Adaptive Trace of Multi-dimensional Clusters by Monitoring Data Streams

Nam Hun Park[1], Kil Hong Joo[2*] and Su Young Han[1]

[1]*Dept. of Computer Science, Anyang University, 102 Samsungli, Buleunmyun, Ganghwagun, Incheon, Korea, 417-833*
[2]*Dept. of Computer Education, Gyeongin National University of Education, San 6-8 Seoksudong Manangu Anyangsi, Gyeonggi, Korea, 430-040*

[1]*{nmhnpark, syhan}@anyang.ac.kr,* [2*]*khjoo@ginue.ac.kr*

Abstract

In recent years, clustering data streams has been actively proposed in the field of data mining. In real-life domains, clustering methods for data streams should effectively monitor the continuous change of a data stream with respect to all the dimensions of the data stream. In this paper, a clustering method with frequency prediction of data elements is proposed. The incoming statistics of data elements in the monitoring range are maintained. For the range of elements with high density, the range is partitioned to detect the detailed boundary of clusters. To identifying the recent change of a data stream quickly, the support of elements is carefully monitored and predicted to determine partitioned ranges to become clusters. Considering the change of the data stream, a threshold is adaptively controlled by a prediction mechanism. By predicting the change of supports, the on-going change of a data stream can be reflected in real-time. The proposed method is comparatively analyzed by a series of experiments to identify its various characteristics.

Keywords: *Data Streams, Clustering, Data mining, Adaptive memory utilization*

1. Introduction

Data mining researches on data streams are motivated by emerging applications involving continuous massive data sets such as customer click streams, multimedia data and sensor data. A real-life data stream usually contains many dimensions and some dimensional values of its data elements may be missing [1]. In order to effectively extract the on-going change of a data stream with respect to all the subsets of the dimensions of the data stream, the abilities to trace its subclusters and to predict the change are very important.

In [2], K-median is a partitioning-based clustering algorithm and it finds the full-dimensional clusters of continuously generated data elements over a data stream. It regards a data stream as a sequence of stream chunks. A stream chunk is a set of consecutive data elements generated in a data stream. Whenever a new stream chunk containing a set of newly generated data elements is formed, the LSEARCH routine which is an O(1)-approximate K-medoid algorithm is performed to select *K* data elements from the data elements of the stream chunk as the local centers of the chunk.

CluStream [3] is proposed to find the clusters of data elements generated in an evolving data stream. It executes the conventional K-means method to find initial *q*

* Corresponding Author

pseudo clusters called *micro clusters*. A cluster feature vector [4] is used to represent the properties of a cluster. As a new data element arrives, the cluster features of the *q* micro clusters are continuously updated. The cluster feature vectors of all clusters at each specified timestamp are stored as a snapshot. The CluStream produces *k* final clusters called *macro clusters* by executing the K-means algorithm once more on the micro clusters of these snapshots.

All of these clustering algorithms for data streams are not targeted for subspace clustering. In recent applications, identifying the change of a data stream quickly enables to find the gradual change of embedded information, so that it can be timely utilized. A typical subspace clustering algorithm for a finite data set is CLIQUE [5] which searches all the subspaces of a data set in a bottom-up manner. CLIQUE [5] first determines the dense regions of each dimension as one-dimensional subspace clusters. After finding all of (*k-1*)-dimensional subspace clusters, candidate k-dimensional rectangular spaces are formulated by intersecting the regions of all the *(k-1)* dimensional subspace clusters. Subsequently, a *k*-dimensional subspace cluster is identified as a set of connected dense *k*-dimensional rectangular spaces. This process is repeated until no higher-dimensional subspace cluster is found. The drawback of this approach is that a data set is repeatedly scanned at every step of pruning candidate rectangular subspaces.

To accomplish the same objective, ENCLUS [6] uses the entropy of data elements. Basically, the clustering process of ENCLUS is the same as that of CLIQUE. It is motivated by the fact that a subspace with clusters typically has a lower entropy value than a subspace without any cluster. The entropy can measure the uncertainty of a random variable. When data points in a data set have a highly skewed probability mass function, their values are likely to fall within a small set of outcomes, so that the entropy of the data set is high. If the entropy of a subspace is smaller than a user-defined threshold, this subspace is excluded for the further clustering process.

FIRES [7] does an approximation technique based on the one-dimensional subspace clusters of each dimension. However, the common characteristic of these approaches is that they should scan a data set multiple times. Consequently, they can not be applied to an infinite on-line data stream. FIRES [7] is targeted to find subspace clusters in a high-dimensional data space. To overcome the curse of dimensionality in subspace clustering, it approximates multi-dimensional subspace clusters by one-dimensional clusters in each dimension. Initially, the one-dimensional clusters of each dimension are identified by using one of well-known clustering algorithms such as DBSCAN [8,10], k-means[2] or STING[9]. Subsequently, the intersected rectangular spaces of these one-dimensional clusters are prioritized based on the number of data elements.

In this paper, the grid-based index structure is adopted for subspace clustering over a data stream. Given a predefined sequence of dimensions $N_1 \rightarrow N_2 \rightarrow \ldots \rightarrow N_n$, initially an independent grid-list for each dimension monitors its one-dimensional subclusters at the first level of a monitoring tree. When a grid-cell of the grid-list for the dimension $N_k(1 \leq k \leq n)$ becomes a dense unit grid-cell, a set of new grid-lists are created as the children of the grid-cell. In order to enumerate all the possible two-dimensional subspaces of the dimension N_k uniquely, only for those dimensions which are after the dimension N_k in the dimension sequence, new grid-cell lists are created. Consequently, there are *(d-k)* distinct grid-cell lists are created as the children of the grid-cell. A grid cell of a node in the k^{th} level of a monitoring tree is corresponding to a rectangular subspace formed by intersecting the intervals of the grid-cells in the path from the root to the node containing itself. Also, to reflect the change of data streams in a real-time,

the support of a grid-cell is monitored and predicted by measuring velocity of density change.

This paper is organized as follows. In Section 2, the grid-cell structure is presented to maintain statistics of subspaces. In Section 3, the support prediction is introduced to measure when grid-cells become dense. In Section 4, various experiment results are comparatively analyzed to evaluate the performance of the proposed method. Finally, Section 5 presents conclusions.

2. Monitoring Data Streams

Given a data stream of an n-dimensional data space $\boldsymbol{N}=N_1\times\ldots\times N_n$, the region of a k-dimensional grid-cell ($1\le k\le n$) can be defined by a set of k intervals each of which lies in a distinct dimension. The rectangular space of a k-dimensional grid-cell defined by dimensions $N_1, N_2,\ldots,N_k$ is $\boldsymbol{RS}=I_1\times I_2\times\ldots\times I_k$ where I_1 , $I_2,\ldots,$ and I_k are intervals in the dimension $N_1, N_2,\ldots, N_k$ respectively. To monitor the distribution statistics of data elements in the rectangular space of such a grid-cell efficiently, a monitoring tree defined in Definition 1 is employed.

Definition 1. Structure of monitoring grid-cells

Given a predefined sequence of dimensions $N_1\rightarrow N_2\rightarrow\ldots\rightarrow N_n$, a monitoring tree of order m is defined for the current data stream $\boldsymbol{D}^t$ of a n-dimensional data space $\boldsymbol{N}=N_1\times\ldots\times N_n$ as follows;

1) A node maintains the following:

i) an entry $E<min, max, G[1,\ldots,m], next\_ptr>$

ii) a child array $U[1,\ldots m][1,\ldots,n-1]$: Each of m grid-cells can have at most $n-1$ child nodes

iii) T_{dim} : the dimension on which its grid-cells are defined

2) If a node p in the j^{th} level is a child of a grid-cell $q.G[i]$ $(1\le i\le m)$ of a node q in the $(j-1)^{th}$ level,

i) the node p becomes the first entry of a new grid-cell list $\boldsymbol{L}$ and the node p is called as the *head* of the list $\boldsymbol{L}$.

ii) the grid-cell $q.G[i]$ is called as the *parent grid-cell* of all the grid-cells in the grid-cell list $\boldsymbol{S}$.

iii) let $Ancestor(g^j)=\{g^1,\ldots,g^{j-1}\}$ denote the set of its ancestor grid-cells in the path from the root to the node. The rectangular subspace $\boldsymbol{RS}(g^j)$ of the grid-cell g^j is $\boldsymbol{RS}(g^j)=g^1.I\times g^2.I\times\ldots\times g^{j-1}.I\times g^j.I$ where $|g^i.I|=\lambda$ and $g^i.c/|\boldsymbol{D}^t|\ge S_{par}$ $(1\le i\le j-1)$.

iv) let $\boldsymbol{D}_g^t$ denote those data elements that are in the range of the grid-cell g^j, i.e., $\boldsymbol{D}_g^t=\{\ \boldsymbol{e}\mid \boldsymbol{e}\in\boldsymbol{D}^t$ and $\boldsymbol{e}\in\boldsymbol{RS}(g^j)\ \}$. The distribution statistics of data elements in the rectangular subspace $\boldsymbol{RS}(g^j)$ are monitored by the grid-cell $g^j(I,c,\mu,\sigma)$. In other words, the current number of data elements in $\boldsymbol{D}_g^t$ is monitored by $g^j.c^t$. In addition, the average and standard deviation of data elements in $\boldsymbol{D}_g^t$ are monitored by $g^j.\mu$ and $g^j.\sigma$ respectively.

□

Given a predefined sequence of dimensions $N_1\rightarrow\ldots\rightarrow N_n$ and a partitioning factor h, grid-cell lists $\boldsymbol{L_1,\ldots, L_n}$ are created to maintain the one-dimensional grid-cells of each

one-dimensional data space respectively. Initially, each grid-cell list at the first level maintains *h* initial grid-cells and a single node is created to form each grid-cell list. As a new object o^t is arrived, the previous statistics are update as follows:

$$g.\mu^t = (g.\mu^v \times g.c^v + e^t)/g.c^t$$

$$g.\sigma^t = \sqrt{g.c^v \times (g.\sigma^v)^2 / g.c^t + \{(g.\mu^v)^2 + (e^t)^2\}/g.c^t - (g.\mu^t)^2}$$

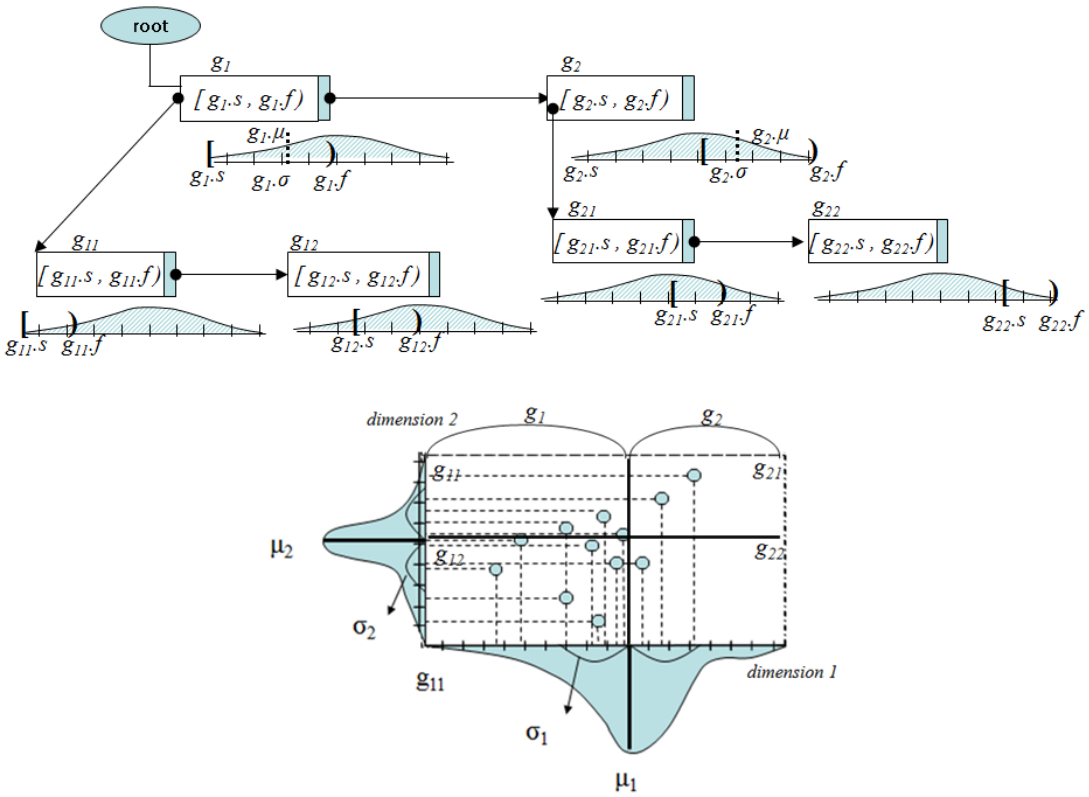


Figure 1. An example of grid-cells and data space

For the continuously generated data elements of a data stream, dense grid-cells in each grid-cell list $\boldsymbol{L_v}(1 \le v \le n)$ are recursively partitioned into *h* smaller grid-cells. Each of the child nodes is the head node of a new grid-cell list for two-dimensional grid-cells. The grid-cell list created for the dimension N_l $(v+1 \le l \le n)$ monitors the on-going distribution statistics of data elements in the two-dimensional rectangular subspace space $g_p^1.I \times N_l$. Given a grid-cell $g_q^2(I,c,\mu,\sigma)$ of the new grid-cell list in the second level, the two-dimensional rectangular subspace denoted by the grid-cell $g_q^2(I,c,\mu,\sigma)$ is $g_p^1.I \times g_q^2.I$.

Whenever a new data element $\boldsymbol{e^t} = <e_1^t, \ldots, e_n^t>$ is generated, according to the dimensional values of $\boldsymbol{e^t}$, the relevant paths of the monitoring tree are traversed from the root. Upon visiting a node in the k^{th} level of the monitoring tree, its child nodes are searched in a depth-first manner since several range lists can be created as the children of a single range. For each child node, the dimensional value of the new data element is

used to determine the right range among its ranges. For each identified range, the new element is processed by updating the distribution statistics of the range. If the size of the range is not a predefined minimum unit and just becomes dense ($\geq S_{par}$), it is partitioned into *h* smaller ranges.

When a grid-cell in the grid-cell list for the dimension N_l ($v+1 \leq l \leq n$) in the second level becomes dense, it is also partitioned into smaller grid-cells. Consequently, the number of grid-cells in the grid-cell list is increased. Furthermore, when it becomes a dense unit grid-cell, *(n-l)* new grid-cell lists for the subsequent dimensions are created in the third level as the children of the dense grid-cell as well. In such a way, the monitoring tree grows up to the n^{th} level at most. On the other hand, when a grid-cell *g* of a node *n* in a monitoring tree becomes sparse, it is merged to be a grid-cell of a rough range. Furthermore, all of its descendent nodes are pruned since they are also sparse.

3. Prediction of the Grid-cell Support

For a grid-cell in a monitoring tree, its support per time changes over time. Analyzing the past frequency of a grid-cell can help to predict its support in the future [11, 12]. However, to predict the support more accurately over time, more information should be maintained each grid-cell entry.

For a grid-cell, its support velocity is defined as the difference of its support. The V_{count} of the grid-cell means its velocity in the most recent time. When the current time is t^{th} time, the support of $(t+1)^{th}$ time can be predicted from V_{count}.

$$V_{count}{}^{t+1} = count^{t} - count^{t-1} \quad (1)$$

From the velocity $V_{count}{}^{t+1}$, the count at $(t+1)^{th}$ time can be predicted as follows:

$$P_{count}{}^{t+1} = count^{t} + V_{count}{}^{t+1} \quad (2)$$

The support at $(t+1)^{th}$ time can be predicted from $P_{count}{}^{t+1}$ as follows:

$$S^{t+1} = (count^{t} + P_{count}{}^{t+1}) / (|D^{t}| + count^{t+1})$$

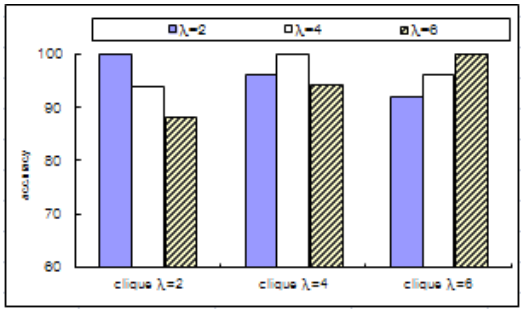
Let a grid-cell with the current support S^t would become a cluster after *v* times. Then, $S^{t+v} = (count^{t} + P_{count}{}^{t+v}) / (|D^{t}| + count^{t+v}) \geq S_{\min}$ is satisfied. From equations (1) and (2), the support is predicted by solving the time *v*.

4. Experiments

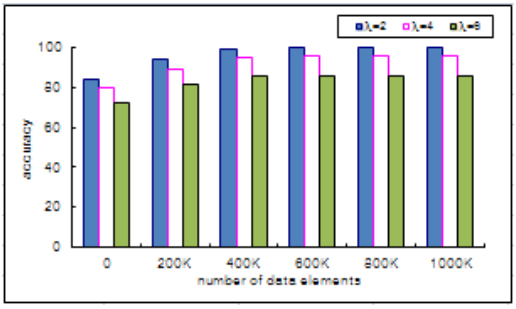
In order to analyze the performance of the proposed method, a data set containing one million 20-dimensional data elements is generated by the data generator used in ENCLUS [6]. The domain size of each dimension is set to 100. Most of data elements are concentrated on randomly chosen 10 data regions whose sizes in each dimension are also randomly varied. The conditions of most experiments are S_{min}=0.01, λ=2, and m=4 unless they are specified differently. The dimension of each level of a monitoring tree is determined dynamically. In all experiments, data elements are looked up one by one in sequence to simulate the environment of an on-line data stream.

The accuracy of the proposed method is presented in Figure 2. CLIQUE [5] is a well-known conventional grid-based subspace clustering algorithm for a finite data set and it is used as a yardstick to measure the accuracy of the proposed method. Among the data

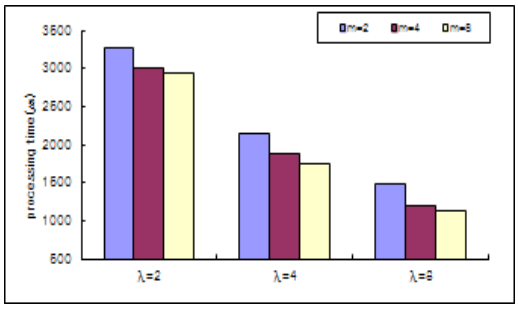
elements of a subcluster grouped by the proposed method, those data elements that are also grouped into the same subcluster by CLIQUE are defined as *correctly clustered data elements*. Figure 2(a) illustrates the accuracy of the proposed method for the four different values of λ. It is measured by the ratio of the number of correctly clustered data elements over the total number of data elements clustered by CLIQUE. When the value of λ for the proposed method is the same as that for CLIQUE, these two methods have the same accuracy. Figure 2(b) shows the variation of the accuracy as new data elements are generated. The accuracy of the subclusters obtained by the proposed method are measured relatively to the clustering result of CLIQUE when λ=2. Since lots of partitioning operations are occurred to find unit grid-cells in the early stage of subspace clustering, the accuracy of the proposed method is relatively low. However, as unit grid-cells are found by consecutive partitioning operations, the accuracy is increased gradually. Figure 2(c) shows the processing time of the proposed method. When the order is too small, *i.e.*, *m*=2, the number of sibling entries in each sibling list is increased rapidly, which prolongs the processing time.



(a) Accuracy comparison



(b) Accuracy variation



(c) Memory usage

Figure 2. Performance evaluations

5. Conclusion

As the number of dimensions for a data set is increased, subspace clustering is useful to analysis interesting groups in the subsets of the dimensions. However, because conventional subspace clustering methods need to create all the possible candidate subclusters and examine the data elements of a data set repeatedly for each candidate. They can not be used for an on-line data stream. In this paper, we have proposed a subspace clustering method over a data stream. By maintaining grid-based structure, the current statistics of a data stream are carefully monitored. As the support of each grid-cell is predicted with the support velocity, the rapid change of a data stream can be predicted for the real-time data mining.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science Technology(2012-0008326).

References

[1] M. Hua, J. Pei and X. Lin, "Ranking queries on uncertain data", The International Journal on Very Large Data Bases, vol. 20, no. 1, **(2011)** February, pp. 129-153.
[2] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha and R. Motwani, "STREAM-data algorithms for high-quality clustering", In Proc. of IEEE International Conference on Data Engineering,**(2002)** March.
[3] C. C. Aggarwal, J. Han, J. Wang, P. S. Yu, "A Framework for Clustering Evolving Data Streams", In Proc. VLDB 29th, **(2003)** Berlin.
[4] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: an efficient data clustering method for very large databases", In Proc. SIGMOD, **(1996)**, pp. 103-114.
[5] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", In Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM Press, **(1998)**, pp. 94–105.
[6] C. -H. Cheng, A. W. Fu and Y. Zhang, "Entropy-based subspace clustering for mining numerical data", In Proceedings of the fifth ACM SIGKDD International Conference on Knowledge discovery and data mining, ACM Press, **(1999)**, pp. 84–93.
[7] H. -P. Kriegel, P. Kroger, M. Renz and S. Wurst, "A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data", In Proceedings of the Fifth IEEE International Conference on Data Mining, **(2005)**, pp. 250-257.
[8] M. Ester, H. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases", **(1996)**.
[9] W. Wang, J. Yang and R. Muntz, "Sting: A statistical information grid approach to spatial data mining, **(1997)**.
[10] W. -j. Xu, L. -s. Huang, Y. -l. Luo, Y. -f. Yao and W. -W. Jing, "Protocols for Privacy-Preserving DBSCAN Clustering", Int. Journal of Security and Its Applications, vol. 1, no. 1, **(2007)** July, pp. 45-56.
[11] C. -I. Cha, S. -W. Kim, J. -I. Won, J. Lee and D. -H. Bae, "Efficient Indexing in Trajectory Databases", Int. Journal of Database Theory and Application, vol. 1, no. 1, **(2008)** December, pp. 21-28.
[12] J. Zhao, X. Li and P. Jin, "A Time-Enhanced Topic Clustering Approach for News Web Search", Int. Journal of Database Theory and Application, vol. 5, no. 4, **(2012)** December, pp. 1-10.

Authors

Nam Hun Park received his B.S., M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2000, 2002 and 2007. He was a post-Ph.D. at the Department of Computer Science, Worcester Polytech Institute, Worcester, MA. He is currently a professor of Department of Computer Science at Anyang University, Korea. His current interests include mining data streams.

Kil Hong Joo received his M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2000 and 2004. He is currently a professor of Department of Computer Education at Gyeongin National University of Education, Korea. His current interests include mining data streams, data analysis and smart learning.

Su Young Han received the B.S. degree, M.S. degree and Ph.D degree from Hanyang University, Seoul, Korea in 1991, 1993 and 2004 respectively, all in electronic engineering. Currently, he is an assistant professor in the Department of Computer Science, Anyang University, Anyang, Korea. Hs research interests include multidimensional signal processing, wavelets, coding and watermarking.