Cogn Comput (2016) 8:30–38
DOI 10.1007/s12559-015-9342-z

An Adaptive Density Data Stream Clustering Algorithm

Shifei Ding[1,2] · Jian Zhang[1,2] · Hongjie Jia[1,2] · Jun Qian[1]

Received: 26 March 2015 / Accepted: 16 June 2015 / Published online: 3 July 2015

Abstract Now we are in the age of big data. Huge amount of data and information are generated every time. Traditional data stream algorithms are suit for the data streams with low dimension and simple structure. However, with the development of information technology, the produced data streams are becoming more and more complicated. It is particularly important to study how to find new associations and patterns from complex data to achieve the cognition ability and judgment ability like human brain. Clustering data streams with mixed attributes of irregular distribution is a big challenge in data mining. To solve this problem, we present an adaptive density data stream clustering algorithm—ADStream. ADStream is based on the online–off-line clustering framework. It can automatically recognize the initial clusters by passing messages between data points. Then a novel time-decay density clustering strategy is designed to group and update the continuously arriving data streams. Comprehensive experimental results demonstrate that ADStream is adaptive to the evolving data streams and may generate high-quality clusters with fast processing rate.

Keywords Data stream · Clustering · Adaptive density

✉ Shifei Ding
dingsf@cumt.edu.cn

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Introduction

The traditional computing technology is focusing on quantitative and deterministic problems, but it is not good at solving the imprecise and uncertain problems in biological systems [1]. Different from the traditional computing, cognitive computing is a new data-centric computing model. It hopes to give machine the ability of self-learning to achieve more natural human–computer interaction. Among all kinds of data information, data stream is an important data type in our daily life. Data stream has its own characteristics compared with traditional data concept. Traditional data is static, stably stored in the database, and can be processed many times. Data stream, however, is rather different. In data stream, the data is consecutive, ever-changing in a flow way [2, 3]. Real-time, continuous, ordered sequences are common words used to describe the data stream. Besides, large amount, uncertain arrival rates are also obvious features of the data stream.

The key to cognitive computing is the data. Through the analysis and processing on large amounts of data, the machine will become more intelligent [4]. A representative of the cognitive computing system is "Watson" supercomputer developed by IBM, which beat humans at "Jeopardy" TV show and won the game in 2011. In many practical applications, the distribution of data varies over time. For example, in news, blogs, BBS and other online media, most topics that people discussed change dynamically. Even for the same topic, the content is not exactly the same as one year ago, such as "fashion" and "high tech." This is known as concept drift in the Internet data analysis. Such kind of problem brings some profound changes and challenges for machine learning. Traditional clustering algorithms have been unable to meet the clustering

Table 1 Comparison of data stream clustering algorithms

Algorithm	Core technology	Clustering shape	Noise analysis	Advantage	Disadvantage
Stream [13]	Divide-and-conquer strategy	Hyper sphere	No	Uses LocalSearch method to find more reasonable cluster number "k"	Clustering on entire data streams, ignore their dynamic changes over time
CluStream [14]	Online micro-clustering, offline macro-clustering	Spherical	No	Using Pyramidal time frame can generate clustering results on different time granularities	Take no account of the attenuation of history data or the importance of recent data
HPStream [15]	Dimension projection principle	Spherical	No	Introduce projection technology to process high-dimensional data and attenuation cluster structure to preserve historical data	Not suitable for handling irregularly distributed data streams, and interfered by noise easily
DenStream [16]	Density method	Arbitrary shape	Yes	Can handle the clustering problem of arbitrary shaped data streams and detect outliers	The clustering process will slow down with the increase of data amount
D-Stream [17]	Density grid structure	Arbitrary shape	Yes	Map data elements into their corresponding grid, clustering based on the grid density	Perform not well when dealing with high-dimensional data streams
ACluStream [18]	Space segmentation	Arbitrary shape	Yes	Can keep the spatial characteristics of data streams in the preliminary clustering	Cannot properly deal with the clustering problem of data streams in non-Euclidean space
P-Stream [19]	Adaptive expiration processing	Arbitrary shape	Yes	Capture the cluster probability which changes timely. Find more clusters with larger existing probability	May produce abnormal points when the clustering model does not suit new arrived probability tuples

requirements of dynamic data streams [5, 6]. On the one hand, in terms of fitting or predicting future data, we cannot use the learning machine which is trained by historical data to test future data directly like traditional learning problems, as the independent identical distribution hypothesis is not true; on the other hand, from the view of modeling, the probability of the sample set cannot be simply written as the product of each sample's probability, for lacking of independent and identical distribution [7]. In order to cluster the data from data streams, we need to modify and improve traditional theory or method, even propose new clustering algorithms.

After years of research, the analysis of data stream clustering has made great progress. Many low-dimensional and small complex data stream problems have been deeply studied [8, 9]. In 2-D computer-assisted animation production, Yu Jun et al. present a semi-supervised patch alignment framework, which introduces pair-wise constraints to improve the performance of correspondence construction [10, 11]. Manifold learning based on graph is a promising method in extracting features from images. As the density of data points' distribution may be different in different regions, Yu et al. [12] develop a novel sparse patch alignment strategy for the embedding of data lying in multiple manifolds. Table 1 shows the comparison of existing data stream clustering algorithms.

However, there are still many difficulties to be resolved in irregular mixed high-dimensional data streams [20]. Such complex hybrid data streams bring new requirements for data stream clustering: First, as hybrid data streams contain a lot of noise data, how to filter out the noise and identify the correct data becomes more difficult [21]; second, the distribution of data streams is irregular, which requires more refined methods to describe the data structure [22]. In addition, the number of generated clusters is unknown in data streams. The cluster number will change along with the ever-changing data flows, which also increases the clustering complexity of uncertain data streams [23, 24].

Faced with these challenges, this paper mainly focuses on the analysis of irregular data streams and proposes an adaptive density data stream clustering algorithm—ADStream algorithm. ADStream combines the advantages of density clustering [25] and affinity propagation clustering [26]. In this algorithm, the initial cluster centers are determined by an improved affinity propagation method, avoiding the negative influence of random initialization. Besides, ADStream is adaptive to the density characteristics of data streams and can well deal with the irregular data such as noise or outliers.

The rest of this paper is organized as follows. Section 2 introduces the related concepts of ADStream algorithm. Section 3 presents the detailed framework of adaptive

density data stream clustering algorithm. Section 4 provides experiments of data stream clustering on MOA platform and real-world machine learning data sets. Conclusions are drawn in Sect. 5.

Related Concepts of ADStream Algorithm

In many practical applications, the state of data streams is time-varying, so the cluster number is hard to predict at each time point. The knowledge contained in recent tuples is often more valuable than that in history tuples. Therefore, in this research, an attenuation window model is used to record different data streams as time goes by. Table 2 lists some important notations used in this paper.

Definition 1 Data weight: In the sliding window model, data points in data stream change over time, and the fading function is $f(t) = 1/2^{\lambda t} < \varepsilon\ (\lambda > 0)$. The larger the value of λ is, the lower the importance of historical data is. Assuming the arrival rate of data streams is v, namely the number of data points that captured by window per unit time, then the data weight of data stream can be expressed as

$$\omega = v \cdot \sum_{t=0}^{t=t_c} 1/2^{\lambda t} = \frac{v}{1-2^{-\lambda t}}\ (t_c \text{ represents the current time}) \quad (1)$$

Definition 2 Dimension radius: the jth dimension of data point x_i is x_{ij}, the dimension radius upon the jth dimension is r_j, then the division on the jth dimension of point x_i is $(x_{ij}-r_j, x_{ij}+r_j)$. $R = (r_1, r_2, \cdots, r_d)$ represents the dimension radius vector composed of d dimension radiuses.

Definition 3 Density units and aggregation blocks: In high-dimensional data streams, define a unit length for each dimension and divide the d-dimensional space into density units $Den(\mathrm{o})$. A density unit is a hyperspace started from a d-dimensional vector o_i extending a unit length along the positive direction in all dimensions. The hyperspace is called aggregation block denoted by $Cub(o_i, \vec{r})$, where $\vec{r}$ is the unit vector on each dimension.

Affinity propagation (AP) algorithm is a novel clustering algorithm [27]. It can adaptively find and update the central exemplars of data streams by transmitting information between data points. And the cluster number will be detected automatically without user specification. AP algorithm has two important parameters: damping factor λ and preference parameter p. Selecting an appropriate value for damping factor is important to the clustering quality. In this section, an improved damping factor for AP algorithm is applied to the online process of data stream clustering.

Definition 4 Shrinkage factor and similarity measure: in order to accelerate the convergence of AP algorithm, we introduce the shrinkage factor ρ to update the messages passed between data points during the clustering [28]:

$$\rho = \frac{2}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|},\ \phi > 4 \quad (2)$$

There are two kinds of passing messages: "responsibility" and "availability." Their updating formulas are as follows:

$$r^{(t)}(i,j) = (1-\lambda)\left(S(i,j) - \max\left\{a^{(t-1)}(i,k) + S(i,k)\right\}\right) + \rho \times \lambda \times r^{(t-1)}(i,j) \quad (3)$$

$$a^{(t)}(i,j) = (1-\lambda)\left\{\min\left\{0, r^{(t-1)}(j,j) + \sum \max\left\{0, r^{(t-1)}(k,j)\right\}\right\}\right\} + \rho \times \lambda \times r^{(t-1)}(i,j) \quad (4)$$

where $S = r^{(t)}(i,j) + a^{(t)}(i,j)$ is the pair-wise similarity of data points. For data $x = (x_1, x_2, \ldots, x_d)$ and data $y = (y_1, y_2, \ldots, y_d)$, $\omega(x_i, y_i)$ is the attribute weight between x and y.

As data streams are always changing, the initial clusters produced by AP algorithm also require constant maintenance and updating. Here a density-based clustering algorithm is used to merge or delete these generated clusters to capture the uneven density distribution of data streams.

Table 2 Important notations used in this paper

Notations	Descriptions
$\omega = \frac{v}{1-2^{-\lambda t}}$	Data weight of data stream
$R = (r_1, r_2, \ldots, r_d)$	The dimension radius vector composed of d dimension radiuses
$Den(\mathrm{o})$	Density unit
$Cub(o_i, \vec{r})$	Aggregation block
ρ	The shrinkage factor of affinity propagation method
$S = r^{(t)}(i,j) + a^{(t)}(i,j)$	The similarity between data points
(c_i, k_i)	Reference point of density cluster

Definition 5 Reference point of density cluster: Use the n representative points of clusters calculated by AP algorithm for density-based clustering and generate m new clusters: $(c_1, c_2, \ldots, c_m)$. Then data will be recorded by m two-tuples structure (c_i, k_i), where k_i is the number of data points attached to c_i in cluster. (c_i, k_i) is called the reference point of density cluster.

Definition 6 Density micro-cluster: Assume the data stream objects $x_{i1}, x_{i2}, \ldots, x_{\text{in}}$ arrive at time $t_{i1}, t_{i2}, \ldots, t_{\text{in}}$. This data is included in the density unit with o_i as the starting point. At time t_i, the data structure of the density unit's characteristic is denoted as: $\left(o_i, H, \overline{CF^1}, \overline{CF^2}, S, t_i\right)$, where $\overline{CF^1} = \sum_{j=1}^{i} x_{ij} 2^{-\lambda\left(t_i - t_{ij}\right)}$, $\overline{CF^2} = \sum_{j=1}^{i} x^2{}_{ij} 2^{-\lambda\left(t_i - t_{ij}\right)}$, $S = r^{(t)}(i,j) + a^{(t)}(i,j)$. H is the frequency of corresponding attributes of data object. When data weight $\omega > \xi$ and data similarity $S > \varepsilon$, the density units become density grid micro-clusters; when $0 \leqslant \omega \leqslant \xi$ and $0 \leqslant S \leqslant \varepsilon$, the density unit is called candidate density micro-cluster.

To simplify the algorithm's calculation complexity, the unit length of each dimension is determined according to the value of parameter ε in the literature [10]. ε is a threshold and $\varepsilon \in [0, 1]$. We also introduce this method to our algorithm. If the data similarity in each dimension is not less than ε, the data similarity on global attributes satisfies the same condition.

Adaptive Density Data Stream Clustering Algorithm

In order to deal with irregular complex data streams, we improve the density-based DenStream algorithm and propose an Adaptive Density data Stream clustering algorithm (ADStream). ADStream algorithm introduces two main concepts: "density micro-cluster" and "time frame" structure. It divides the data stream clustering process into the online part (micro-clustering) and off-line part (macro-clustering). The online part handles the newly arrived data in real time and stores these statistical results periodically; the off-line part uses these statistical results, combined with user-entered parameters, to approximately calculate the clustering results of a certain time in the past.

Online Clustering Stage

For the new arrival data objects of data streams, assign them to the corresponding grid according to their attributes. In each grid, update the similarity of data and weight of attribute in the density unit by AP algorithm. According to the similarity matrix and calculated weights, use density clustering algorithm to determine the density micro-clusters and candidate density micro-clusters. Because data stream is potentially unlimited, with the real-time updating of data, the method to classify the most recent data objects is essential. When the data point is arrived, we should determine whether to incorporate the data into the density micro-clusters or a density unit according to their attribute values. For the density micro-clusters that added new data, its eigenvector $\left(o_i, H + h, \overline{CF^1} + x, \overline{CF^2} + x_2, S + 1, t_i\right)$ should be updated. For the density unit with new data, its eigenvector also needs to be updated. Meanwhile, the algorithm will determine whether it meets the conditions of candidate density micro-cluster or candidate density micro-cluster, then will make appropriate changes.

When new data continuously arrives from data stream, the constructed density units, density micro-clusters and candidate density micro-clusters will become more, and they will need greater storage space to be recorded. However, most of the data which is received long time ago is useless because the information they carry is descending with time index. We can delete or fuzzy record this data. There are two principles for the reduction in data stream: (1) direct degenerate as a candidate density micro-cluster. If the feature weights of density micro-cluster are smaller than the threshold, then it will directly degrade as a candidate micro-cluster; (2) modify the eigenvectors of density micro-clusters. If there is no data update in density units or density micro-cluster, modify its eigenvector as $\left(o_i, H, \overline{CF^1} \times 2^{-\lambda\Delta t}, \overline{CF^2} \times 2^{-\lambda\Delta t}, S \times 2^{-\lambda\Delta t}, t_i\right)$ (Δt means the time interval of modifying eigenvectors. In order to reduce the consumption of resources, generally let $\Delta t = 1/\lambda \log(\xi/\xi - 1)$). If there are no data updates in density micro-cluster for a very long time, the eigenvector will not stop attenuation, and feature weights will be lower than the threshold value, and thus the density micro-cluster will be gradually attenuated as a candidate density micro-cluster.

To save memory space and store the recent arrival information, the algorithm gives the method of deleting density unit: For a large amount of density units, calculate the similarity S of each density unit and weights ω, record the density unit that meets the conditions of similarity and weight as density micro-cluster, record the density unit that meets the conditions of weight ω as candidate density micro-cluster, and delete the density unit that does not meet the conditions. For the candidate density micro-clusters, if there is no new data coming into the candidate density micro-clusters for a period of time, then sort these candidate density micro-clusters according to the weight of eigenvectors and delete the candidate micro-clusters with

weights w from small to large to meet the memory requirements.

Off-line Clustering Stage

At off-line clustering stage, we can get the connected density units by density algorithm. The interconnected density units can be merged into a clustering unit, and its combined attributes can be represented by geometric distance of these two density units [34-36]. After merging two density units, only if their attributes still meet the threshold conditions, they are regarded as connected units, and the combination between them can be founded. In the algorithm, the connectivity here has transitivity, namely: For the density unit d_i and d_j, if d_j and d_k are connected, respectively, then d_i and d_k are connected.

Concrete Steps of ADStream Algorithm

Algorithm name: ADStream data stream clustering algorithm

Algorithm input: Data Flow D, Parameter λ, ξ, ε

Algorithm process:

(1). procedure ADStream(D, λ, ξ, ε)
(2). time ==0//Initial time
(3). while($D \neq NULL$)
(4). Read the initial data of a period of time D_1, to construct the initial data similarity matrix;
(5). Use AP algorithm for the initial clustering of initial data and initialize a cache area;
(6). Calculate the density and eigenvector of each cluster built at initial clustering;
(7). Reading each dimension attributes of new data $x = (x_1, x_2, \ldots, x_d)$ and use AP algorithm to calculate the similarity between data, put the data into each cluster of step (5); if they do not meet the conditions of incorporating into the existing clusters, then put them into a temporary buffer.
(8). Modify and update the eigenvector of each cluster;
(9). if (the number of density units reaches the limit)
(10). Calculate similarity S and weights ω of all density units;
(11). if ($S > \varepsilon \&\& \omega > \xi$)
(12). The density unit is a density of micro-cluster;
(13). else if($\omega > \xi$)
(14). The density unit is a candidate density micro-cluster;
(15). else
(16). Remove density unit and reclaim memory;
(17). end if
(18). if (($t_i \bmod \Delta t$) == 0)//Check the attenuation of density unit
(19). if ($\omega > \xi$),
(20). Remove density unit and reclaim memory;
(21). end if
(22). end if
(23). for ($i = 0$; i < number of density units; i ++)
(24). for ($j = 0$; $j <$ number of density units; j ++)
(25). if (i and j are adjacent)
(26). Calculate the similarity S weights ω of merged density units;
(27). if($S > \varepsilon \&\& \omega > \xi$)
(28). Merge density unit i and density unit j;
(29). end if
(30). end if
(31). end for
(32). end for
(33). Output the clustering results
(34). end procedure

Convergence and Computational Complexity Analysis of ADStream Algorithm

Although in data streams new data arrives every time, the total number of data sub-blocks waiting to be clustered is finite. We only need to show that each clustering on each subset converges, and then it can be inferred that algorithm in the whole clustering process will converge [29]. In the online clustering stage of ADStream, AP algorithm is used to calculate the similarities of new arrival data objects and the weights of attributes. Then the similarity matrix and weights can help judge density micro-clusters and the candidate density micro-clusters. In each iteration t of AP algorithm, responsibility $r^{(t)}$ and availability $a^{(t)}$ will be weighted and updated with the last iteration of $r^{(t-1)}$ and $a^{(t-1)}$: $r^{(t)} = (1-\lambda)r^{(t)} + \lambda r^{(t-1)}$, $a^{(t)} = (1-\lambda)a^{(t)} + \lambda a^{(t-1)}$ (where $\lambda \in [0, 1]$, the default value is 0.5), which reflects the effect of damping factor λ. Another role of λ is to improve the convergence: When AP algorithm has numerical oscillations (the number of produced clusters swings during the iterative process) and cannot converge in some circumstances, increasing λ can eliminate the oscillation [26].

When traditional AP algorithm falls in an oscillation, it needs to increase λ manually and rerun the program until the algorithm converges. In order to avoid oscillations, another approach is directly setting λ close to 1, but the update of responsibility $r^{(t)}$ and availability $a^{(t)}$ will become very slow, which increases the iteration number and running time of the algorithm. Maurice Clerc's research shows that using constriction factor can effectively ensure the algorithm convergence [30]. Therefore, we introduce the constriction factor (shown as formula 2) into the updating formula of responsibility and availability. When the oscillation occurs, the damping factor can be adjusted automatically to help the algorithm get rid of the oscillation. Oscillation detection is critical to the adaptive

damping technology, but it is very difficult to describe the characteristics of oscillation. So we define non-oscillation characteristics instead, in which it is easier to describe the number of generated cluster exemplars declined or unchanged during the iterative process (this is also the characteristic that algorithm will go toward convergence). In order to record the occurrences of non-oscillation characteristics in the iterative process, we design a movable monitoring window $K_b(j)$ ($j = 1,2,\ldots,t$, t is the window width), which can record t iterations continuously. For example, in the ith iteration, if non-oscillation characteristic emerge, $K_b(i) = 1$; otherwise, $K_b(i) = 0$. The criterion to judge whether an oscillation occurs is designed as follows: If the number of values $K_b(i) = 1$ in K_b is smaller than two-third window width, then the oscillation emerges. This criterion is a kind of tolerance design, which considers a few occasional oscillations and the unstable stages at the beginning of the algorithm.

Next is the computational complexity analysis of ADStream algorithm. Firstly, some symbols should be described in advance:

- D: feature space dimensions of data samples;
- S: the size of data sub-block arrived each time;
- C: the number of data clusters that the whole data stream sample set contains;
- s: the number of data sub-blocks that ADStream algorithm needs to traverse.

For the new arrival data sub-block, ADStream algorithm will iteratively calculate the pair-wise similarity s_{ij}, the cluster center c_{ik}, and the feature weighted coefficient ω_{ik} of S new arrival data samples in this sub-block. The computational complexity of this procedure is $O((S + C)CD)$. Suppose the maximum iteration number of ADStream algorithm is M, the complexity of dividing single data sub-block into clusters is $O((S + C)CDM)$. In fact, there are s data sub-blocks in data streams waiting for the proposed adaptive density data stream clustering algorithm to traverse, thus the final complexity of ADStream algorithm is $O(s(S + C)CDM)$.

Experimental Analysis

To test the effectiveness of proposed adaptive density data stream clustering algorithm, first we analyze the algorithm using simulated data stream on MOA analog data stream clustering platform and compare it with the classic DenStream algorithm [16]. The programming and operating environment of the algorithm is JDK 1.6, and use Eclipse SDK 3.4.1, WEKA 3.7.7 (Waikato Environment for Knowledge Analysis) and MOA-20120301 (Massive Online Analysis) platform. The operating system is Windows XP; the configuration of computer used for experiment is 2.6 GHz Intel CPU and 2 GB RAM.

The parameters of ADStream algorithm are set as: the attenuation factor $\lambda = 0.001$, shrinkage factor $\rho = 0.5$, similarity threshold $\varepsilon = 0.5$, minPoint = 10, weight threshold $\xi = 5$, initPoint = 1000, window size horizon = 1000; the parameters of DenStream algorithm are set as: $\varepsilon = 0.01$, $\mu = 1.1$, $\beta = 0.001$, initPoint = 1000, horizon = 1000.

The visualization results generated by ADStream algorithms and DenStream algorithm on MOA simulated environment are shown in Fig. 1.

According to Fig. 1, the clustering quality of ADStream algorithm is obviously superior to DenStream algorithms. DenStream algorithm is not sensitive to noise and often excessively deletes micro-clusters, which results in poor clustering accuracy. From the clustering results shown in Fig. 1, we can see that ADStream algorithm can detect clusters over a broad region. With the help of AP algorithm, ADStream algorithm will adaptively find the right cluster centers, which can guide the density method to group the neighbor points together and generate appropriate clusters.

The comparison of clustering purity of ADStream algorithm and DenStream algorithm is shown in Fig. 2. The blue curve represents ADStream algorithm, and the red curve represents DenStream algorithm. The simulation results indicate that ADStream algorithm is much more stable than DenStream algorithm with the continuous arrival of data streams, and the clustering purity of ADStream is relatively high, which means ADStream algorithm is not susceptible to outliers and has strong robustness for the ever-changing data stream structures.

In addition to using data stream simulation experiment platform, we also compare the clustering accuracy of ADStream algorithm with DenStream algorithm [16] and P-Stream algorithm [19] on KDD-CUP'98 and KDD-CUP'99 data sets of UCI machine learning database. KDD-CUP'98 is a relatively stable data set. It stores the information of charitable donation, totally having 95,412 records and 481 dimensions. Clustering on this data set can reflect the similarity of donation behaviors. Similar to the literature [14], 56 dimensional values are selected in the experiment and the input sequence of records is simulated as the arrival sequence of data streams. KDD-CUP'99 is a network intrusion detection data set with significant data evolutions. It consists of the original records of TCP connection in a local-area network. It is distributed irregularly and contains noise data. There are 23 different network intrusions or network attacks and 34 continuous attributes (without 7 discrete attributes) in the data set. In the experiment, 10 % data of KDD-CUP'99 data set is used for data stream clustering analysis, totally 49,032 test data.

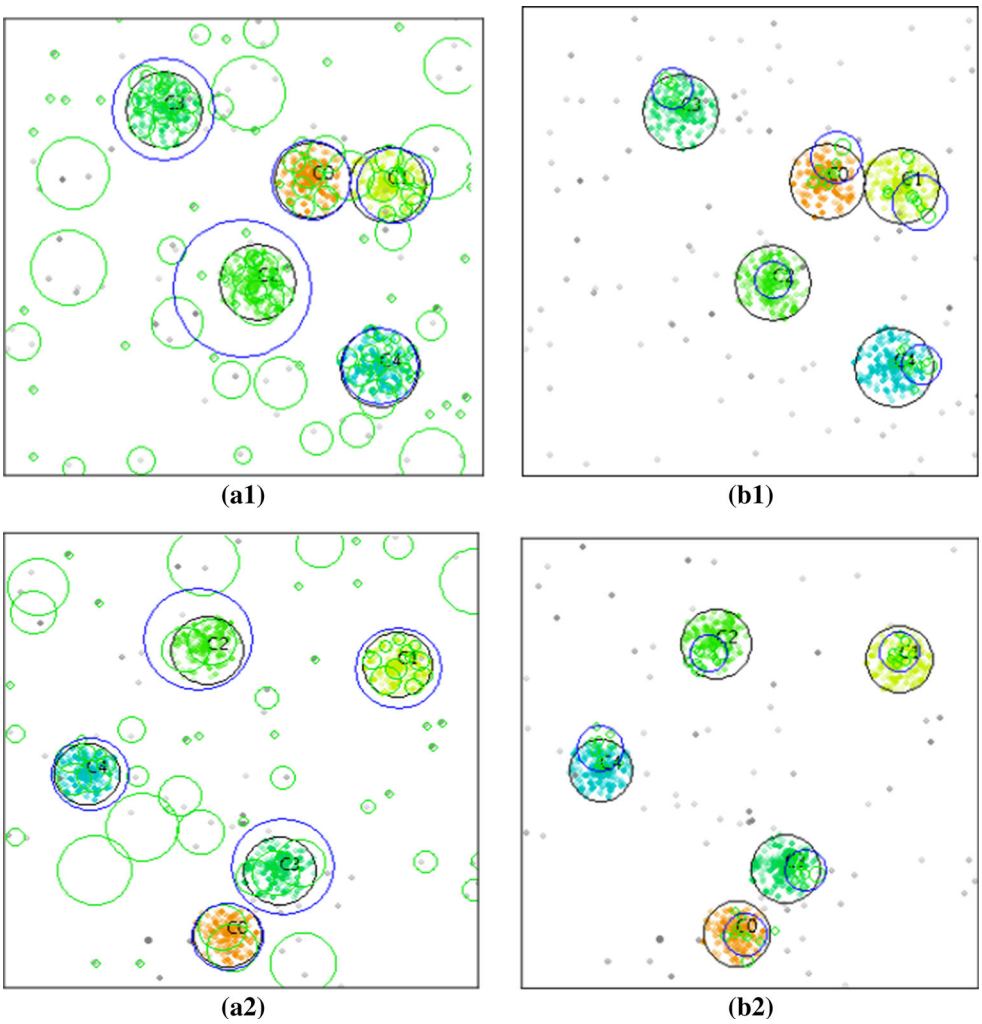

Fig. 1 Clustering results of ADStream algorithm and DenStream algorithm. **a1** Clustering results of ADStream algorithm, **b1** Clustering results of DenStream algorithm, **a2** Clustering results of ADStream algorithm, **b2** Clustering results of DenStream algorithm

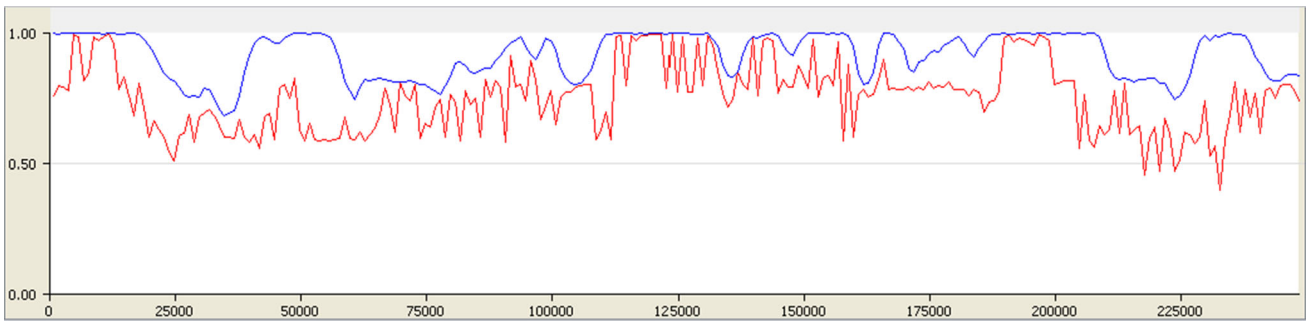

Fig. 2 Clustering purity of ADStream algorithm and DenStream algorithm

The parameters of ADStream algorithm are set as: similarity threshold $\varepsilon = 0.5$, the attenuation factor $\lambda = 0.001$, shrinkage factor $\rho = 0.5$, weight threshold $\xi = 5$. The speed of reading data streams is set as 200 data per second. The average clustering accuracy of these algorithms on KDD-CUP'98 and KDD-CUP'99 data sets is shown in Fig. 3.

Figure 3 shows that the clustering accuracy of ADStream algorithm is generally higher than that of DenStream algorithm and P-Stream algorithm. ADStream algorithm uses the micro-cluster analysis mechanism to determine whether the micro-cluster becomes the core micro-cluster or outlier cluster according to the threshold, and deletes the expired clusters. This maintains the

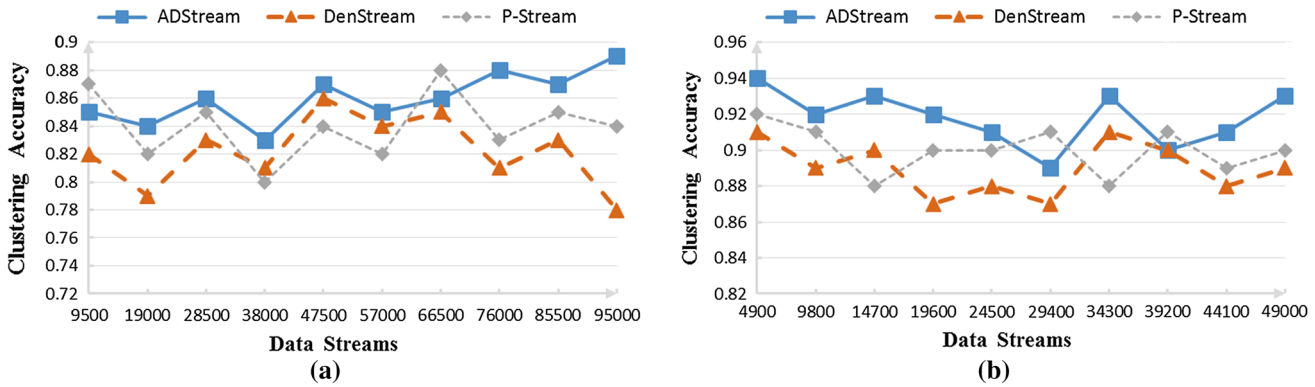

Fig. 3 Clustering accuracy of algorithms on different data sets. **a** Clustering on KDD-CUP'98 data set, **b** Clustering on KDD-CUP'99 data set

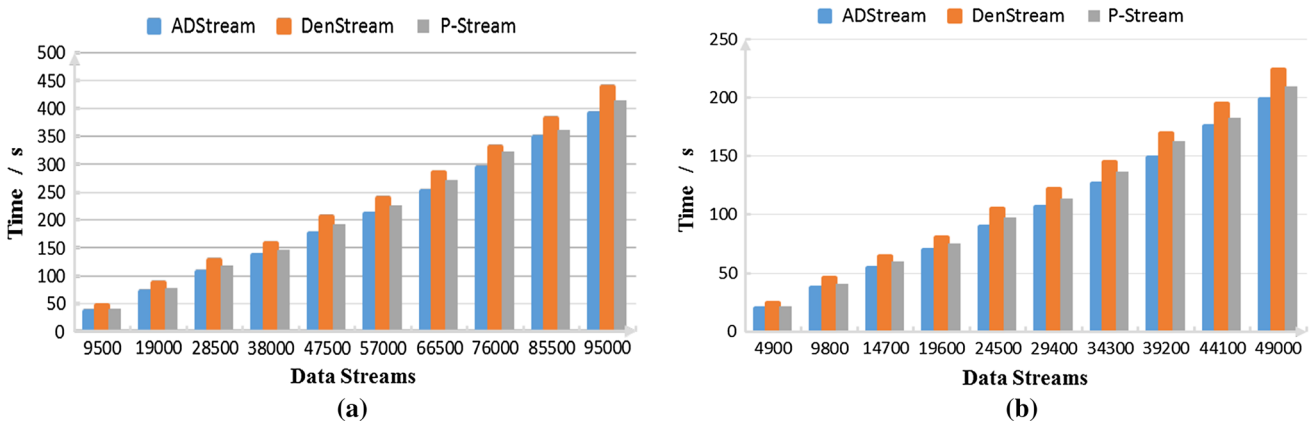

Fig. 4 Time cost of algorithms on different data sets. **a** Clustering on KDD-CUP'98 data set, **b** Clustering on KDD-CUP'99 data set

potential micro-clusters in data streams and removes noise points at the same time. Then analyze potential micro-clusters and update core micro-clusters and outlier-clusters, which ensure the high quality of clustering results. But DenStream and P-Stream algorithms lack the mechanism of distinguishing potential micro-clusters. Therefore, they need to consume a large amount of memory to process noises, and inappropriate division affects the precision of clustering. Figure 4 is the average clustering time of these algorithms on KDD-CUP'98 and KDD-CUP'99 data sets.

It can be seen from Fig. 4 that along with the increase in arrived data streams, the clustering time of algorithms is increasing as well. But for a certain amount of data streams, ADStream algorithm spends less time to generate appropriate clusters compared with DenStream algorithm and P-Stream algorithm. ADStream algorithm does not need to initialize the cluster number. With the help of the improved affinity propagation method, ADStream algorithm can dynamically adjust the cluster number and adaptively determine cluster centers according to the relationships among data points. Besides, ADStream algorithm introduces the sliding window mechanism and sets the attenuation factor so that data streams decay with time. The data in current window will have higher weights, and the decay rate of their weights will decrease; if the data have been out of the window, the decay rate will increase. Combined with the mutual transformation of density micro-cluster and candidate density micro-cluster, the time complexity of clustering procedure can be effectively reduced.

Conclusion

This paper reviews the development of current data stream clustering and proposes an adaptive density data stream clustering algorithm—ADStream. ADStream is composed of two stages: online micro-clustering and off-line macro-clustering. In the online part, the dynamic data streams are analyzed in a sliding window, and an improved affinity propagation clustering is applied to adaptively calculate the initial micro-clusters; in the off-line part, the clustering results in different time granularities are generated and updated by density grid clustering. The experiments show

that ADStream algorithm has strong abilities of detecting clusters in complex hybrid data streams. ADStream algorithm performs quite well on both artificial and real-world data sets compared with DenStream and P-Stream algorithm.

Although the proposed ADStream clustering algorithm is effective, there still exist some problems which need further research, for example: the influence of various parameter settings on the algorithm should be investigated; how to improve the robustness of the algorithm and eliminate the negative impact of noise in complex data streams on the clustering; whether the algorithm works well or not in diverse reality environments also remains to be tested.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No. 61379101), and the National Key Basic Research Program of China (No. 2013CB329502).

References

1. Huang XX, Huang HX, Liao BS, et al. An ontology-based approach to metaphor cognitive computation. Mind Mach. 2013;23(1):105–21.
2. Ding SF, Wu FL, Qian J, Jia HJ, Jin FX. Research on data stream clustering algorithms. Artif Intell Rev. 2015;43(4):593–600.
3. Byun SS, Balashingham I, Vasilakos AV, et al. Computation of an equilibrium in spectrum markets for cognitive radio networks. IEEE Trans Comput. 2014;63(2):304–16.
4. Zeng XQ, Li GZ. Incremental partial least squares analysis of big streaming data. Pattern Recogn. 2014;47(11):3726–35.
5. Mital PK, Smith TJ, Hill RL, et al. Clustering of gaze during dynamic scene viewing is predicted by motion. Cogn Comput. 2011;3(1):5–24.
6. Sancho-Asensio A, Navarro J, Arrieta-Salinas I, et al. Improving data partition schemes in Smart Grids via clustering data streams. Expert Syst Appl. 2014;41(13):5832–42.
7. Bian XY, Zhang TX, Zhang XL, et al. Clustering-based extraction of near border data samples for remote sensing image classification. Cogn Comput. 2013;5(1):19–31.
8. Amini A, Wah TY, Saboohi H. On density-based data streams clustering algorithms: a survey. J Comput Sci Technol. 2014;29(1):116–41.
9. Jia HJ, Ding SF, Xu XZ, Nie R. The latest research progress on spectral clustering. Neural Comput Appl. 2014;24(7–8):1477–86.
10. Yu J, Liu DQ, Tao DC, et al. Complex object correspondence construction in two-dimensional animation. IEEE Trans Image Process. 2011;20(11):3257–69.
11. Ding SF, Jia HJ, Zhang LW, et al. Research of semi-supervised spectral clustering algorithm based on pairwise constraints. Neural Comput Appl. 2014;24(1):211–9.
12. Yu J, Hong RC, Wang M, et al. Image clustering based on sparse patch alignment framework. Pattern Recogn. 2014;47(11):3512–9.
13. O'Callaghan L, Mishra N, Meyerson A, et al. Streaming-data algorithms for high quality clustering. In: Proceedings of IEEE international conference on data engineering, 2002, p. 685–694.
14. Aggarwal C, Han J, Wang J, et al. A framework for clustering evolving data streams. In: Proceedings of the 29th VLDB conference, 2003, p .81–92.
15. .Aggarwal CC, Han JW, Wang JY, et al. A framework for projected clustering of high dimensional data streams. In: Proceedings of the 30th international conference on very large data bases, 2004, p. 852–863.
16. Cao F, Ester M, Qian W, et al. Density-based clustering over an evolving data stream with noise. In: Proceedings of the SIAM conference on data ming, 2006, p. 328–339.
17. Chen Y, Tu L. Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, 2007, p. 133–142.
18. Zhu WH, Yin J, Xie YH. Arbitrary shape cluster algorithm for clustering data stream. J Softw. 2006;17(3):379–87.
19. Dai DB, Zhao G, Sun SL. Effective clustering algorithm for probabilistic data stream. J Softw. 2009;20(5):1313–28.
20. Pereira CMM, de Mello RF. TS-stream: clustering time series on data streams. J Intel Inform Syst. 2014;42(3):531–66.
21. Miller Z, Dickinson B, Deitrick W, et al. Twitter spammer detection using data stream clustering. Inf Sci. 2014;260:64–73.
22. Rodrigues PP, Gama J. Distributed clustering of ubiquitous data streams. Wiley Interdiscip Rev Data Mining Knowl Discov. 2014;4(1):38–54.
23. Albertini MK, de Mello RF. Energy-based function to evaluate data stream clustering. Adv Data Anal Classif. 2013;7(4):435–64.
24. Jin CQ, Yu JX, Zhou AY, et al. Efficient clustering of uncertain data streams. Knowl Inf Syst. 2014;40(3):509–39.
25. Vallim RMM, Andrade JA, de Mello RF, et al. Unsupervised density-based behavior change detection in data streams. Intell Data Anal. 2014;18(2):181–201.
26. Frey BJ, Dueck D. Clustering by passing messages between data points. Science. 2007;315(5814):972–6.
27. Wang KJ, Zheng J. Specified number of classes under the affinity propagation clustering fast algorithm. Comput Syst Appl. 2010;19(7):207–9.
28. Wang CD, Lai JH, Suen CY, et al. Multi-exemplar affinity propagation. IEEE Trans Pattern Anal Mach Intell. 2013;35(9):2223–37.
29. Mu Y, Ding W, Zhou TY, et al. Constrained stochastic gradient descent for large-scale least squares problem. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, 2013, p. 883–891.
30. Clerc M, Kennedy J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput. 2002;6(1):58–73.