Effective Evaluation Measures for Subspace Clustering of Data Streams

Marwan Hassani[1], Yunsu Kim[1], Seungjin Choi[2], and Thomas Seidl[1]

[1] Data Management and Data Exploration Group
RWTH Aachen University, Germany
{hassani,kim,seidl}@cs.rwth-aachen.de
[2] Department of Computer Science and Engineering
Pohang University of Science and Technology, Korea
seungjin@postech.ac.kr

Abstract. Nowadays, most streaming data sources are becoming high-dimensional. Accordingly, subspace stream clustering, which aims at finding evolving clusters within subgroups of dimensions, has gained a significant importance. However, existing subspace clustering evaluation measures are mainly designed for static data, and cannot reflect the quality of the evolving nature of data streams. On the other hand, available stream clustering evaluation measures care only about the errors of the *full-space* clustering but not the quality of subspace clustering.

In this paper we propose, to the first of our knowledge, the first subspace clustering measure that is designed for streaming data, called *SubCMM*: **Sub**space **C**luster **M**apping **M**easure. *SubCMM* is an effective evaluation measure for stream subspace clustering that is able to handle errors caused by emerging, moving, or splitting subspace clusters. Additionally, we propose a novel method for using available *offline* subspace clustering measures for data streams within the *Subspace MOA* framework.

1 Introduction

Data sources are increasingly generating more and more amounts of data. Additionally, the huge advances of data sensing systems resulted in cheap means for satisfying the eagerness for collecting data with a high number of attributes. The big size of the data together with its high dimensionality motivated the research in the area of high dimensional data mining and exploration. Data stream is a form of data that continuously and endlessly evolves reflecting the current status of collected values. Clustering is a well known data mining technique that aims at grouping similar objects in the dataset together into same clusters, and dissimilar ones into different clusters, where the similarity is decided based on some distance function. Thus, objects separated by far distances are dissimilar and thus belong to different clusters.

Stream clustering algorithms search for clusters that are formed out of the streaming objects when considering all dimensions of these objects. We call them

J. Li et al. (Eds.): PAKDD 2013 Workshops, LNAI 7867, pp. 342–353, 2013.

in this context: *full-space* stream clustering algorithms to differentiate them from other types of stream clustering algorithms which consider all subgroups of dimensions while searching for clusters.

Evaluating full-space stream clustering algorithms, can done mainly by assessing: (a) the **efficiency** represented by the runtime, the memory usage or the number of microclusters processed by the algorithm when mining the stream with different speeds, and (b) the **effectiveness** represented by the quality of the resulted clusters which mainly compares the found evolving clusters to the ground truth ones. Most of these were inherited from the offline clustering world, only one was mainly designed for streaming algorithm (cf. CMM [21]).

In many applications of streaming data, objects are described by using multiple dimensions (e.g. the Network Intrusion Dataset [1] has 42 dimensions). For such kinds of data with higher dimensions, distances grow more and more alike due to an effect termed *curse of dimensionality* [7] (cf. the toy example in Figure 1). Applying traditional clustering algorithms (called in this context: *full-space* clustering algorithms) over such data objects will lead to useless clustering results. In Figure 1, the majority of the black objects will be grouped in a single-object cluster (outliers) when using a full-space clustering algorithm, since they are all dissimilar, but apparently they are not as dissimilar as the gray objects. The latter fact motivated the research in the domain of *subspace* and *projected clustering* in the last decade which resulted in an established research area for static data.

In parallel to developing these static data subspace clustering algorithms, a group of measures for evaluating the clustering quality of offline subspace clustering algorithms were established. Additionally, other measures were inherited from traditional full-space clustering world (e.g. RNIA, CE [29], Entropy [30], Accuracy [9] and F1).

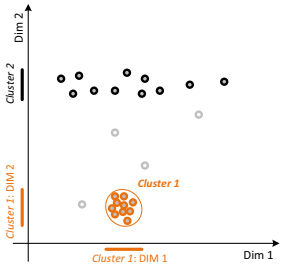


Fig. 1. An example of subspace clustering

For streaming data on the other hand, although a considerable research has tackled the full-space clustering, relatively limited work has dealt with subspace clustering. HPStream [3], PreDeConStream [17], HDDStream [27] and SiblingList [28] are the only works that have been done on projected/subspace stream clustering.

Almost all of the above mentioned algorithms have used the clustering purity [31] as the only measure for assessing the clustering quality. The purity was not

mainly designed for subspace stream clustering, and does not reflect the cases when clusters hidden in some subspaces are completely not discovered.

In this paper we propose, to the first of our knowledge, the first subspace clustering measure that is designed for streaming data, called *SubCMM*: **Sub**space **C**luster **M**apping **M**easure. *SubCMM* is an effective evaluation measure for stream subspace clustering that is able to handle errors caused by emerging, moving, or splitting subspace clusters. Additionally, we propose a novel method for using available *offline* subspace clustering measures for data streams within the *Subspace MOA* framework [14].

The remainder of this paper is organized as follows: Section 2 gives a short overview of the related work from different neighboring areas to full-space and subspace stream clustering algorithms as well as the measured used there. Section 3 introduces the *Subspace MOA* framework [14] and the novel method that we use for adapting the offline subspace measures as well as our *SubCMM* measure for using it under streaming environments. Our SubCMM measure is introduced in Section 4. The suggested measures are then thoroughly evaluated using the Subspace MOA framework in Section 5. Then we conclude the paper with a short outlook in Section 6.

2 Related Work

In this section, we list the related work from three areas: subspace clustering measures for static data, full-space stream clustering measures, and available subspace stream clustering and measures. Finally we will detail CMM [21].

2.1 Subspace Clustering Measures for Static Data

SubClu [23] is a subspace clustering algorithm that uses the DBSCAN [12] clustering model of density connected sets. PreDeCon [8] is a projected clustering algorithm which adapts the concept of density based clustering [12]. and the preference weighted neighborhood contains at least μ points. IncPreDeCon [22] is an incremental version of the algorithm PreDeCon [8] designed to handle accumulating data.

Evaluating the quality of the clustering delivered by the above algorithms was performed using a set of measures, which can also be categorized according to [26] depending on the required information about the ground truth clusters into two categories:

1. **Object-Based Measures:** where only information on which objects should be grouped together to form a cluster are used. Examples are: **entropy** [30] which measures the homogeneity of the found clusters with respect to the ground truth clusters, **f1**[6] which evaluates how well the ground truth clusters are represented and **accuracy** [9].
2. **Object- and Subspace-Based Measures:** where information on objects as well as their relevant dimensions (i.e. the subspaces where they belong

to) are used. Examples are: (a)**RNIA** [29] (Relative Non Intersecting Area) which measures to which extent the ground truth subobjects are covered by the found subobjects and (b) **CE** [29] (Clustering Error) which is an advanced version of RNIA and differs that it maps each found cluster to at most one ground truth cluster and also each ground truth cluster to at most one found cluster.

2.2 Full-Space Clustering Measures for Streaming Data

There is a rich body of literature on stream clustering. Convex stream clustering approaches are based on a k-center clustering [2,16]. Detecting clusters of arbitrary shapes in streaming data has been proposed using kernels [18], fractal dimensions [24] and density based clustering [10,11]. Another line of research considers the anytime clustering with the existence of outliers [15].

To reflect the quality of the *full-space* clustering algorithm, many evaluation measures are used. Some of those are inherited from the offline clustering world (cf. for instance: **SSQ** [13], **Silhouette Coefficients** [19] and **purity** [31]). Other measures were mainly developed specifically for assessing the quality of full space stream clustering algorithms like **CMM** [21] (cf. Section 2.4).

2.3 Subspace Clustering Measures for Streaming Data

Sibling Tree [28] is a grid-based *subspace* clustering algorithm where the streaming distribution statistics is monitored by a list of grid-cells. Once a grid-cell is dense, the tree grows in that cell in order to trace any possible higher dimensional cluster. HPStream [3] is a k-means-based *projected* clustering algorithm for high dimensional data stream. PreDeConStream [17] and HDDStream [27] are recent density-based projected stream clustering algorithms that were developed developed upon PreDeCon [8] in the offline phase.

Almost all of the above mentioned subspace stream clustering algorithms have used the clustering **purity** [31] as the only measure for assessing the clustering quality. Although the purity has proved to be popular and good when used with full-space stream clustering, it was not mainly designed for subspace stream clustering, and does not reflect the cases when clusters hidden in some subspaces are completely not discovered. Additionally, because of its property of neglecting the shaped of the ground truth, errors occurring on the borders of detected microclusters are not correctly punished due to the fast change of the shape or the position of the cluster.

2.4 Review: CMM [21]

We will review CMM separately here, since it is the only stream clustering measure that was designed for streaming applications, and because it is the full-space version of our proposed measure: SubCMM. CMM (Cluster Mapping Measure) consists of three phases.

First, each found cluster is assigned to one of the ground truth clusters based on class distribution in each cluster. In Figure 2(a), a plain circle represents a ground cluster, and a dashed circle means a predicted cluster. Each dot is a data point having its class label expressed by colors. Class frequencies are counted for each cluster, and each prediction cluster is mapped to a ground truth cluster that has the most similar class distribution. For Figure 2(a), the found cluster is mapped to the gray circle ground truth cluster.

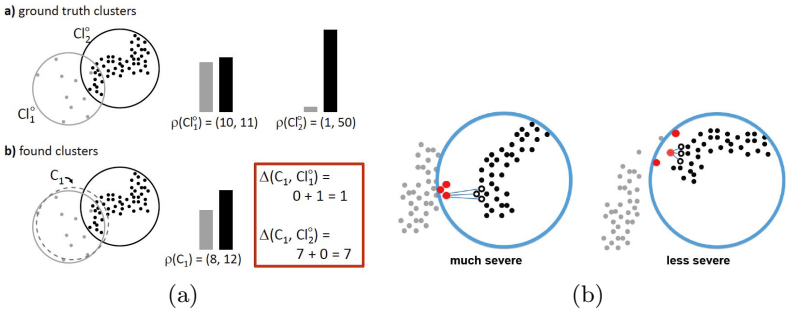


Fig. 2. CMM: (a) The mapping phase of the found cluster to a ground truth cluster, (b) Different penalties for two clusterings having the same accuracy

Second, the penalty for every incorrectly predicted point is calculated. In Figure 2(a), it can be seen that a lot of black points are included in the found cluster, which are incorrectly clustered, and some of the gray points are excluded in the cluster even if they are not noises. These points are "fault objects" and give they are penalized like this: $pen(o, C_i) = con(o, Cl(o)).(1 - con(o, map(C_i)))$ where C_i is a prediction cluster to which the object o belongs to, $Cl(o)$ is a ground truth cluster (hidden cluster) representing the original class label of o, and $map(C_i)$ is the hidden cluster on which C_i is mapped through the cluster mapping phase. The two clusters in Figure 2(b) have same accuracy, but it looks obvious that the left clustering has a bigger problem. If a fault object is closely connected to its hidden cluster, then the error becomes much severe since it was meant to be easily clustered. On the other hand, if the object has high connectivity to the found cluster, CMM allows a low penalty since it was hard to be clustered correctly. The connectivity $con(o, C)$ from an object to a cluster is exploiting the average k-neighborhood distance.

Third, derive a final CMM value by summing up all the penalties weighted over its own lifespan: $CMM(R, H) = 1 - \frac{\sum_{o \in F} w(o).pen(o,R)}{\sum_{o \in O} w(o).con(o,Cl(o))}$ Where R: represents the found clusters, H: represents the ground truth (hidden) clusters, O: is the set of objects o, F: is fault set, and $w(o)$: is the weight of o.

In the next section, we will explain our Subspace MOA tool, which we used to adapt the offline subspace clustering measures for the streaming scenario.

3 The *Subspace MOA* Framework [14]

OpenSubspace framework [25] was proposed to evaluate and explorate subspace clustering algorithms in WEKA with a rich body of most state of the art subspace/projected clustering algorithms and measures. In *Subspace MOA* [14], we have used this framework within the MOA framework [20] to bring the powerful subspace evaluation measures from the offline to the streaming world. Additionally, we built all the required additional units:

3.1 Subspace Stream Generator

Subspace MOA offers a synthetic random RBF subspace generator with the possibility of varying multiple parameters of the generated stream and its subspace events. One can vary: the number of dimensions, the number of relevant dimensions (i.e. the number of dimensions of the subspaces that contain the ground truth clusters), the number of the generated clusters, the radius of the generated clusters, the speed of the movement of the generated clusters, the percentage of the allowed overlapping between clusters, and the percentage of noise. Please note that some dimensions of a point could represent a noise within some subspace, while other dimensions could be a part of a ground truth cluster in other subspace. The generated noise percentage in this case represents a guaranteed noise in all subspaces. Subspace MOA gives also the possibility of reading external ARFF files.

3.2 Subspace Stream Clustering Algorithms

In order to have a rich number of variants, Subspace MOA offers the possibility of making your own flavor of subspace stream clustering algorithm. It follows for this reason the online-offline model used in most stream clustering algorithms (cf. [2], [10], [17]). In the **online phase**, a summarization of the data stream points is performed and the resulting microclusters is given by sets of cluster features $CF_i = (N, LS_i, SS_i)$ which represent the number of points within that microcluster, their linear sum and their squared sum, respectively. Subspace MOA offers three algorithms to form these microclusters and continuously maintain them. These are: CluStream, DenStream, and the cluster generator. In the **offline phase**, the clustering features are used to reconstruct an approximation to the original N points using Gaussian functions to reconstruct spherical microclusters centered at $c_i = \frac{LS_i}{N}$ with a radius: $r = \sqrt{\frac{SS}{N} - (\frac{LS}{N})^2}$ ($SS = \frac{1}{d}\sum_{i=1}^{d} SS_i$ and $LS = \frac{1}{d}\sum_{i=1}^{d} LS_i$). The generated N points are forwarded to one of the five subspace clustering algorithms. These are SubClu [23], ProClus [4], Clique [5], P3C, and FIRES. This results with 15 different combinations of algorithms that can be tested.

3.3 Subspace Stream Evaluation Measures

Then we have adapted the most famous offline subspace clustering measures (CE [29], Entropy [30], F1 [6], RNIA [29]) measures to the streaming scenario. Additionally, we have implemented our novel **SubCMM** measure (cf. Section 4). The user has the possibility to select the evaluation frequency, the window size is then set accordingly, and the evaluation measure is applied over the found clusters when compared against the ground truth clusters within that window. The output of these evaluation measures is delivered to the user according to the MOA conventions in three ways: (a) in a **textual form**, where summarization values are printed gradually in the output panel under the "Setup" tab as the stream evolves, (b) in a **numerical form**, where recent values are of all measures are printed instantly under the "Visualization" tab, and (c) in a **plotted** form of a selected measure from the recent values. The evolving of the final clustering of the selected subspace clustering algorithms as well as the evolving of the ground truth stream is visualized in a two dimensional representation. Users can select any pair of dimensions to visualize the evolving ground truth as well as the resulted clustering. Different to MOA, Subspace MOA is able to visualize and get the quality measures of arbitrarily shaped clusters.

4 *SubCMM*: Subspace Cluster Mapping Measure

We adopted important concepts of CMM (cf. Section 2.4) and revised its internal structure to develop a novel evaluation measure for subspace clusterings. The motivation for having a special subspace version of CMM becomes clear when using CMM directly for subspace clustering scenarios. Consider the matrix representation of data in Figure 3, where columns represent the objects and rows represent the attributes. Thus, each object is represented by a column, where its lines represent the attributes of this object. Assume that neighboring columns represent neighboring objects. Each circle is an attribute value of an object and the color denotes its class label (gray means noise). Blue, red, purple and orange subspace colors represent ground truth classes and the green dashed rectangle represents the found cluster of some stream clustering algorithm. In Figure 3(a), the found cluster is delivered by a full-space stream clustering algorithm, and thus it contains only complete columns in the matrix representation. CMM would not be able to map the found cluster C to the class blue since no obvious single class lable for each object can be found. Additionally, a subspace stream clustering algorithm would deliver clusters that look like C in Figure 3(b). Here, clusters could contain an arbitrary number of rows. Again, data objects in different clusters are defined in different spaces so we cannot simply count objects to compute class distributions as in CMM. We propose checking the class label of each attribute value (represented here by circle), instead of objects, we call it: Subobjects e_{ij}. Thus, in the matrix representations in Figure 3(b), it seems reasonable to assign the found cluster to class blue, since it contains 13 blue circles, one red circle and one noise circle. A similar discussion was mentioned in [29], to define the distance between subspace clusters.

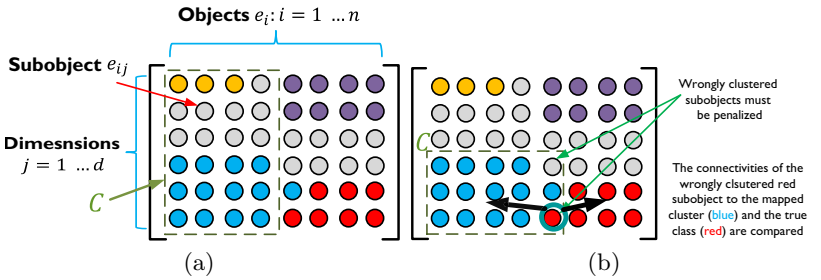


Fig. 3. (a) Full-space clustering and CMM over the matrix representation of subobjects, (b) Subspace clustering idea using SubCMM and penalizing fault subobjects

Thus, the penalty calculations in current CMM should be changed according to the revised clustering mapping phase. As we construct class distributions in a cluster in the matrix-element-wise way, the fault set consists of fault *matrix elements*, and a fault object o in $pen(o, C_i)$ is to be replaced with a fault matrix element e_{ij}, which is the $j-th$ subobject of $i-th$ object (cf. Figure 3(b)). Thus the penalty for each wrongly clustered subobject e_{ij} can be calculated as: $pen(e_{ij}, C) = con(e_{ij}, Cl(e_{ij})).(1 - con(e_{ij}, map(C_i))$. To calculate the penalty in this fashion, we have to define the connectivity between a subobject e_{ij} and a subspace cluster C. In CMM, the connectivity is based on average k-neighborhood distance, which computes Euclidean distance between two data objects and only the difference between attribute values of a same dimension is needed. In the SubCMM, we consider additionally the distance between different dimensions: $con(e_{ij}, C) = subcon(e_{ij}, C).objcon(e_{ij}, C)$ where $subcon(e_{ij}, C)$, the subspace connectivity, represents how much e_{ij} is connected to the subspace of C, and $objcon(e_{ij}, C)$, the object connectivity, means how much e_{ij} is connected to the (sub)objects of C. We define the object connectivity as:

$$objcon(e_{ij}, C) = \begin{cases} 1 & \text{if } [e_{ij} \in C] \text{ or if } [e_{ij} \notin C] \text{ AND } [knhObjDist^S(e_{ij}, C) < knhObjDist^S(C)] \\ 0 & \text{if } C = null \\ \frac{knhObjDist^S(C)}{knhObjDist^S(e_{ij}, C)} & \text{else} \end{cases}$$

where $knhObjDist^S(e_{ij}, C)$ is the average k-neighborhood distance from e_{ij} to the subobjects in C within the subspace S, and $knhDist^S(C)$ is the average k-neighborhood distance between objects in C within S. The subspace connectivity $subcon(e_{ij}, C)$ is similarly defined as:

$$subcon(e_{ij}, C) = \begin{cases} 1 & \text{if } [j \in S] \text{ or if } [j \notin S] \text{ AND } [knhDimDist^{e_{ij}}(j, C) < knhDimDist^{e_{ij}}(C)] \\ 0 & \text{if } C = null \\ \frac{knhDimDist^{e_{ij}}(C)}{knhDimDist^{e_{ij}}(j, C)} & \text{else} \end{cases}$$

where: $knhDimDist^{e_{ij}}(j, C)$ is the average k-neighborhood distance from the vector $v_j = [e_{aj}]$ where $a \in C$ to all the vectors $v_l = [e_{al}]$ where $a \in C$ and all $l \in S$ constructed from the objects of C defined in S.

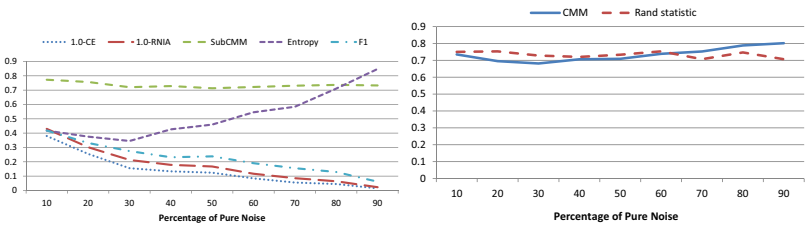
And $knhDimDist^{e_{ij}}(C)$ is the average k-neighborhood distance between vectors v_l constructed from C as above. One can regard this as performing the same procedure of calculating object connectivity on a transposed data matrix. Finally, we have to compute the final SubCMM value with the revised penalties. In this phase, we can just follow the CMM, but the fault object o must be a fault matrix element e_{ij}, and the weights of $e_{ij}s$ are equal when they belong to a same object.

$$SubCMM(R, H) = 1 - \frac{\sum_{e_{ij} \in F} w(i) \cdot pen(e_{ij}, R)}{\sum_{i \in DB} w(i) \sum_{j \in D} con(e_{ij}, Cl(e_{ij}))}$$

5 Experimental Evaluation

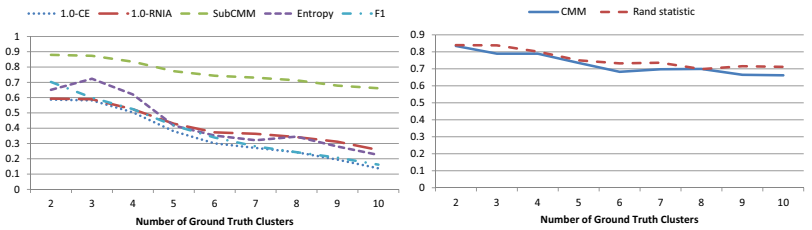
To test the performance of the suggested subspace stream clustering measures, we have used Subspace MOA [14] as the testing framework. We have used: CluStream+PROCLUS as the tested subspace stream clustering algorithm, and the RBF subspace stream generator as the ground truth stream. We compared the performance of SubCMM, RNIA, CE, Entropy and F1 as representatives of subspace stream clustering measures against the performance of CMM and Rand statistic as representatives of full-space stream clustering measures. In all of the following experiments, the stream and algorithm parameter settings, unless otherwise mentioned, are: number of stream attributes= 12, number of attributes of generated clusters= 4, number of generated clusters=5, noise level=10%, apeed of movement of clusters=0.01 per 200 points (which reflects the evolving speed of the concept drift of the stream), the evaluation frequency= 1000, and the decaying threshold= 0.1. Figure 4 compares the performance of subspace stream clustering measures (left) against full-space clustering algorithms when varying the pure noise percentage around the generated ground truth clusters. Apparently, most subspace measures are sensitive to the increasing of noise, different to the full-space ones. The stable high value that full-space measures give is due to the clusters which are accidentally created out the combination of clusters generated in the lower dimensions. Even for those clusters, when using the full-space measures, the quality does not decrease as in the subspace measures. Figure 5 shows the performance of both subspace and full-space measures when varying the number of generated ground truth clusters. Here the expected effect is a decreasing of the quality as the number of clusters increases. Again, full-space measures are relatively stable, while most subspace measures are sensitive.

Figure 6 depicts the quality of subspace and full-space measures when varying the radius of the generated clusters. Again the expected change here is a quality decrease as the radius increases. This is clear to see with most subspace measures, while only a slightly decrease can be seen on the full-space measures. The latter decrease, is due to the higher density of the noisy points around the accidentally



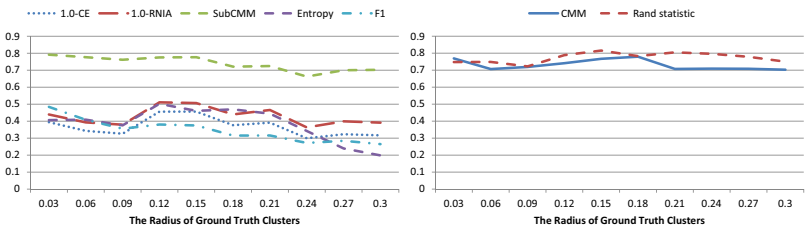
(a) (b)

Fig. 4. Clustering quality of a subspace stream clustering algorithm when varying the noise level using: (a) Subspace measures, (b) Full-space measures



(a) (b)

Fig. 5. Clustering quality of a subspace stream clustering algorithm when varying the number of clusters using: (a) Subspace measures, (b) Full-space measures



(a) (b)

Fig. 6. Clustering quality of a subspace stream clustering algorithm when varying the radius of clusters using: (a) Subspace measures, (b) Full-space measures

generated clusters in the full space. This noise might wrongly be added to the clusters in the full-space, and only this noise is punished by full-space measures and not the noise in lower dimensions.

6 Conclusion and Future Work

In this paper we have suggested a new way for evaluating stream subspace clustering algorithms by making use of available offline subspace clustering algorithms

as well as using the streaming environment to be able to handle streams. Additionally, we have suggested a first subspace clustering measure which was mainly designed for streaming algorithms. We have thoroughly tested these measures by comparing them to full-space ones. We could show the superiority of most of the suggested measures in the subspace streaming cases. In the future we plan to further improve the performance of SubCMM, and we want to test available subspace stream clustering algorithms using our measures.

Acknowledgments. This work has been supported by the UMIC Research Centre, RWTH Aachen University, Germany.

References

1. KDD Cup 1999 Data, `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
2. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proc. of the 29th Int. Conf. on Very Large Data Bases, VLDB 2003, vol. 29, pp. 81–92 (2003)
3. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for projected clustering of high dimensional data streams. In: Proc. of VLDB 2004, pp. 852–863 (2004)
4. Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C., Park, J.S.: Fast algorithms for projected clustering. SIGMOD Rec. 28(2), 61–72 (1999)
5. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of SIGMOD 1998, pp. 94–105 (1998)
6. Assent, I., Krieger, R., Müller, E., Seidl, T.: Inscy: Indexing subspace clusters with in-process-removal of redundancy. In: ICDM, pp. 719–724 (2008)
7. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
8. Bohm, C., Kailing, K., Kriegel, H.-P., Kroger, P.: Density connected clustering with local subspace preferences. In: ICDM, pp. 27–34 (2004)
9. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: ICDM, pp. 63–72 (2007)
10. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: 2006 SIAM Conference on Data Mining, pp. 328–339 (2006)
11. Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proc. of KDD 2007, pp. 133–142 (2007)
12. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of KDD 1996, pp. 226–231 (1996)
13. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier Science & Tech. (2006)
14. Hassani, M., Kim, Y., Seidl, T.: Subspace moa: Subspace stream clustering evaluation using the moa framework. In: Meng, W., Feng, L., Bressan, S., Winiwarter, W., Song, W. (eds.) DASFAA 2013, Part II. LNCS, vol. 7826, pp. 446–449. Springer, Heidelberg (2013)

15. Hassani, M., Kranen, P., Seidl, T.: Precise anytime clustering of noisy sensor data with logarithmic complexity. In: Proc. 5th International Workshop on Knowledge Discovery from Sensor Data (SensorKDD 2011) in Conjunction with KDD 2011, pp. 52–60 (2011)
16. Hassani, M., Müller, E., Seidl, T.: EDISKCO: energy efficient distributed in-sensor-network k-center clustering with outliers. In: Proc. SensorKDD 2010 Workshop in Conj. with KDD 2009, pp. 39–48 (2009)
17. Hassani, M., Spaus, P., Gaber, M.M., Seidl, T.: Density-based projected clustering of data streams. In: Hüllermeier, E., Link, S., Fober, T., Seeger, B. (eds.) SUM 2012. LNCS, vol. 7520, pp. 311–324. Springer, Heidelberg (2012)
18. Jain, A., Zhang, Z., Chang, E.Y.: Adaptive non-linear clustering in data streams. In: Proc. of CIKM 2006, pp. 122–131 (2006)
19. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley (1990)
20. Kranen, P., Kremer, H., Jansen, T., Seidl, T., Bifet, A., Holmes, G., Pfahringer, B., Read, J.: Stream data mining using the MOA framework. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part II. LNCS, vol. 7239, pp. 309–313. Springer, Heidelberg (2012)
21. Kremer, H., Kranen, P., Jansen, T., Seidl, T., Bifet, A., Holmes, G., Pfahringer, B.: An effective evaluation measure for clustering on evolving data streams. In: Proc. of KDD 2011 (2011)
22. Kriegel, H.-P., Kröger, P., Ntoutsi, I., Zimek, A.: Towards subspace clustering on dynamic data: an incremental version of predecon. In: Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques, StreamKDD 2010, pp. 31–38 (2010)
23. Kröger, P., Kriegel, H.-P., Kailing, K.: Density-connected subspace clustering for high-dimensional data. In: SDM, pp. 246–257 (2004)
24. Lin, G., Chen, L.: A grid and fractal dimension-based data stream clustering algorithm. In: ISISE 2008, pp. 66–70 (2008)
25. Müller, E., Assent, I., Günnemann, S., Jansen, T., Seidl, T.: Opensubspace: An open source framework for evaluation and exploration of subspace clustering algorithms in weka. In: In Open Source in Data Mining Workshop at PAKDD, pp. 2–13 (2009)
26. Müller, E., Günnemann, S., Assent, I., Seidl, T.: Evaluating clustering in subspace projections of high dimensional data. PVLDB 2(1), 1270–1281 (2009)
27. Ntoutsi, I., Zimek, A., Palpanas, T., Kröger, P., Kriegel, H.-P.: Density-based projected clustering over high dimensional data streams. In: Proc. of SDM 2012, pp. 987–998 (2012)
28. Park, N.H., Lee, W.S.: Grid-based subspace clustering over data streams. In: Proc. of CIKM 2007, pp. 801–810 (2007)
29. Patrikainen, A., Meila, M.: Comparing subspace clusterings. IEEE Transactions on Knowledge and Data Engineering 18(7), 902–916 (2006)
30. Sequeira, K., Zaki, M.: Schism: A new approach to interesting subspace mining, vol. 1, pp. 137–160 (2005)
31. Zhao, Y., Karypis, G.: Criterion functions for document clustering: Experiments and analysis. Technical report (2002)