure with online-offline phases is known as MuDi-Stream (Amini et al. 2016). The online stage retains the core mini-clusters, which are the summary facts of the progressing multi-density data. Meanwhile, the offline stage used the modified density-based clustering algorithm to produce the concluding clusters. The grid-based scheme operated as a buffer of the outlier to manage the noises and multi-density data to lessen the clustering integration process. The quantity of vacant grids escalates that causes an extensive period of

processing thus MuDi-Stream is inappropriate for high-dimensional data. In addition, this algorithm is sensitive to the value of input parameters.

DCSTREAM (Khalilian, Mustapha & Sulaiman 2016) approach is suggested in view of the iterative data fragments and observing the changes in the structure of the clustering for ordinal data streams by means of the vector model and divide and conquer of the *k*-means. An essential part of DCSTREAM is the utilization of dividing instance according to their stretch inside the vector model explanation that permits this technique to retain the clusters' overall assembly even in the absence of the whole cluster data saved in memory. Usually, performing a second phase of algorithm is not required. Experimental results of this algorithm show that the usage of batch processing is time consuming. In addition, DCSTREAM is not able to detect the non-convex shaped clusters.

Str-FSFDP (Chen & He 2016) is proposed as a density-based data streams clustering method for clustering numerical and categorical stream data. The technique uses ACC-FSFDP method for automatic detecting cluster centre in initial phase. This method applied regression analysis and residual analysis to determine the centre of mixed data. However this method does not work in limited time.

In (Al-Mamory 2016) the authors presented a new solution to cluster data sets with different densities by using a new technique to separate data depending on the density contained in them, then using a new sampling technique to get data with homogenous density. The data resulted from the separation and sampling are clustered by DBSCAN, and finally, KNN is applied on the core points resulted from the previews step with the dense data remaining after sampling. Although this algorithm is able to detect clusters in varying density data sets but it is not applicable in stream environments.

1.3 PROBLEM STATEMENT

Density-based clustering techniques consider the dense areas of objects as clusters where they are separated with low density sparse areas in dataset. These techniques

are able to recognize the arbitrary shaped clusters and can handle outliers. Also they do not require former information of number of clusters. Generally, data streams clustering methods consist of two phases: online and offline. In the former phase the summary of data is created as the micro-clusters and in the latter phase the method uses the synopsis of data that has been stored from the first phase to generate the final clusters which named macro-clusters. Finding the accurate micro-clusters is the goal of online phase. When a new data object arrives, the procedure of finding the closest and best fit micro-cluster is the time consuming process. This procedure can lead to increase the execution time. Lately, several density-based clustering methods have been proposed to cluster data stream (Amini et al. 2016; Chen & He 2016; Singh & Meshram 2017). Although these algorithms try to decrease the execution time by reducing the number of comparisons however it is still high to be useable for streaming data environments.

Another important problem in density-based methods is to keep the finite number of micro-clusters in their online phase. By passing the time, the number of micro-clusters rise that lead to increase the memory usage. To address this problem a pruning algorithm is often along with summarization process. The purpose of pruning algorithm is to remove micro-clusters that have not received the objects frequently and they have become outliers (Forrest 2011; Forestiero, Pizzuti & Spezzano 2013). Sometimes this pruning process takes long time to discard micro-clusters that cause to increase the memory usage (memory usage is calculated by the number of micro-clusters).

The important challenge in density-based methods is that they use global parameters in the datasets with varying densities that can lead to the dramatic decrease in the clustering quality. For example, as illustrated in the Figure 1.1, if the value of clustering radius is selected to be high, the five clusters can be detected as three clusters A, B, C (three small clusters C_1, C_2, C_3 are considered as one big cluster C). On the other hand, if this value is considered as a small value, three small clusters C_1, C_2, C_3 can be detected and two big clusters A and B are detected as noises (Esfandani, Sayyadi & Namadchian 2012; Aggarwal 2015; Tari et al. 2018).

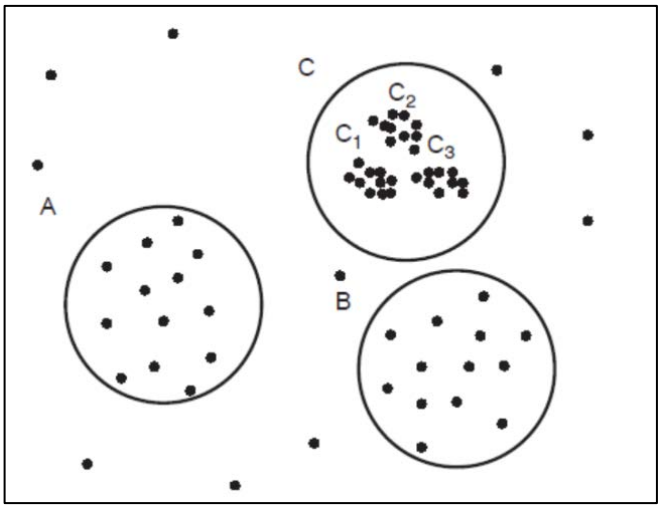


Figure 1.1 Dataset with varying densities

1.4 RESEARCH QUESTIONS

The main aim of this thesis is to propose the proper clustering method to extract the valuable knowledge from streaming data. The following questions should be investigated to complete this issue:

- Why do we require a proper merging algorithm in the online phase of data stream clustering methods?
- Why do we need a good pruning algorithm in the online phase of data stream clustering methods?
- What are the limitations of current density-based clustering techniques? Moreover, what are the effects of considering a new density-based clustering algorithm that is suitable for datasets with varying densities in the offline phase?

1.5 RESEARCH OBJECTIVES

Based on the problem statement presented in Section 1.3, the main objectives of this thesis are summarized as follows:

- To propose a new merging algorithm in the online phase of data stream clustering method to summarize the data stream and decrease the execution time.
- To propose a new pruning algorithm in the online phase of data stream clustering method that can lead to decrease the memory usage.
- To propose a new density-based clustering algorithm in the offline phase of data stream clustering method to handle the datasets with varying density.

1.6 RESEARCH SCOPE

It is essential to utilize the datasets that enable us to show and analyze the effectiveness and efficiency of the suggested method. In this thesis, both real and synthetic data sets are utilized. The most applicable and famous real data sets are collected grounded on literature. The synthetic data sets are generated synthetically. The reason behind use of this type of datasets is that the synthetic data sets can be created by controlling the number of data objects, and the number of clusters, with various distributions or evolution properties, they are utilized to assess the scalability in our experiments. Nevertheless, since synthetic data sets are generally rather various from real ones, we will basically utilize real data sets for the clustering evaluation.

1.7 RESEARCH METHODOLOGY

Figure 1.2 illustrates the research methodology of this study. The descriptions of all actions taken during this research from start to end are as follows:

Firstly, an extensive literature review is conducted, including all the pertinent works published till 2018. In this part of the thesis, the historical development of the methods relevant to this study is considered, with the aim of identifying the advantages and disadvantages of different techniques, thus revealing the areas where this field can benefit from further improvements. In this fundamental phase, only the articles published in renowned peer-reviewed journals are selected, with particular focus on those that provide strong evaluation methods. Comparison of different

studies and their approaches to the development of their respective techniques is also performed in this phase.

Based on the outcomes of the literature review performed in the previous phase, the main difficulties of density-based clustering methods for streaming data are identified. The objectives and the corresponding method of the research are determined next. In the proposed method an integrated structure is designed with combination of solutions to overcome difficulties in data stream clustering. So an online-offline method is proposed. In the online step, the summary of data is created as the micro-clusters by a new merging algorithm. These micro-clusters are pruned by a new pruning algorithm to keep the memory usage low. In the offline phase, the output of online phase applied to create a final cluster. In this step a new clustering algorithm is introduced to apply for datasets with varying densities.

Different data sets comprising the real and synthetic ones are chosen for evaluation purposes. Different evaluation methods including clustering quality, execution time, and memory are applied.

Finally, for the evaluation of results, a comparison among the newly developed method and the state of the art techniques are demonstrated.

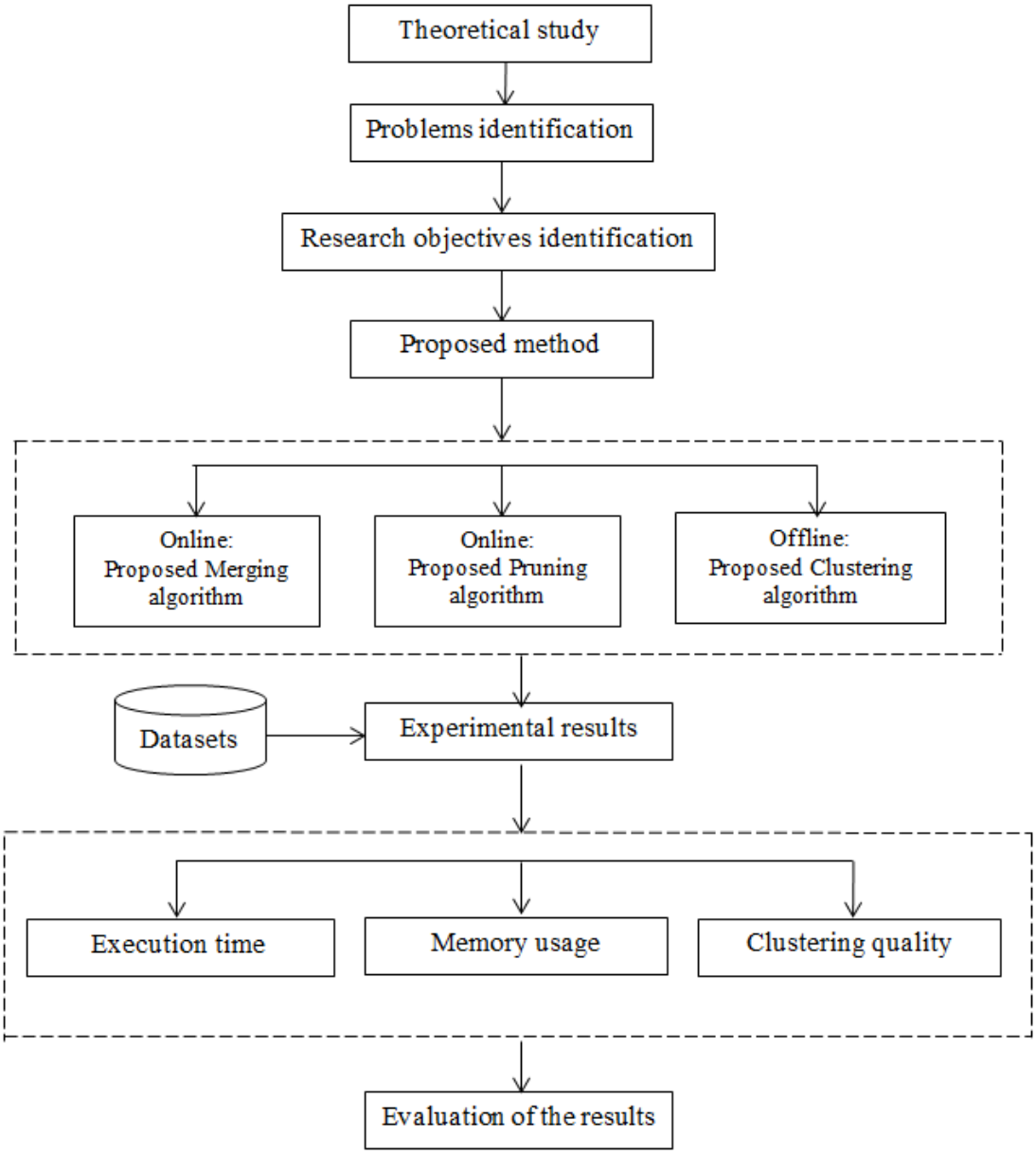


Figure 1.2 Research methodology

1.8 RESEARCH CONTRIBUTIONS

The main contribution of this research is to devise an effective data stream clustering method to get accurate results in those datasets which have different densities. The research contributions are summarized below:

- Following an extensive study of the existing clustering methods, their weaknesses are identified, allowing the present study to propose an enhanced density-based method for clustering varying density streaming data.

- Proposing a new merging algorithm for online phase of method to summarize the data stream in the form of potential micro-clusters and outlier micro-clusters.

- Proposing a new pruning algorithm for online phase of method to control the number of micro-clusters with the purpose of keeping the memory usage low.

- Proposing a new density-based algorithm for offline phase of method that works based on one parameter (MinPts) unlike the previous algorithm that is based on two parameter (MinPts and radius) that can solve the problem regarding to datasets with varying densities.

1.9 THESIS ORGANIZATION

The rest of this thesis is organized as follows:

Chapter II provides background knowledge of data stream mining techniques and algorithms. This chapter will give an overview of used algorithms, techniques and Framework in data stream mining. Clustering, classification, and association rules mining for data stream mining are major algorithms that will be discussed in this chapter.

Chapter III shows a summary and rather overall overview of various clustering methods on data streams. In addition, it possesses an extensive review on the available density-based techniques to cluster data streams. Also the extant density-based clustering methods for varying densities data sets are reviewed in this chapter.

Chapter IV explains the methodology of this research used to attain the research objectives. This chapter presents an overall overview of the proposed density-based clustering method for streaming data and a summary overview on its

structures. Also, in this chapter, various synthetic and real dataset are described. In addition, different evaluation metrics are presented.

Chapter V describes the proposed density-based clustering method. The new concepts applied in the proposed framework are explained in this chapter. Furthermore, the components of the new clustering method are described in detail.

Chapter VI shows the experimental results obtained by applying different evaluation methods on different datasets. In this chapter, the results are compared to several well-known methods in order to test the robustness of the proposed method.

Chapter VII concludes the research and reviews the main contributions of the research and also provides some ideas for the future works in this field.

1.10 CHAPTER SUMMARY

This chapter provided the introduction to the thesis. It briefly described the study background, outlined its main purposes, and provided the problem statement. The motivations behind the choice of this study topic along with the research questions were given in this chapter. Also, the study contributions and methodology of this research were delineated, and then thesis outline briefly was described.

CHAPTER II

BACKGROUND

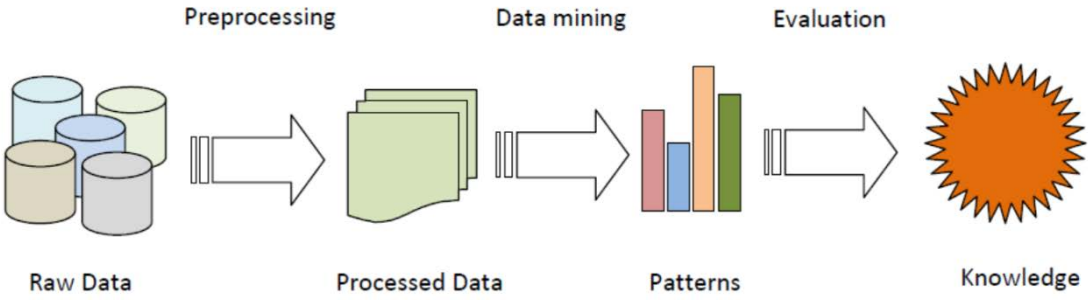
2.1 INTRODUCTION

Data stream mining as a category of the data mining is widely used to discovery useful knowledge such as hidden patterns from high scaled data sets. Although traditional data mining techniques are applicable in most applications but they should be tuned based on the problems in data stream mining. An overview of stream mining is given in Section 2.2 with regard to differences with traditional data mining. Data preprocessing in data mining is a critical stage, because we will be able to increase quality in final results; thus, a review on data preprocessing for data stream mining is presented in Section 2.3. A categorization of data pattern discovery in data stream is given in Section 2.4 that considering the main tasks in data mining, include frequent pattern stream mining, classification and clustering algorithms in data stream. Due to the importance of data stream clustering as the main topic in this thesis, Sections 2.5, 2.6, and 2.7 outline the data stream clustering methods along with the most important applications in data stream clustering area together with their general process and abstract framework. This chapter concludes by providing a summary in Section 2.8.

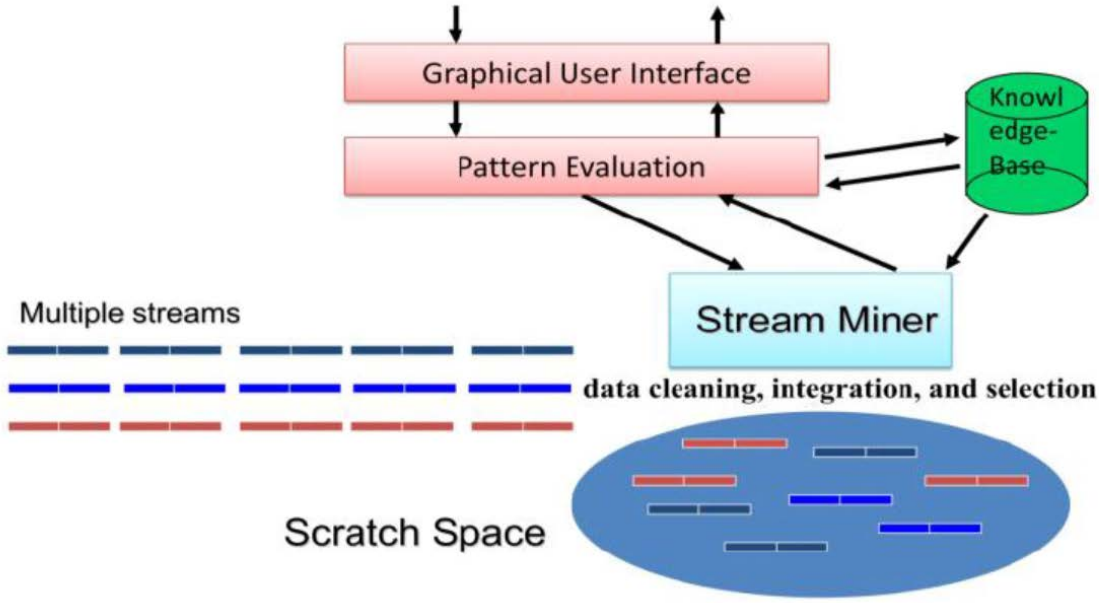
2.2 DATA STREAM MINING

Stream mining aims to discover useful information or knowledge from the stream data. In other word, extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data is the most prominent task in data stream mining (Gama et al. 2009). As shown in Figure 2.1(b), stream mining is one stage in knowledge discovery process (Figure 2.1(a))

which involved classification, clustering, frequent pattern discovery or probably a combination of these methods in stream miner component.



(a)



(b)

Figure 2.1 Knowledge discovery: (a) Knowledge Discovery in Databases (KDD) process (b) Stream knowledge discovery

The stream mining process is similar to the data mining process. The difference is usually in nature of processing that is dynamic meanwhile the size of data sets is high scaled. In traditional data mining, the data are often collected and stored in a data warehouse or database and processing method is static without constraints in time and memory. In stream mining, data processing method can be a substantial task, especially when stream includes evolving data or concept changes during the time, whereas visiting data is possibly once. Table 2.1 demonstrates the differences between traditional data mining and data stream mining.

Table 2.1 Differences of traditional data mining and data stream mining

Traditional Data Mining	**Data Stream Mining**
Number of scans is unlimited	Only one scan is possible
Processing time is unlimited	Processing time is restricted
Persistent relations	Transient streams
Random access	Sequential access
Available memory is unlimited	Bounded main memory
Only current state matters	Historical data is important
No real-time services	Real-time requirements
Low update rate	High update rate
Data at any granularity	Data at fine granularity
Accurate result	Approximate result
No distributed	Distributed potentially

Those differences making the process of data stream mining more difficult. Therefore, novel solutions and techniques should be adopted for stream data which can be categorized into two main groups: algorithm oriented and data oriented

methods (Mohamed Medhat, Arkady & Shonali 2005). Algorithm oriented methods refer to those techniques which modify traditional data mining methods or devise new algorithms in order to solve special challenges in data stream mining. On the other hand, data oriented methods are those methods in data stream processing that summarize the entire dataset or selecting a subset of the incoming stream to analyze. This categorization of techniques is depicted in Figure 2.2.

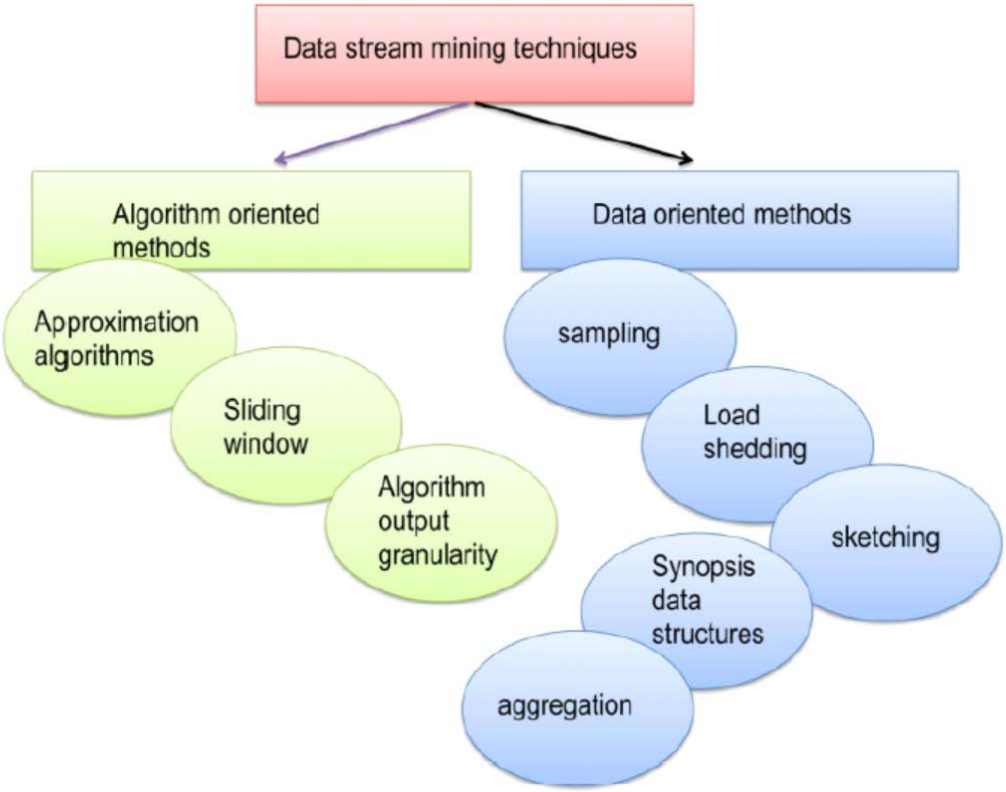


Figure 2.2 Data stream mining techniques

2.2.1 Algorithm Oriented Methods

Approximation algorithms

These kinds of solutions come from algorithm design with considering computationally hard problems. Approximation algorithms are methods used to find approximate solutions. They suffer from complex data structures or sophisticated algorithmic techniques which lead to difficult implementation problems. (Cormode &

Muthukrishnan 2003) proposed a new method for identifying hot items which occur more than some frequency threshold as it is based on approximation algorithms.

Sliding window

Make decisions based only on recent data of sliding window size w; consequently, an element arriving at time *t* expires at time *t + w*. This approach has been adopted in many techniques (Aggarwal & Yu 2006; Aggarwal & Yu 2008; Aggarwal & Yu 2009b; Aggarwal 2009c). Processing of data can be done in two manners: batch processing and incremental online processing. In the first strategy, entire data has been considered for mining, while in the second approach, data has been processed as it arrived one by one. Supporting concept drift is the main challenge during the stream mining in both batch and incremental online processing. Concept drift means that the concept about which data is being collected may shift from time to time. In the real world, concepts are often not stable but change with time. Typical examples of this are weather prediction rules and customers' preferences. Meanwhile, distinguishing between true concept drift and noise is problematic. Some algorithms are highly sensitive to concept drift and consider noise as the concept drift, while, some others are robustness to noise and are not sensitive to concept drift (Tsymbal 2004).

Algorithm output granularity

Multi-resolution models have three main stages. Mining followed by adaptation to resources and data stream rates represent the first two stages. Merging the generated knowledge structures when running out of memory represents the last stage. They have been used in clustering (Aggarwal et al. 2003), classification (Aggarwal et al. 2004a) and frequency counting (Gurmeet Singh & Rajeev 2002; Jiang & Gruenwald 2006).

2.2.2 Data Oriented Methods

Sampling

Maintain a set of s candidates in the reservoir, which form a true random sample of the element seen so far in the stream. As the data stream flow, every new element has a certain probability (S/N) of replacing an old element in the reservoir (Guha et al. 2002; Gurmeet Singh & Rajeev 2002; Guha et al. 2003). This method is not suitable in anomaly detection because object which is outlier may miss during sampling process.

Load shedding

It refers to the process of dropping a sequence of data streams. Load shedding has been used successfully in querying data streams. It has the same problems of sampling, namely it has side effect on accuracy and final results.

Sketching

This method is the process of randomly sample the incoming stream vertically. Sketching method has been applied in comparing various streaming data and in aggregate queries. Synopsis data structures mean forming summarization of data regarding to the process of employing summarization methods that are able to summarize the arriving data streams for further analysis. The wavelet analysis, histograms, and frequency moments were suggested as synopsis data structures. Since the synopsis of data does not indicate all the features of the data set, estimated results are created when utilizing such data structures (Aggarwal et al. 2004a; Aggarwal et al. 2004b; Aggarwal 2009c).

Aggregation

Aggregation is the process of calculating statistical measures such as means and variance that summarize the arriving data stream. Applying this aggregated data could

be utilized by the mining method. The issue with aggregation is that it does not carry out well with highly fluctuating data distributions (Mohamed Medhat, Arkady & Shonali 2005; Zhizhong, Ruichun & Yinzhen 2011).

2.3 PREPROCESSING FOR DATA STREAM MINING

Preprocessing affects quality of result in data mining; consequently, designing a special component for preprocessing data is significant during the data stream mining. This process is critical to the successful extraction of useful patterns from the data. The process may involve preprocessing the original data, integrating data from multiple sources, and transforming the integrated data into a suitable form for input into specific data mining operations. Collectively, we refer to this process as data preparation. Moreover, methods in traditional data mining should be specialized for data stream mining. Mostly, the initial stream data do not have the suitable format for applying data mining algorithm. Therefore, a substantial data preprocessing must be applied. The most common preprocessing tasks in stream data are data cleaning and filtering, data transformation and data reduction.

2.3.1 Data Cleaning

Due to the noisy, incomplete, inconsistent and duplicated data, it can be dirty data and it has to be cleaned before we do the mining. Renaming label, computing number of features where it is unknown (Lühr & Lazarescu 2009) and removing data with noise value are some of the tasks in data cleaning. It is important to recall the definition of noisy data where it is random error or variance in a measured variable which caused by faulty data collection instruments, data entry problems, data transmission problems, technology limitation and inconsistency in naming convention (Han & Kamber 2006). Data cleaning includes the following tasks:

- Missing or unfilled values. The different procedures below can be used depends on the applications:

    a. Ignore the object by removing it.

b. Assign default value to feature.

c. Compute and predicate value for specific feature by clustering.

- Identify and smooth out the noisy data.
- Correct the inconsistent data.
- Resolve redundancy caused by data integration.

2.3.2 Data Transformation

The data obtained from different sources (normalizing) and also with different types (transforming) therefore it should be converted to monotonous data type and normalized by employing a specific method. Generally, there are three main methods for normalizing:

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \quad (2.1)$$

- z-score normalization (μ: mean, σ: standard deviation)

$$v' = \frac{v - \mu_A}{\sigma_A} \quad (2.2)$$

- Normalization by decimal scaling

$$v' = \frac{v}{10^j} \quad (2.3)$$

where j is the smallest integer such that: $max(|v'|)<1$

Determining parameters such as maximum or minimum values for normalization method in data stream mining is a difficult task since we do not have the whole data set in hand; thus, selecting the suitable method depends on the application.

2.3.3 Data Reduction

Complex data stream mining may take a very long time to run on the complete data set; thus, a reduced representation of the data set that is much smaller in volume leads to produce the same (or almost the same) analytical results. There are some effective strategies for data reduction including feature selection, projection and data selection.

Feature selection

In streaming data the concept of irrelevant or redundant features are now restricted to a certain period of time. Features previously considered irrelevant may become relevant, and vice-versa to reflect the dynamics of the process generating data. While in standard data mining, an irrelevant feature could be ignored forever whereas in the streaming setting we still need to monitor the evolution of those features. Recent work based on the fractal dimension (Barbara & Chen 2000; Zhizhong, Ruichun & Yinzhen 2011) could point interesting directions for fractal dimension research and introduced the interaction between dimensionality reductions in the time decaying high dimensional stream environment and proposed the on-line fractal dimensionality reduction algorithm. Its experiments over a number of real data sets illustrate the effectiveness and efficiency provided by this approach. Selecting a minimum set of features such that the probability distribution of different classes given the values for those features are as close as possible to the original distribution given the values of all features. (Asbagh & Abolhassani 2009) proposed a feature-based data stream clustering method. It is a novel one-pass method for data stream clustering which benefits from feature selection in its body. In this method all clusters exist in the same feature space in the same time even though this space may vary over time. It continuously ranks features based on compactness and separateness measures and removes unimportant features using an automatic algorithm. It is clear that employing this method is time consuming; consequently, it affects speed up.

Projection

HPStream (Aggarwal et al. 2004b; Aggarwal 2009c) uses technique called projected clustering. For each cluster, a subset of features is considered that optimizes a quality criterion to that cluster. It is obvious that this subset of features may be different from

the other different clusters. In other words, two different clusters may have same, overlapping, or disjoint subset of features. The drawback of this technique arises from these differences which results in complicated interpretation of all clusters at once.

Data Selection

Reduce data volume by choosing alternative, smaller forms of data representation such as histogram and sampling (Guha et al. 2003; Giovanni, Kostas & Theodoros 2008). Accuracy is the most important challenge in this method.

2.4 PATTERN DISCOVERY IN DATA STREAM MINING

Nowadays, many applications generate tremendous and potentially infinite volumes of data particularly in real time status, whereby this stream of data is massive, fast changing and temporally ordered. It is obvious that pattern discovery from stream data with these characteristics is challenging. Traditional data mining techniques such as frequent pattern discovery, classification and clustering must be modified to apply in data stream with mentioned challenges.

2.4.1 Frequent Pattern Stream Mining

An important task in many data mining applications is the creation of the frequent pattern and association rule which satisfies minimum support and confidence criteria. This is particularly important in applications such as Web click stream mining, network traffic monitoring and wireless sensor network. Association rule is a concept in the form of X➔Y where X and Y are two frequent item sets in transactional database ($X \bigcap Y = \varnothing$) with two main parameters (minimum support and minimum confidence) which are specified by user. Support of the rule s is the percentage of records that contains both X and Y in the database and confidence of the rule c is defined as the percentage of records containing X that also contain Y. Based on data stream characteristics, new issues arise that need to be considered when developing association rule mining methods for stream data.

Table 2.2 Challenges in frequent pattern stream mining

General Challenges	Category of solutions	Strength	Weakness
Data processing model (Yunyue & Dennis 2002)	Landmark	Represent an entire history of stream data from landmark to present	Not suitable for real time applications
	Time fading	Support evolving data	Too many user specified parameters are needed
	Sliding window	Support time and space limitations	Do not support evolving data
Memory management	Information collected is least but enough information to generate association rules (Yang & Sanver 2004)	Support memory constraint	Degrade in accuracy
	Compact data structure (Gurmeet Singh & Rajeev 2002; Li, Lee & Shan 2004; Li & Chen 2009)	Increased efficiency	Increased complexity
One pass algorithms	Exact frequent item sets finding (Yang & Sanver 2004)	Exact answer	Two scans or mine only short items
	Approximate frequent item sets finding (Gurmeet Singh & Rajeev 2002; Li, Lee & Shan 2004)	Efficient in memory and time consumption	Approximate answer
	Updating association rule (Yun et al. 2004)	Support evolving data and concept drift	Cost of computation
Resources aware	Control parameter to control output rate (Gaber, Zaslavsky & Krishnaswamy 2004)	Support memory constraints	Cost of computation
	RAM-DS (Teng, Chen & Yu 2004)	Support memory constraints	Cost of computation

Traditional association rule mining algorithms are not suitable for data stream processing because they are developed to work on static data and, thus, cannot be employed directly to mine association rules in stream data. The first and the most famous algorithm for association rule mining in traditional database is Apriori (Rakesh et al. 1993) but Apriori-based algorithms require several scans to process data; consequently, it is not suitable for data stream environments, in which visiting data is possible only once. Another popular algorithm in traditional databases is FP-growth (Jiawei, Jian & Yiwen 2000) which used a tree data structure and thus achieved a higher performance but still it is not suitable for data stream because it needs to scan data twice in constructing the tree. Based on the literatures from the existing researches in frequent pattern mining, the general issues and challenges in frequent pattern mining for data stream can be summarized as in Table 2.2.

Notice that the different data stream application environments arise different challenges and issues which are listed below:

- Distributed Environment

  In this kind of applications, data comes from multiple remote sources; whereby communication cost and parallel updating must be taken into account in this kind of applications (Liu et al. 2006).

- Multidimensional Stream Data

  In this situation we need to consider information in different directions such as a network of sensors which gathers data about where some of these data are in contradiction of each other (Chung-Ching & Yen-Liang 2005).

- Timeline Query

  In some applications, user maybe interested in getting association rules based on the data available during a certain period of time. Therefore, the storage structure needs to be dynamically adjusted to reflect the evolution of item set frequencies over time. How to efficiently store the stream data with timeline and how to efficiently retrieve them during a certain time interval in response to user queries is an important issue (Giannella et al. 2003).

- Online interactive processing

  In some applications, users may need to modify the mining parameters during the processing period. In processing data streams, there is no specific stop point during the mining process. Therefore, how to make the online processing interactive according to user inputs before and during the processing period is another important issue (Veloso et al. 2003).

2.4.2 Classification Data Stream Mining

There exist emerging applications of data streams that require data classification, such as network intrusion detection and web click streams analysis. In many data stream mining applications, the goal is to predict the class or value of new instances in the data stream given some knowledge about the class membership or values of previous instances in the data stream. Machine learning techniques can be used to learn this prediction task from labeled examples in an automated fashion. In many applications, the distribution underlying the instances or the rules underlying their labeling may change over time, i.e. the goal of the prediction, the class to be predicted or the target value to be predicted, may change over time.

Many studies have been done to support data stream mining especially for concept drift. (Salganicoff 1997) uses an incremental method to classify data and constructs an incremental tree whenever concept drift happened and tries to build an alternate sub tree and replace with old one. In this algorithm, concept drift is detected by error measuring of classifier. Although it is suitable for data stream evolving but it causes a main problem such as: what should be done when concept drift happen on top of the tree where update propagating is very time consuming from top to down. This disadvantage can cause overhead to process data. (Haixun et al. 2003) employed ensemble classifier method that each classifier is weighted based on accuracy of test data classification. Whenever accuracy decreases to some threshold value then another classifier is selected for classification. It introduced a dynamic classification for concept drift and applied on both real and synthetic data sets. The experiments show that in both data sets, it works much better than incremental approach. However, changing concepts during the time is not certainly random. Many events repeat during

the time and it is possible to predict behaviour of stream whereby this situation has not been considered in their study. Another problem is distributions of data. The most studies assume distributions of data relatively balanced and stable data streams. (Jing et al. 2008) designed an effective framework based on sampling and ensemble techniques. The algorithm generates a balanced training set by keeping all positive examples and under sampling negative examples. The training set is further divided into several samples and multiple models are trained on these samples. The final outputs are the averaged probability estimates on test data by multiple models. The error reduction is significant according to the experimental results. In addition, various kinds of concept change should be studied under this framework since it has been applied on two class problems only.

Most of the data stream classification algorithms are interested in maintaining a decision model consistent with the current status of the nature. This leads to the sliding window models whereby data is continuously inserted and deleted from a window. Thus, learning algorithms must be well designed for incremental learning and forgetting (Gama et al. 2009). As mentioned, concept drift and novelty detection have been well studied but there are still open issues in this area.

2.4.3 Clustering Data Stream Mining

Cluster analysis, primitive exploration with little or no prior knowledge, consists of research developed across a wide variety of communities. The diversity, on one hand, equips us with many tools. On the other hand, the profusion of options causes confusion. Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. Clustering is a main task of explorative data mining and a common technique for statistical data analysis It was used in many fields including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Clustering data stream is particularly important in Web click stream mining, network traffic monitoring and wireless sensor network applications.

Some general terms in clustering data stream that will be used in this study are described in Table 2.3.

Table 2.3 General terms in data stream clustering

Terminology	Description
Sample (data object)	Set of attributes which is represented by a multidimensional vector
Feature	A dimension of vector which can be quantitative or qualitative, continuous or binary, nominal or ordinal
Distance function	Symmetry: $D(x_i,x_j) = D(x_j,x_i)$ and positivity: $D(x_i,x_j) >= 0$
Metric	Symmetry, positivity, triangle inequality: $D(x_i,x_j) \leq D(x_i,x_k) + D(x_k,x_j)$ and reflexivity: $D(x_i,x_j) = 0$ if $x_i = x_j$
Proximity matrix	Similarity or dissimilarity measure for the *i*th, *j*th patterns
Outlier	Data object that does not comply with the general behavior of the data
Noise	Noise is random error or variance in a measured variable
Concept drift	Clusters changing in term of content during the time
Novelty detection	New clusters appearing during the time

When developing clustering methods for stream data based on data stream characteristics, new issues arise that require to be taken into account. Traditional clustering algorithms are not suitable for data stream processing because they are developed to work on static data. Hence, they are cannot be employed directly to mine classes in stream data.

Generally two main categories of algorithms have been developed for clustering, known as partitioned and hierarchical methods. Partitioned clustering directly divides samples S into some predefined number of clusters $C=\{c_1,\ldots, c_k\}$ such as:

(1) $c_i \neq \emptyset$; $where\ i = 1, \dots, k$

(2) $\bigcup_{i=1}^{k} c_i = S;\ where\ i = 1, \dots, k$

(3) $c_i \cap c_j = \emptyset; where\ i, j = 1, \dots, k\ and\ i \neq j$

In contrast to partitioning clustering, hierarchical clustering groups data S in a nested tree structure of partitions M={m_1,…, m_p} (m <= n), as $c_i \in m_h, c_j \in m_l\ imply\ that\ c_i \subset c_j\ or\ c_i \cap c_j \neq \emptyset\ \ where\ \forall i, i \neq j,\quad h, l = 1 \dots p$.

There are two main approaches for hierarchical clustering, top down that is called divisive and bottom up that is agglomerative hierarchical clustering (Han & Kamber 2006). Although the quality in these methods is very high but computational time is one of the most important criticisms. This weakness caused that they would not be suitable for large and high dimensional datasets. Furthermore reallocating samples is not feasible when tree structure is constructed. These methods are also sensitive to outliers and noises data. Regardless of hierarchical clustering which yields a successive level of clusters, partitioned clustering assigns a set of samples into k clusters without any nested structure. This process uses a criterion function and attempts to optimize it. Many studies have been done and proposed methods for partitional clustering (Selim & Ismail 1984; Pelleg & Moore 1999; Stoffel & Belkoniene 1999; Bagirov & Mardaneh 2006; Wilkin & Huang 2007; Li et al. 2008; Wang, Ye & Huang 2008; Jain 2009; Khalilian et al. 2009; Mahdavi & Abolhassani 2009; Patra, Nandi & Viswanath 2011). Based on divide and conquer concept (Figure 2.3) we are capable to achieve nested structure of hierarchical clustering and primitive knowledge has been leveraged in dividing data whereas dividing whole data without any criteria or prior knowledge (Guha et al. 2003; Cormode, Muthukrishnan & Wei 2007). Most partitional clustering algorithm such as K-Means has the shortcomings of depending on the initial state and converging to local minima (Zhang, Ouyang & Ning 2010). Therefore, heuristic clustering algorithms are the alternate methods for clustering data. Although, heuristic algorithm e.g. genetic algorithm (Chang, Zhang & Zheng 2009), simulated annealing, tabu search, bee colony (Zhang, Ouyang & Ning 2010) and harmony search (Mahdavi & Abolhassani 2009) attempt to overcome the weaknesses of partitional clustering algorithms but they suffer from the huge number

of repetition to be convergent to the optimum solution (Abul Hasan & Ramakrishnan 2011). Thus, heuristic search algorithms are not suitable for high scaled datasets. Since data stream clustering is the main focus in this study, the details discussion will be provided in the next chapter on data stream clustering methods.

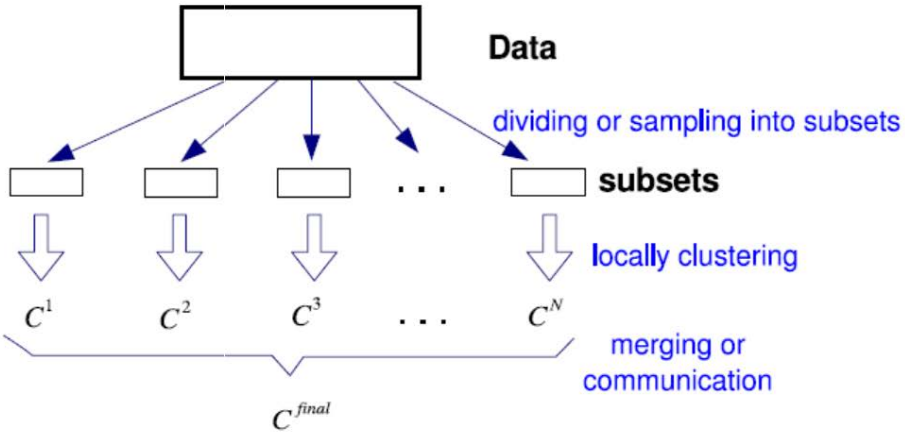


Figure 2.3 The framework of divide and conquer strategy

2.5 DATA STREAM CLUSTERING PROCESS

There are four basic steps in data stream clustering process which is depicted in Figure 2.4 which has been described by (Rui & Wunsch 2005). These steps are closely related to each other and affect the final results.

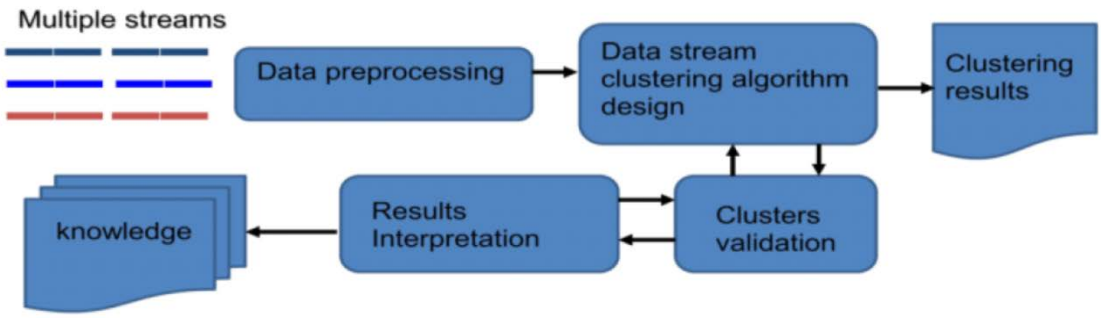


Figure 2.4 Data stream clustering process

Data preprocessing

As mentioned in data preprocessing for data stream mining in Section 2.3, it includes data cleaning, transformation and data reduction. One of the most important issues for these steps is feature selection as it affects the final result directly. Generally, ideal features must be used in distinguishing patterns of different clusters, easy to extract and immune to noise.

Data stream clustering design

There is no universal clustering algorithm that can be used to solve all problems. Thus, it is important to specify the characteristics of application in order to design an appropriate clustering strategy. In the next chapter we will discuss the different strategies and different problems in data stream clustering.

Cluster validation

Generally, there are three categories of testing criteria: external indices, internal indices, and relative indices. External indices are based on some pre-specified structure, which is the reflection of prior information on the data and used as a standard to validate the clustering solutions. Internal tests are not dependent on external information (prior knowledge). On the contrary, they examine the clustering structure directly from the original data. Relative criteria places the emphasis on the comparison of different clustering structures in order to provide a reference to decide which one may best reveal the characteristics of the objects.

Results interpretation

The ultimate goal of clustering is to provide users with meaningful insights from the original data, so that they can effectively solve the problems encountered. Experts in the relevant fields have to interpret the data partition. Thus, further analyzes and experiments may be required to guarantee the reliability of extracted knowledge.

2.6 DATA STREAM CLUSTERING METHODS

There are various categories of methods for clustering data stream from various perspectives:

i. First group refers to type of solution includes algorithm-oriented and data-oriented which was described in Section 2.2.

ii. Second group of methods, attempt to develop novel algorithms with regard to computational complexity:

- Condensation-based: BIRCH is the first method for high scale datasets which introduces additive and subtractive property (Tian, Raghu & Miron 1996). Furthermore, it generates and stores the compact summaries of original data in a tree structure. Therefore it is capable to capture the clustering information efficiently and largely reduce computational complexity. There are lots of diversity methods which have employed the basic concept in BIRCH and generalized it into the broader framework (Aggarwal & Yu 2008; Aggarwal 2009c).

- Data sampling: the key point lies that the appropriate size of samples has an important role which affects on quality of the final results. On the other hand, memory and time constraints depend on this sample size, for instance in STREAM (Guha et al. 2003) draw a sample of size $s = \sqrt{nk}$ where n is number of objects and k is the number of clusters.

- Density-based: if an object belongs to a cluster, it requires that the density in a neighborhood for that object must be high enough. The neighborhood needs to satisfy density threshold which has been specified by the user. The most prominent advantage for this group of approaches is detecting arbitrary shaped clusters. On contrary, most density based algorithms require relearning due to keeping only the most recent data points in memory and likely to discard possibly reusable cluster information. Another important point is sensitivity to noise around specific object.

- Grid-based methods: there are two main approaches for this group of methods including, wave and fractal clustering (Rui & Wunsch 2005). Most efforts in data stream clustering have been carried out by fractal clustering which combines the concept of both incremental clustering and fractal dimension. Data objects are assigned to the clusters incrementally and represented as cells in a grid considering stability of fractal dimension.

- Hierarchical structure methods: obviously, classical hierarchical clustering algorithms, including single linkage, complete linkage, average linkage, median linkage are not suitable for using in data stream clustering area because of quadratic computational complexity in both time and memory usage. However, there are some modified hierarchical clustering algorithms which have been utilized for data stream. For instance (Rodrigues, Gama & Pedroso 2007) have presented ODAC, a clustering system for streaming time series. ODAC uses a top-down strategy to construct a binary tree hierarchy of clusters with the goal of finding highly correlated sets of variables. A common measure of cluster quality is the cluster's diameter, which is defined as the highest dissimilarity between objects of the same cluster. The system evolves by continuously monitoring the diameter of the clusters. Table 2.4 shows the pros and cons of general group of methods for data stream clustering.

Table 2.4 General group of methods for data stream clustering

Method	Advantages	Disadvantages
Condensation-based (Aggarwal & Yu 2009b) (Aggarwal & Yu 2008)	Having summary of data (global view) Linear complexity Scalability Additive & subtractive property	Resource constraints Detecting only spherical shape Relearning Applicable in low dimension
Data sampling (Guha et al. 2003; Heinz & Seeger 2008)	Speed up Memory usage Low computational complexity	Low quality
Density-based (Chen & Tu 2007; Jin-yin & Hui-hao 2015; Chen & He 2016; Ding et al. 2016)	Arbitrary shaped clusters	Density threshold must be determined Noise sensitivity Outlier sensitivity Applicable in low dimension Relearning
Grid-based (Guopin & Leisong 2008; Lin & Chen 2008; Tu & Chen 2009; Pardeshi et al. 2011)	Arbitrary shaped clusters High dimension (Zhizhong, Ruichun & Yinzhen 2011)	Stability Relearning
Hierarchical structure (Rodrigues, Gama & Pedroso 2007; Chen & Liu 2009; Wei & Brice 2009; Pardeshi et al. 2011)	Support evolving and concept drift No need to determining extra parameters	Relearning Inflexibility

iii. Third group of methods that look into data processing perspective for each stream can be carried out in two fashions: batch and incremental online. In batch processing, each stream of data processed in the memory and produce results. Despite of batch processing, samples arrive one by one and they assign to the nearest cluster. Table 2.5 shows the differences between batch and incremental online processing.

Table 2.5 Batch processing vs. incremental online processing

Batch processing	Incremental online processing
Outlier is easy to detect	Distinguishing between new cluster and outlier is problematic
Managing concept drift can be done by frequent split and merge	Managing concept drift requires a method such as fading function and decay concept which cause some challenges e.g. time for triggering concept drift
Speed up is low in each window	Speed up is high in each window
More accurate specifically in clusters boundary	Accuracy depends on many parameters which should be determined when a sample must be assigned to a cluster
Few parameters to set	Many parameters to set
Free re-learning	Need to re-learn except using a repository to keep history of previous clusters

iv. The last group of approaches refers to solutions and techniques which are component-based and non-component-based methods from framework structure paradigm. We review these methods with their advantages and disadvantages in the next subsections.

2.6.1 Component-Based Methods

Generally, these methods are based on two main components: online and offline components; consequently, stream clustering take places in two steps online and offline. This kind of framework was proposed for the first time by (Aggarwal et al. 2003) and in continue they have applied this framework to solve other problems in data stream clustering (Aggarwal & Yu 2006; Aggarwal 2009a; Aggarwal 2009c). As mentioned before, main problems in data stream clustering are visiting data once and concept drift. Thus, online component is dedicated to generate micro-clusters and

offline produces macro-clusters. Generally, approaches which are applied k-Means or k-Medians suffer from lack of accuracy when there are a lot of outliers. In addition, these methods are not suitable for discovering clusters with non-convex shapes or clusters of very different size because they calculate clusters based on means or medians. Moreover, number of clusters should be determined as value of parameter k; thus, they increased the number of parameters which should be controlled by user. K-means and K-medians are also sensitive to initial values as depicted in Figure 2.5.

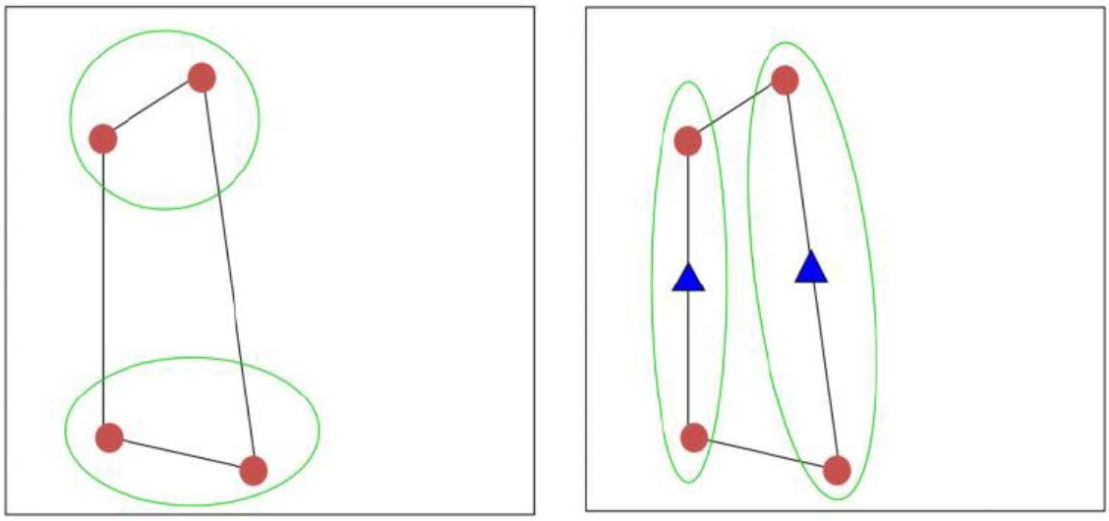


Figure 2.5 Different initial values for k-means yields different outcomes

Aforementioned weaknesses motivated researchers to employ some other techniques, e.g. (Lühr & Lazarescu 2009) have developed a connectivity based representative points to cluster data stream. They include the items below in online component (1-2) and offline component (3-4):

1. Adding arrival point to the sparse graph.
2. If it is reciprocally connected to representative vertex then joins existing clusters otherwise it is considered as exemplar or predictor.
3. Using AVL tree structure for search representative vertices.
4. Using usefulness parameter to update of repository based on decay concept.

As the result, accuracy is outstanding in their research but it exhibits low speed. Therefore, the main problems of their finding are as follows:

i. Managing lots of pointers in memory.

ii. Violent memory constraint condition.

iii. Complexity in its programs.

iv. Lots of parameters must be determined.

v. Using fixed value for decay in fading function for offline component. It employs a priority FIFO queue to manage evolving data (may be a cluster archived whereas in near future some new data points arrive).

(Yeh, Dai & Chen 2007) presented an enhanced method for online monitoring clusters over multiple evolving streaming data by correlations and events. The data streams are smoothed by piecewise linear approximation, and each end point of the line segment can be regarded as a trigger point. At each trigger point, for clusters that have trigger streams, they update the weighted correlations related to trigger streams in clusters. While an event occurs, the clusters are improved via effective split and merge procedures. An enhanced entropy-based technique has been proposed for mixed numeric and categorical data streams clustering (Wang et al. 2008). They also utilize online and offline components to process data.

(Rodrigues, Gama & Pedroso 2007) have proposed a clustering method for streaming time series and used a top-down strategy to make a binary tree hierarchy of clusters, with the aim of finding highly correlated sets of variables. An usual measure of cluster quality is the cluster's diameter, which is determined as the highest dissimilarity among points of the same cluster. The system evolves by continuously monitoring the diameter of the clusters. The samples are processed as they arrive by utilizing a single scan over the whole data. The system incrementally calculates the dissimilarities between time series, maintaining and updating the adequate statistics at each new sample arrival, updating only the leaves. The splitting criterion is supported by a confidence level given by the Hoeffding bound, which is detected when the system has gathered enough information to confidently define the diameter of each

individual cluster. The system includes an agglomerative phase, based on the diameters of existing clusters, also supported by the Hoeffding bound. The aggregation phase enables the adaptation of the cluster structure to smooth changes in the correlation structure of time series.

ConStream is one of the famous works (Aggarwal 2009a; Aggarwal & Yu 2009b) which propose a condensation based approach for stream clustering. It summarizes the stream into a number of fine grained cluster droplets. These summarized droplets can be used in conjunction with a variety of user queries to construct the clusters for different input parameters. Thus, this provides an online analytical processing approach to stream clustering. ConStream also provides a method to detect noisy and outlier records in real time. Due to use incremental online processing, ConStream distinguishes between the different kinds of clusters and abnormalities by trend-setter and mature cluster. When a cluster is newly discovered during the arrival of the data stream, it is referred to as a trend-setter. From the point of view of the user a trend-setter is an outlier, until trend-setter absorbs other points. Thus, it can certify the fact that it is actually a cluster. If and when a sufficient number of new points have arrived in the cluster, it is referred to as a mature cluster. At a given moment in time, a mature cluster can either be active or inactive. A mature cluster is said to be active when it has continued to receive data points in the recent past. When a mature cluster is not active, it is said to be inactive. In some cases, a trend-setting cluster becomes inactive before it has had a chance to mature. The main difficulty for this method is threshold value specification. If it specifies threshold too small, many clusters do not have a chance to become mature. On the other hand, if it specifies threshold too large, all trend- setters become mature.

Many parameters should be determined in this method. In addition, in order to generate clusters, it uses incremental online processing with sophisticated decay concept. Thus, it causes decreasing quality of the clusters.

2.6.2 Non-Component-Based Methods

BIRCH (Zhang, Ramakrishnan & Livny 1996) clustering at it initial stage was developed to extract traditional data. The approach was later found capable of handling massive date thus it is also used to mine data streams, hence the notions of micro and macro-clustering is growing. These two concepts enable BIRCH to overcome two major drawbacks found in the HAC algorithm, namely, scalability and failure to undo what has been previously executed. This method consists of two phases: in the first phase, it scans the data base and then forms a tree including information regarding to data clusters. In the second phase BIRCH prunes the tree by discarding sparse nodes (outliers) and creating new original clusters. However, this technique has a major disadvantage in the form of the limited capacity of its leaves. Moreover, this algorithm will not execute well if the clusters do not have spherical shapes because BIRCH controls the cluster's boundary by applying the notion of radius/diameter (Khalilian et al. 2013).

STREAM is the next main method which has been designed especially for data stream clustering (Guha et al. 2002). In this method K-Medians is leveraged to cluster objects with SSQ (Sun of Square) criterion for error measuring. To understand STREAM, the first step is to show that clustering can take place in small space (not caring about the number of passes). Small-Space is a divide-and-conquer algorithm that divides the data, S, into pieces, clusters each one of them and then clusters the centers obtained. In the first scan, objects grouped and medians of each group is gathered and associated them a weight with regard to the number of objects in the cluster. In second step these medians is clustered until top of the tree. The problem with the Small-Space is that the number of subsets that we partition S into is limited, since it has to store in memory the intermediate medians in X'. So, if M is the size of memory, we need to partition S into subsets such that each subset fits in memory, (n/M) and so that the weighted k centers also fit in memory, k<M. The STREAM algorithm as shown in Figure 2.6 solves the problem of storing intermediate medians and achieves better running time and space requirements. The algorithm works as follows:

1. Input the first m points; using the randomized algorithm presented in (Guha et al. 2003) reduce these to O(k) (say 2k) points.

2. Repeat the above till we have seen m2/(2k) of the original data points. We now have m intermediate medians.

3. Using a local search algorithm, cluster these m first-level medians into 2k second-level medians and proceed.

4. In general, maintain at most m level-i medians, and, on seeing m, generate 2k level-i+ 1 median, with the weight of a new median as the sum of the weights of the intermediate medians assigned to it.

5. When we have seen all the original data points, we cluster all the intermediate medians into k final medians, using the primal dual algorithm.

Although it overcomes data stream clustering in terms of memory constraint and visiting data once by employing divide and conquer method but we can realize two main disadvantages for this method: time granularity and data evolving.

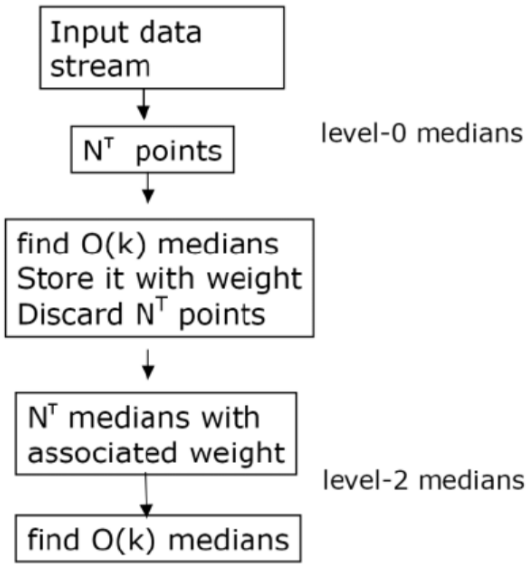


Figure 2.6 STREAM algorithm