Pattern Recognition Letters 33 (2012) 641–645

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Density-based hierarchical clustering for streaming data

Q. Tu [a], J.F. Lu [a,*], B. Yuan [b], J.B. Tang [c], J.Y. Yang [a]

[a] School of Computer Science, Nanjing University of Science & Technology, Nanjing, China
[b] Intelligent Computing Lab, Division of Informatics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[c] China Telecom Jiangsu Corp., Nanjing, China

ARTICLE INFO

Article history:
Received 17 May 2011
Available online 2 December 2011
Communicated by P. Franti

Keywords:
Streaming data
Density-based clustering
Hierarchical method

ABSTRACT

For streaming data that arrive continuously such as multimedia data and financial transactions, clustering algorithms are typically allowed to scan the data set only once. Existing research in this domain mainly focuses on improving the accuracy of clustering. In this paper, a novel density-based hierarchical clustering scheme for streaming data is proposed in order to improve both accuracy and effectiveness; it is based on the agglomerative clustering framework. Traditionally, clustering algorithms for streaming data often use the cluster center to represent the whole cluster when conducting cluster merging, which may lead to unsatisfactory results. We argue that even if the data set is accessed only once, some parameters, such as the variance within cluster, the intra-cluster density and the inter-cluster distance, can be calculated accurately. This may bring measurable benefits to the process of cluster merging. Furthermore, we employ a general framework that can incorporate different criteria and, given the same criteria, will produce similar clustering results for both streaming and non-streaming data. In experimental studies, the proposed method demonstrates promising results with reduced time and space complexity.

1. Introduction

The streaming data model refers to the scenario where a large volume of data arrives continuously and it is either unnecessary or impractical to store the entire data set in memory (O'Callaghan et al., 2002). Due to the following inherent restrictions of the streaming data model, significant new challenges have to be properly handled by algorithm designers (Henzinger et al., 1998).

(1) Data points can only be accessed in the order according to which they arrive;
(2) Memory is assumed to be limited compared to the size of input and random access to the data is not allowed due to intolerable time and space costs.

Streaming data mining is developed in order to extract the unknown and hidden knowledge from streaming data (Babcock et al., 2002; Dong et al., 2003). Among various data mining techniques, clustering is one of the most important. It can help analyze and uncover the underlying character of the data. Existing streaming data clustering algorithms are mainly based on partitioning methods, such as K-means, K-median, etc. For example, CLUSTREAM is one of the classical algorithms for streaming data clustering (Aggarwal et al., 2003). For non-streaming data, clustering algorithms can update each data point toward the most appropriate cluster within each iteration, but for streaming data, after the local clustering in the current iteration, the original data points cannot be accessed in following iterations. Consequently it's difficult for these data points to be updated effectively.

Hierarchical clustering analysis is a widely used clustering technique, which can be further divided into two categories: agglomerative methods, which proceed by making a series of merges of the n objects into more general groups, and divisive methods, which separate n objects successively into finer groups. In practice, hierarchical agglomerative clustering (HAC) is more commonly used and the number of clusters can be manually specified (Han and Kamber, 2006). For HAC, the popular criteria include the sum of within-group sums of squares (Ward, 1963) and the shortest distance between groups, which underlies the single-link method.

HAC algorithms, such as CURE and ROCK, use a static model to determine the most similar clusters to be merged in each level (Karypis et al., 1999). For merge-based hierarchical clustering, the limitation is that once a data point is assigned to a certain cluster, its membership cannot be modified. Franti and Virmajoki presented a cluster removal approach (Franti and Virmajoki, 2006), which does not suffer from this limitation. However, their approach requires multiple scans of data, which is infeasible for streaming data. In practice, there are two key issues to be considered: how to define an effective criterion for cluster merging and

* Corresponding author. Tel.: +86 25 84315751x824; fax: +86 25 84315960.
E-mail addresses: jstq2000@qq.com (Q. Tu), lujf@mail.njust.edu.cn (J.F. Lu), yuanb@sz.tsinghua.edu.cn (B. Yuan), tjb@telecomjs.com (J.B. Tang), yangjy@mail.njust.edu.cn (J.Y. Yang).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.11.022

how to define an efficient clustering framework with low time and space complexity. In this paper, our work will mainly focus on these two aspects.

Each cluster can be specified by a number of parameters, such as center, number of data points, density and variance. Traditional hierarchical clustering methods often ignore the density and variance properties of clusters when measuring the distance between two clusters, which may lead to unsatisfactory results. In fact, density plays an important role in clustering (Cao et al., 2006; Lu et al., 2008). OPTICS (Ankerst et al., 1999) is a classical agglomerative algorithm based on density where two factors (core-distance and reachability-distance) are used to measure the density of clusters. In OPTICS, a core needs to be determined for each cluster, which is difficult for streaming data. Wu and Tommy (2004) used the intra-cluster distance and inter-cluster distance to calculate the density. Chen and Tu (2007) also adopted the density-based grid clusters function to measure the distance between two clusters. In this paper, the concept of density (defined by variance and intra-cluster density) is also used to assess the clustering results. We show that some of the important properties such as the variance within clusters and the intra-cluster density can be estimated accurately by accessing the data points only once. With the help of this favorable characteristic, a novel hierarchical clustering algorithm for streaming data is presented.

As to cluster merging, previous methods mainly use Euclidean distance as the criterion. However, if only the center is used to represent all elements of each cluster, certain useful properties, such as variance and density, will be lost. This may in turn compromise the quality of clustering. Therefore, we propose to use density-based distance measurement instead of the Euclidean distance for cluster merging, and a new criterion is also designed for this purpose.

The remainder of this paper is organized as follows. Section 2 describes how to calculate the variance within the cluster, the intra-cluster density and the inter-cluster distance. Section 3 defines the criterion for cluster merging. The new clustering algorithm is presented in Section 4. Section 5 provides the experimental results and some analysis; and this paper is concluded in Section 6.

2. Variance and density of clusters

In HAC, cluster density is an important property. Ideally, data points in each cluster are expected to be as compact as possible, indicating a homogenous cluster. At the same time, the distinction between any two clusters needs to be as prominent as possible. For non-streaming data clustering, variance is often used as a measure of the density within each cluster. Fortunately, for streaming data clustering, variances within clusters can be calculated accurately according to Theorem 1 (Equitz, 1989).

Theorem 1. *Let G_x and G_y be two clusters, and let $\overline{x_x}, S_x^2, n_x$ be the mean, variance, and number of points of G_x, while $\overline{x_y}, S_y^2, n_y$ are those of G_y. The mean of the new cluster by merging these two data sets is:*

$$\bar{x} = \frac{\overline{x_x} \times n_x + \overline{x_y} \times n_y}{n_x + n_y}$$

The variance of the new cluster is:

$$S^2 = \frac{n_x - 1}{n_x + n_y - 1} S_x^2 + \frac{n_y - 1}{n_x + n_y - 1} S_y^2 + \frac{n_x n_y \|\overline{x_x} - \overline{x_y}\|^2}{(n_x + n_y - 1)(n_x + n_y)}$$

where $\|\overline{x_x} - \overline{x_y}\|^2 = (x_x - x_y)^T (x_x - x_y)$

Theorem 1 states that, given the variance and mean of each cluster, the variance and mean of the new cluster merged from the two clusters can be accurately calculated. This fact ensures that the density property is always available even if the data points can only be accessed once. During the iterative clustering procedure, the variance is updated successively to ensure that its value is up to date.

The distance measure within and between clusters is critical for hierarchical clustering. Traditional streaming data clustering techniques such as AGNES (Agglomerative Nesting) (Kaufman and Rousseeuw, 1990) use the center of each cluster as its representation and the Euclidean distance as the distance metric, which usually cannot achieve satisfactory results.

There are many ways to measure the distance between two clusters. The average distance between two clusters, C_i and C_j, can be defined as: $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} \|p - p'\|$, where n_i: the size of cluster C_i;n_j: the size of cluster C_j;p: data point in C_i;p': data point in C_j;$\|p - p'\|$: the distance between p and p'. Let $Intra(x)$ denote the intra-cluster density and $Inter(x_1,x_2)$ denote the inter-cluster distance. According to Theorem 2, their values can be calculated accurately from the statistical information from the previous iteration. The detailed proof is presented in the Appendix.

Theorem 2. *Let G_x and G_y be two clusters, and let $\overline{x_x}, \sum_{i=1}^{n_1} x_i^2, n_x$ be the mean, the sum of squares, the number of data points in G_x, while $\overline{x_y}, \sum_{j=1}^{n_2} y_j^2, n_y$ are those of G_y. The inter-cluster distance can be expressed as follows:*

$$Inter(G_x, G_y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \|x_i - y_j\|^2 = n_y \sum_{i=1}^{n_x} x_i^2 - 2 n_x n_y \overline{x_x} \overline{x_y} + n_x \sum_{j=1}^{n_y} y_j^2.$$

According to the above theorem, the average inter-cluster distance can be expressed as:

$$\frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \|x_i - y_j\|^2 = \frac{1}{n_x n_y} \left(n_y \sum_{i=1}^{n_x} x_i^2 - 2 n_x n_y \overline{x_x} \overline{x_y} + n_x \sum_{j=1}^{n_y} y_j^2 \right).$$

In the above equation, the mean of the cluster can be obtained according to Theorem 1. The number of data points is known in advance, and the sum of squares of data points can be obtained from the previous iteration. Initially, there is only one data point within a cluster and the sum of squares of data points $\sum x^2$ is just the square of the data point itself.

The intra-cluster density is defined as the squared sum of distances between each pair of points in the cluster. So, the intra-cluster density of G_x can be calculated as:$Intra(G_x) = \sum_{i=1}^{n_x} \sum_{j=i+1}^{n_x} \|x_i - x_j\|^2$ and the intra-cluster density of G_y can be calculated as: $Intra(G_y) = \sum_{i=1}^{n_y} \sum_{j=i+1}^{n_y} \|y_i - y_j\|^2$. Given cluster G as the new cluster after merging, its intra-cluster density is defined by:

$$Intra(G) = \sum_{i=1}^{n_x} \sum_{j=i+1}^{n_x} \|x_i - x_j\|^2 + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \|x_i - y_j\|^2 + \sum_{i=1}^{n_y} \sum_{j=i+1}^{n_y} \|y_i - y_j\|^2$$

Based on the above formula, the intra-cluster density of the new cluster G can be updated as:

$$Intra(G) = Intra(G_x) + Intra(G_y) + Inter(G_x, G_y).$$

3. The proposed method

3.1. Algorithm framework

The proposed algorithm is illustrated as follows:

Set the values for *min*, *max* and *target*.

Repeat until no more new data points
  Repeat until the number of clusters is equal to *max*
    Read a new data point into the window as a new cluster.
  End Repeat
  Conduct clustering (merging) until the number of clusters in the window is reduced to *min*./* Local clustering */
End Repeat
Conduct clustering (merging) until the number of clusters in the window is reduced to *target*.

In the above algorithm, the parameters include *target* (the final number of cluster), *min* (the minimum size of sliding window) and *max* (the maximum size of sliding window). Large *max* and *min* values can improve the clustering accuracy but the time cost increases accordingly. Smaller values, on the other hand, can reduce the time cost, but at the expense of clustering accuracy.

The variables to be updated in each round of iteration include: the number of data points in each cluster, cluster centers, the variance of each cluster, the sum of squares of data points (used for updating the $Inter(G_x, G_y)$), and intra-cluster density.

3.2. Merging criterion

To overcome the drawbacks of existing criteria for cluster merging, a new heuristic criterion defined as $\frac{1}{n}Intra(G) \times Variance(G)$ is proposed, which considers the impact of density and shape simultaneously and is similar to Wu's method (Wu et al., 2004). Although $\frac{1}{n_x n_y}Inter(G_x, G_y)$ can be directly used to determine whether to merge two clusters or not, it is preferable to calculate the value of $\frac{1}{n}Intra(G)$ where $n = n_x + n_y$ for each possible new cluster in the next round of iteration and choose the case with the smallest value as the optimal merging. As $\frac{1}{n}Intra(G)$ indicates the property of the new cluster by merging a pair of clusters, while $\frac{1}{n_x n_y}Inter(G_x, G_y)$ focuses on two separate clusters, it is expected that $\frac{1}{n}Intra(G)$ can perform better in most cases, as demonstrated by the experimental results in Tables 2–7.

The reason that $Variance(G)$, which represents a different way to measure the cluster density, is also incorporated into the proposed criterion is to make it better adapted to different cluster shapes (e.g., bar, ellipse). Although there is no proof of the optimality of our criterion, as will be shown in the experiments, this criterion works well in practice.

The streaming clustering algorithm presented in the paper is based on agglomerative methods with some new features. In our algorithm, a sliding window is used. There are three parameters *min*, *max* and *target* to be specified in advance ($max > min > target$). The value of *target* gives the number of clusters in the final step. The values of *min* and *max* are, respectively, the minimum number and maximum number of clusters in the process of clustering. During clustering, the number of clusters is kept between *min* and *max*. Merging of clusters will not stop until the number of clusters reaches *min,* while data accessing will not stop until the number of clusters reaches *max*. After all data points have been accessed, merging will continue until the number of clusters is equal to *target*.

3.3. Space complexity

Five variables are required to describe each cluster: the middle point $Mid(G)$, the number of data points $Count(G)$, the variance, the sum of squares of data points, and the Intra-cluster distance $Intra(G)$. The size of data depends on the operating system; we illustrate it here using the case of WinXP in following analysis. Typically, $Count(G)$ is stored as an integer, which usually requires 4 bytes. The variance, the sum of squares, and Intra-cluster distance are all stored in double precision, which requires a total of 24 bytes. The size of $Mid(G)$ is dependent on the dimension of data and we assume that it occupies m bytes. Given the maximum number of clusters in the sliding window (*max*), the total space cost is: $(4 + 24 + m) \times \max$ bytes $= (m + 28) \times \max$ bytes. Since m and *max* are constants, the space complexity is $O(1)$, which is significantly lower than that of other algorithms.

3.4. Time complexity

Compared with alternative existing algorithms, the time complexity of the proposed algorithm is also lower. For example, the time complexity of ROCK (Robust Clustering using links) is: $O(n^2 + nm_m m_a + n^2 \log n)$ (Han and Kamber, 2006). The time complexity of DBSCAN (Density-Based Spatial Clustering of Application with Noise) is $O(n^2)$ (Han and Kamber, 2006). In our algorithm, each time the amount of data to be read is max–min. If the size of the data set is n, there will be approximately $\frac{n}{\max - \min}$ times of data reading and $\frac{n}{\max - \min}$ times of local clustering. After each round of reading and local clustering, the number of clusters is reduced from *max* to *min* due to merging. In each round of clustering, max–min times of the merging happen. In order to carry out cluster-merging, we should find the minimum distance between clusters, the time complexity for this purpose is $O(\max^2)$, the time complexity of each local clustering is $O(\max^2(\max–\min))$. The total time complexity is $O(\max^2(\max - \min)\frac{n}{\max - \min}) = O(\max^2 n)$. If the *max* is constant, the total time complexity is $O(n)$. If the max is pertinent to n, the total time complexity is $O(\max^2 n)$.

4. Results and analysis

To evaluate the proposed algorithms, a series of experiments was conducted with six benchmark data sets. Four data sets were used in both stream and non-stream clustering. The remaining two data sets were only used in stream clustering due to their size (the running time of non-stream methods was intolerable). The properties of each data set and the corresponding parameter setting are shown in Table 1. The mean of variances, the overall mean of intra cluster distances, time cost (ms) were used to evaluate the performance of algorithms. If the cluster number is n, the mean of the variances is defined as $\frac{\sum_{i=1}^{n} Variance(G_i)}{n}$, and the overall mean of intra cluster distances is defined as $\frac{\sum_{i=1}^{n} Intra(G_i)}{\sum_{i=1}^{n} C^2_{Count(G_i)}}$, where the denominator represents the sum of all pairwise connections over every cluster. For comparison, another criterion related to density, $E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y}) = (n_x \times \overline{x_x} - n_y \times \overline{x_y})^T (n_x \times \overline{x_x} - n_y \times \overline{x_y})$, was also adopted in which $n_x \times \overline{x_x}$ is the weighted average of points in G_x and $n_y \times \overline{x_y}$ is the weighted average of points in G_y (i.e., the Euclidean distance between the centers of two clusters). Other criteria used were $\frac{1}{n}Intra(G) \times Variance(G)$, $\frac{1}{n_x n_y}Inter(G_x, G_y)$ and $\frac{1}{n}Intra(G)$.

Table 1
The properties of each data set and the corresponding parameters setting.

Datasets	Size	Target	Min	Max	Clustering methods
Iris	150	3	10	20	stream and non-stream
Wine	178	3	10	20	stream and non-stream
Breast Cancer Wisconsin (Diagnostic)	569	2	7	14	stream and non-stream
Abalone	4,177	29	90	180	stream and non-stream
Covtype	581,012	7	25	50	stream only
KDD-CUP99	4,800,000	5	15	30	stream only

4.1. Results

HAC is used as the non-stream method and the algorithm framework in section 3.1 is used as the stream method. Table 1 describes the properties of the datasets, the parameters setting of the algorithm and the clustering methods used. Note that only the 34 continuous variables of KDD-CUP99 were used. The experimental results are presented in Tables 2–7.

4.2. Analysis

From the above results, it is clear that the results of non-stream clustering using $\frac{1}{n}Intra(G) \times Variance(G)$ as the criteria are better than those using $E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$, $\frac{1}{n_x n_y}Inter(G_x, G_y)$ and $\frac{1}{n}Intra(G)$ as the criteria. In particular, as shown in Table 7, the clustering results with $\frac{1}{n_x n_y}Inter(G_x, G_y)$ and $\frac{1}{n}Intra(G)$ as the criteria are so inferior that they are out of the range of double type.

In the meantime, the running time of stream clustering is significantly less than that of the non-stream clustering. Comparing running times using the four different criteria, although the running time of $\frac{1}{n}Intra(G) \times Variance(G)$ is generally longer than that of other criteria, the clustering results are much better than others. In summary, the proposed algorithm framework is effective in handling both stream and non-stream data and can be applied with various criteria. Also, given the same criterion, the difference between the results of stream and non-stream methods is not significant.

As to related work, a recent study also used inter and intra cluster variance (Karkkainen et al., 2007), which bears some similarity with our method. However, the accuracy of clustering in each iteration cannot be guaranteed, while our method is always accurate. Furthermore, the size of the buffer is between 2000 and 4000 in Karkkainen's method (Karkkainen and Franti, 2007), while the size of the buffer is less than 100 in our method. Compared with Zhong's method, generally speaking, their criteria are better than ours. When two clusters are merged, two indexes: Inter-connectivity (IC) and Inter-similarity (IS) are used to rank the subgroup, which consider the distance, density and similarity comprehensively (Zhong et al., 2011). However, their method is based on multiple scans, which is not suitable for streaming data.

Table 2
The results on the Iris data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	0.53	1.08	70
	(non-stream)	0.53	1.08	1562
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	1.64	3.37	35
	(non-stream)	1.24	2.66	1118
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	0.56	1.62	48
	(non-stream)	0.53	1.09	685
$\frac{1}{n}Intra(G)$	(stream)	0.55	1.45	50
	(non-stream)	0.54	1.10	563

Table 3
The results on the Wine data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	15630	32082.63	172
	(non-stream)	15254	33726.2	9344
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	96681	199876	115
	(non-stream)	70681	190292.3	5555
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	16254	33726.2	123
	(non-stream)	15499	48590.86	2387
$\frac{1}{n}Intra(G)$	(stream)	15702	33179.36	134
	(non-stream)	15254	33726.2	2426

Table 4
The results on the Breast Cancer Wisconsin data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	273338	$2.82*10^5$	399
	(non-stream)	225339	$2.46*10^5$	473984
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	448981	$9.12*10^5$	275
	(non-stream)	411989	$8.86*10^5$	346407
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	476059	$6.67*10^5$	326
	(non-stream)	355912	$5.58*10^5$	144664
$\frac{1}{n}Intra(G)$	(stream)	482211	$5.21*10^5$	335
	(non-stream)	235097	$4.33*10^5$	145157

Table 5
The results on the Abalone data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	0.015	0.016851525	201958
	(non-stream)	0.013	0.012937738	58269441
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	0.1	0.045209064	136095
	(non-stream)	0.03	0.040713647	42009570
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	0.011	0.037290074	160289
	(non-stream)	0.011	0.029379745	18770044
$\frac{1}{n}Intra(G)$	(stream)	0.012	0.01885799	162454
	(non-stream)	0.011	0.011774436	19572218

Table 6
The results on the Covtype data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	$1.26*10^6$	$2.41*10^6$	10831813
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	$2.7*10^6$	$2.76*10^7$	7967694
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	$7.4*10^6$	$4.93*10^7$	8285309
$\frac{1}{n}Intra(G)$	(stream)	$2.4*10^6$	$1.42*10^7$	8279659

Table 7
The results on the KDD-CUP99 data set.

Criteria (Clustering type)		Mean of variance	Overall mean of intra cluster distances	Time cost (ms)
$\frac{1}{n}Intra(G) \times Variance(G)$	(stream)	$5.2*10^{15}$	$1.37*10^{11}$	25281462
$E\_dis(n_x \times \overline{x_x}, n_y \times \overline{x_y})$	(stream)	$5.8*10^{16}$	$1.87*10^{12}$	18601356
$\frac{1}{n_x * n_y}Inter(G_x, G_y)$	(stream)	–	–	–
$\frac{1}{n}Intra(G)$	(stream)	–	–	–

In order to calculate the Inter-distance, the boundary point should be used, but for streaming data, the detailed information of data points will be lost in the next iteration. By contrast, in our method, a few statistical parameters are employed to calculate the criteria for merging instead of using boundary points. The density and similarity properties are also taken into account. Although our criteria may be less accurate due to the use of statistical parameters instead of the original data, our method is more appropriate for streaming data.

5. Conclusion

In this paper, we present a density-based hierarchical method with sliding windows for effective streaming data clustering. In the proposed method, some important properties of the current clusters are preserved to calculate the density in each round of iteration, resulting in improved accuracy of clustering. In the meantime, the variance and intra-cluster density of clusters are both taken into account in the design of a new criterion for cluster merging. Experimental results suggest that the proposed method can noticeably improve the accuracy of clustering compared with traditional techniques based on the Euclidean distance and other criteria. As to further work, it is also possible to further improve the efficiency of our algorithm based on fast PNN algorithm (Franti et al., 2000).

Acknowledgements

This work was partially funded by Jiangsu Natural Science Foundation (Project No. BK2009489), National Natural Science Foundation of China (No. 60775015, 60905030), Priority Academic Program Development of Jiangsu Higher Education Institutions and Jiangsu Qinglan Project. We also appreciate the anonymous reviewers' constructive feedback and Prof. Steven B. Skaar's help on improving this manuscript.

Appendix A

Theorem 2. *Let G_x and G_y be two data sets, and let $\overline{x_x}, \sum_{i=1}^{n_1} x_i^2, n_x$ be the mean, the sum of squares, the number of data points in G_x respectively, while $\overline{x_y}, \sum_{j=1}^{n_2} y_j^2, n_y$ are, similarly, those of G_y. The inter-cluster distance can be expressed as follows:*

$$Inter(G_x, G_y) = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\|x_i - y_j\|^2 = n_y\sum_{i=1}^{n_x} x_i^2 - 2n_x n_y \overline{x_x}\overline{x_y} + n_x\sum_{j=1}^{n_y} y_j^2.$$

Proof.

$$\begin{aligned}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\|x_i - y_j\|^2 &= \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left(x_i^2 - 2x_i y_j + y_j^2\right)\\ &= \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} x_i^2 - 2\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} x_i y_j + \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} y_j^2\\ &= n_y\sum_{i=1}^{n_x} x_i^2 - 2\sum_{i=1}^{n_x} x_i\sum_{j=1}^{n_y} y_j + n_x\sum_{j=1}^{n_y} y_j^2\\ &= n_y\sum_{i=1}^{n_x} x_i^2 - 2n_x n_y\overline{x_x}\overline{x_y} + n_x\sum_{j=1}^{n_y} y_j^2 \qquad \square\end{aligned}$$

References

Aggarwal, C., Han, J.W., Wang, J., 2003. A framework for clustering evolving data streams. In: Proc. 29th Internat. Conf. on Very large databases, pp. 81–92.
Ankerst, M., Breunig, M.M., Kriegel, H.P., 1999. OPTICS: Ordering points to identify the clustering structure. In: ACM SIGMOD'99 Internat. Conf. on Management of Data, Philadelphia, pp. 40–60.
Babcock, B., Babu, S., Datar, M., 2002. Models and issues in data streams. In: Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, pp. 1–16.
Cao, F., Ester, M., Qian, W., 2006. Density-based clustering over an evolving data stream with noise. In: Proc. SIAM Conf. on Data Ming, pp. 328–339.
Chen, Y.X., Tu, L., 2007. Density-based clustering for real-time stream data. IN: Internat. Conf. on Knowledge Discovery and Data Mining, pp. 133–142.
Dong, G.Z., Han J.W., Laks, V.S., 2003. Online mining of changes from data streams: Research problems and preliminary results. In: Proc. 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams, pp. 225–236.
Equitz, W.H., 1989. A new vector quantization clustering algorithm. IEEE Trans. Acoustics, Speech, Signal Process. 37 (10), 1568–1575.
Franti, P., Virmajoki, O., 2006. Iterative shrinking method for clustering problems. Pattern Recognit. 39 (5), 761–765.
Franti, P., Kaukoranta, T., Shen, D.-F., Chang, K.-S., 2000. Fast and memory efficient implementation of the exact PNN. IEEE Trans. Image Process. 9 (5), 773–777.
Han, J.W., Kamber, M., 2006. Data mining: Concepts and techniques, 267–273.
Henzinger, M.R., Raghavan, P., Rajagopalan, S., 1998. Computing on data streams, Compaq Systems Research Center, Technical Report TR 1998-011.
Karypis, G., Han, E.H., Kumar, V., 1999. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer: Special Issue on Data Analysis and Mining.
Kaufman, L., Rousseeuw, P.J., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley& Sons, New York.
Lu, J.F., Tang, J.B., Tang, Z.M., Yang, J.Y., 2008. Hierarchical initialization approach for K-Means clustering. Pattern Recognition Lett. 29 (6), 787–795.
O'Callaghan, L., Mishra, N., Meyerson, A., 2002. Streaming-data algorithms for high-quality clustering. In: 18th Internat. Conf. on Data Engineering (ICDE'02), pp. 685–694.
Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. J. Amer. Statist. Assoc. 58 (301), 236–244.
Wu, S., Tommy, W.S., 2004. Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. Pattern Recognit 37 (2), 175–188.
Zhong, C., Miao, D., Franti, P., 2011. Minimum spanning tree based split-and-merge: A hierarchical clustering method. Informat. Sci. 181 (16), 3397–3410.