

Instant Selection of High Contrast Projections in Multi-Dimensional Data Streams

Andrei Vanea, Emmanuel Müller, Fabian Keller, and Klemens Böhm

Technical University of Cluj-Napoca, Romania

`andrei.vanea@cs.utcluj.ro`

Karlsruhe Institute of Technology (KIT), Germany

`{emmanuel.mueller, fabian.keller, klemens.boehm}@kit.edu`

Abstract. In many of today’s applications we have to cope with multi-dimensional data streams containing dimensions which are not relevant to a particular stream mining task. These irrelevant dimensions hinder knowledge discovery as they lead to noisy distributions in the full dimensional space, while knowledge is hidden in some sets of dependent dimensions. This dependence of dimensions may change over time and poses a major open challenge to stream mining.

In this work, we focus on dependent dimensions having a high contrast, i.e. they show a clear separation between outliers and clustered objects. We present *HCP-StreamMiner*, a method for selecting high-contrast projections in multi-dimensional streams. Our quality measure (the *contrast*) of each projection is statistically determined by comparing the data distribution in a set of dimensions to their marginal distributions. We propose a technique for computing the score out of stream data summaries and a procedure for progressively tracking interesting subspaces. Our method was tested on both synthetic and real world data, and proved to be effective in detecting and tracking high contrast subspaces.

1 Introduction

Stream mining has seen an increased interest from researchers in the last decade. This interest can be attributed to at least two key factors: the prevalence of stream data and the challenges in its processing. Applications that produce stream data range from patient monitoring in medicine, click streams in e-commerce, up to energy surveillance with smart meters. What makes stream processing much more challenging than static data processing is the speed at which data is produced, the way in which stream data changes over time, the limited computational resources for processing, and the complexity of today’s data streams. We observe multi-dimensional streams to be such a complex data type where users are overwhelmed by many concurrent measurements. While some dimensions seem to be relevant at some time, this might change, and other sets of dimensions may become relevant. It is hard to capture this change by manual exploration of the exponential set of possible projections. Hence, automatic detection of *subspace projections* (i.e. sets of relevant dimensions), is an important issue for the instant analysis of multi-dimensional data streams. Such a subspace search can assist many data mining tasks.

As a use case for our subspace search, we focus on outlier detection. Outlier mining is an emerging research field for data streams [1–5]. As all of these existing techniques utilize the entire data space (all given dimensions), they fail in streams with many dimensions due to the well-known *curse of dimensionality* [6]. We tackle this challenge by subspace search and select the relevant projections for stream outlier mining. However, our technique is by far more general and could also be used for stream clustering or directly for user-driven exploration of relevant projections. In all of these scenarios, one requires local projections that reveal the hidden patterns (clusters, outliers, etc.) in a subset of the dimensions. We aim at subspaces with a clear separation between clusters and outliers. This separation, or *contrast*, is what we are looking for when selecting relevant projections. However, there are several challenges that need to be addressed to select high contrast projections in a data stream, including *instant selection of projections* and *the constant refinement of selected projections over time*.

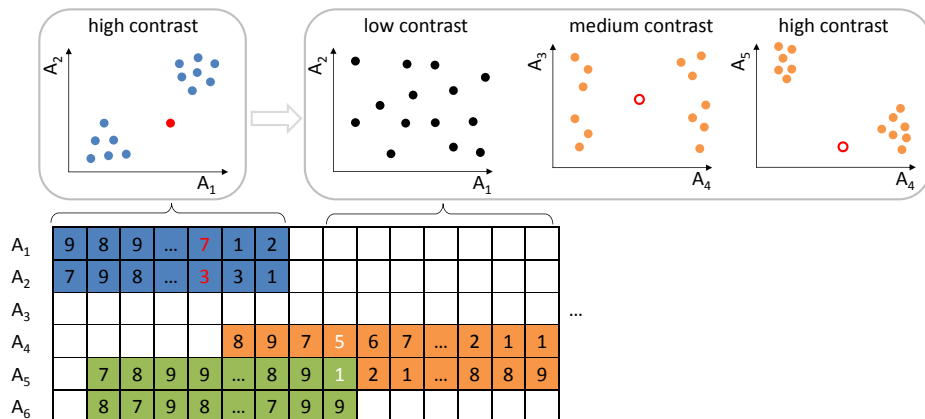


Fig. 1. Toy example for high contrast projections considering two stream snapshots

Monitoring and selecting relevant projections is demanding, as keeping track of every projection would require a daunting amount of memory, computational power and time. A core issue is that it requires an efficient contrast measure to keep track of relevant projections throughout the data stream. The projections of interest are those with both, dense and empty regions interleaved, i.e., they show a high contrast. Figure 1 shows an example of a data stream with six dimensions (lower half), denoted as A_1, \dots, A_6 , and the distribution of the data (upper half) in selected projections. As illustrated in our toy example, high contrast projections such as $\{A_1, A_2\}$ are relevant for outlier detection in the first snapshot of the stream. This situation of high contrast changes over time, and $\{A_1, A_2\}$ becomes irrelevant due to scattered data measurements in these two dimensions. A new projection becomes relevant and shows high contrast between clusters and outliers in $\{A_4, A_5\}$. To assess the relevance of projections,

we aim at a comparison of data distributions in different projections. As depicted for the low, medium, and high contrast subspaces, random distributions hinder the detection of hidden outliers, and outlier analysis should leave them aside. Measuring the contrast based on the data distribution is a major challenge for stream data. For our analysis it is important to summarize the data stream efficiently in order to correctly assess a contrast measure. Another open challenge for our selection is the changes over time. As illustrated in our example we have to keep track of changing projections.

In this work we present *HCP-StreamMiner*, a method for detecting high contrast subspace projections in stream data. The algorithm keeps a ranked list of *subspaces* that are continuously updated based on the new arriving data. The contrast of each subspace is statistically determined by comparisons of marginal density distribution functions and conditional density distribution functions. The two distributions are computed out of a stream summary. It is important to base on such summaries in order to efficiently assess our contrast measure. In order to address the issue of change in the data stream, we assess a given subspace based on the contrast, taking both old and new data, with a temporal decay, into account. Overall, we present the first method for subspace search on multi-dimensional data streams. It transfers the statistical selection of high contrast subspaces [7] that has been developed for static databases into the dynamic world of data streams.

2 Related Work

A number of techniques has been developed for dynamic stream processing and for outlier mining on data streams in particular. We review some approaches out of these two paradigms, in contrast to recent developments in static multi-dimensional databases.

Stream summarization aims at the efficient representation of stream history, with limited amount of memory and fast throughput rates. Histograms have been used to summarize data streams [8, 9]. Multi-dimensional data has been summarized into *micro-clusters* or *cluster features* [10], which store the linear sum and the sum of squared values as a data summary. A *sketch* of the data stream has been proposed for a summary that is incrementally updated with every multi-dimensional data object on the stream [11]. Furthermore, quad trees are proposed for efficient approximations of data distributions [12]. All of these techniques are limited to efficient summaries and do not address high contrast projections. They assist in efficient data access but do not provide any knowledge about the data. Users cannot derive the relevant projections neither for data analysis nor for interactive exploration with the stream.

Stream outlier analysis has been proposed recently. Outlier models range from distance-based outlier detection in data streams [1] to statistical approaches that measure the deviation of each object based on a model learned [2]. Furthermore, learning of different outlier types in multi-dimensional data streams has been proposed [3]. Some outlier models have been developed for sensor data [4]. Others focus on varying data rates of data streams using *any-time* approaches

for outlier detection [5]. In all of these approaches, the focus is on the outlier models and on the stream data rates. None of these approaches considers projections of the multi-dimensional data stream. Thus, they might miss outliers that occur only in a subset of the given dimensions.

Subspace Projections are well-known for static databases. Techniques such as *PCA* detect one global projection for the entire database [13]. However, as a major limitation of *PCA* and other dimensionality reduction techniques, they all provide a single projection only. Thus, they miss outliers hidden in different projections. Subspace outlier mining has tackled this challenge for high dimensional databases [14–16]. All of these methods detect individual projections for each outlier. As a pre-processing step to such outlier analysis, *subspace search* has been proposed. It uses entropy [17], density [18], or statistical approaches [7] for quality assessment of subspaces. All of these techniques succeed in detecting local projections for each outlier. However, they rely on multiple passes over the database and are not suitable for stream data. In this work, we overcome this issue and propose the first subspace search method for data streams.

3 Open Challenges

We observe several open challenges for stream outlier analysis on multi-dimensional data streams. In this work, we focus on two of them, addressing the main issues of instant outlier analysis on multi-dimensional streams:

Challenge 1 *Instant Selection of High Contrast Projections*

As first challenge, we observe the instant selection, i.e., a one-pass selection of high contrast projections. Traditional methods for high contrast projections require multiple passes over the database, which are not feasible for stream data. Measuring the contrast of arbitrary projections in an online fashion is still an open challenge. A solution has to distinguish between relatively even distributions in some dimension combinations vs. the high contrast of outliers and clusters in other dimension combinations. This calls for an online computation of the contrast measures, only based on concise data summaries derived from the stream.

Challenge 2 *Adaptive Refinement of Projections over Time*

A second challenge is the change of high contrast projections over time. Due to this change, relevant and irrelevant projections need to be adapted. A static solution for the entire data stream would not be sufficient especially in situations where data arrives in large quantities and at a high speed. One has to keep track of data changes and provide an updated list of top ranked subspaces. Refining the set of selected projections in an incremental fashion is the key challenge for such an adaptation over time.

4 HCP-StreamMiner

Based on the open challenges presented in the previous section, we present an algorithm for detecting high contrast projections in streaming environments. The algorithm takes as input the data arriving in the stream and outputs a ranked list of high contrast subspaces. This list can be used as input for other algorithms (e.g., outlier detection in our study) or point users to those subspaces which they may analyze in more detail with a manual exploration.

4.1 High Contrast Projections

Let us start with some basic notions for our formalization. We model a stream database DB as an infinite set of time points $DB = \{t_0, t_1, t_2, \dots\}$, with each time point i storing a d -dimensional vector $t_i \in \mathbb{R}^d$. The full data space is represented by the dimension set $DIM = \{D_1, \dots, D_d\}$. A subspace projection $S \subseteq DIM$ is a subset of this data space. We output a ranking of relevant subspaces for each time point. The selection and order of this ranking is based on the contrast function $contrast : \mathcal{P}(DIM) \rightarrow \mathbb{R}$. It provides the contrast of a subset of the dimensions. Please note that for processing reasons we base on a window-based computation of the contrast. As processing unit we consider a window $W = \{t_i, t_{i+1}, \dots, t_{i+k}\}$, a collection of multiple time points. The size of the windows is depending on the storage and processing capabilities of the underlying system. Each window is captured by a stream summarization technique [10] and all required measures are derived out of this summary. The required data distribution of each subspace is extracted from projections of the stream summaries on those particular dimensions. This allows the selection of subspaces with a one-pass solution based on stream summaries (cf. Section 4.2).

For our contrast function, we rely on a recent definition of *contrast* on static databases [7], as the basic notion for our selection and refinement of projections on data streams. Let us briefly review this contrast definition. A high contrast projection $S \subseteq DIM$ is a selection of dimensions which shows a data distribution with a high dependency between the selected dimensions. This dependency leads to clear clustered structures vs. individual outliers. This contrast is measured by comparing the *Marginal Density Distribution Function* (MDF) of a single dimension, and the *Conditional Density Distribution Function* (CDF), within a subregion. For every subspace, random subregions are selected as a conditional slice of the database. For each of these subregions, a *deviation* score is calculated, corresponding to the differences between the MDF and CDF. This comparison is performed for M random subregions. The overall *contrast score* of a projection is calculated as the mean of all deviation values obtained over the M iterations.

$$contrast(S) \equiv \frac{1}{M} \sum_{k=1}^M deviation(MDF_{D_j}, CDF_{sl})$$

The CDF refers to the conditional distribution in a random *slice* sl , with $|S| - 1$ conditions given by $[left_i, right_i] \forall D_i \in S \setminus D_j$. The MDF is the distribution of the entire database projected on dimension D_j . The contrast score is computed

from M comparisons of CDFs for randomly selected slices, and the MDF of the free (i.e. non-conditioned) dimension D_j . For a complete description of the MDF and CDF computation please refer to [7]. Our *HCP-StreamMiner* extends this contrast definition and computes it in a one-pass solution. Therefore, we will restrict the discussion to the novel stream properties only.

At each time point in the stream, the algorithm uses the data in the newly arrived data windows to update the stream summary. Out of this summary, it computes the contrast of S by updating its contrast score $contrast(S)$ to the new data distribution. Initially, the 2-dimensional subspaces from new data window are generated and ranked to capture the new trends in subspace contrast. Additionally, new candidate subspaces are generated from the existing subspaces, by progressively tracking higher dimensionality subspaces (cf. Section 4.3).

In addition to the number M of random selections, the algorithm takes three user controlled input parameters: top_k , gen_k and $rand_k$. The top_k parameter specifies the number of top ranked subspaces that are assessed at each time point. Since there is a large number of possible dimension combinations, assessing all subspaces requires to many resources. The top_k parameter therefore caps this effort. Furthermore, the parameters gen_k and $rand_k$ control the the generation of new candidate subspaces.

4.2 Subspace Search based on Stream Summaries

The data distribution over the stream is needed to compute the contrast score of a subspace at a point of time. Summaries of the data need to be built, such that the data distribution can be approximated accurately. Cluster Features (CF) are data structures used to summarize data by creating so-called *micro-clusters* [10]. A CF is a triple $C(CF1^x, CF2^x, n)$ with: the number of data points n summarized by the CF; $CF1^x$ and $CF2^x$ are two d -dimensional vectors storing the linear sum of each dimension and the sum of the squares of each data value in every dimension, respectively.

The total amount of information that a CF needs to store is $2 \cdot d + 1$ values, regardless of the number of data objects used to create the *micro-cluster*. $CF1^x$ can be viewed as the centroid of the cluster. Since the total number of parameters n is known, assuming a Gaussian distribution and also knowing the standard deviation σ , we can approximate the number of items around the centroid. The standard deviation can be computed using the following formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n} - \left(\frac{\sum_{i=1}^n x_i}{n}\right)^2} \iff \sigma = \sqrt{\frac{CF2^x}{n} - \left(\frac{CF1^x}{n}\right)^2}$$

We base our data stream summary on cluster features. Each data window is summarized by a CF. Additionally, a *random stretch* area RS in the full data space is monitored; one RS is constructed for each data window. A RS is a subregion of the full data space, which adheres to a set of domain constraints. For every dimension D_i two values $left_i$ and $right_i$ are randomly selected, such that $left_i < right_i$; they represent the start- and end-point of the subregion of

D_i that is summarized: $RS = \{(left_i, right_i) \mid left_i < right_i \forall D_i \in DIM\}$. We define N_{RS} as the number of items in the stream that respect all the conditions $(left_i, right_i)$ in RS . Therefore, the summary of the stream consists of a pair (CF, RS) , representing a summary of a particular time point in the stream. A strong point of using CFs is that they are built incrementally, accessing the data only once. This is a strong requirement in the data stream environment and also addresses the first challenge referred to earlier. A list of CFs is stored and maintained as follows. A queue storing CFs ($CFList$) is constructed at launch time, having a size in line with the memory capabilities of the system. Since the CF queue has a fixed size, whenever a new CF must be constructed, and the queue is full, the oldest CF will be deleted. The queue of RSs ($RSList$) is created and updated in a similar manner. We thus maintain a recent history of past windows.

Using the stream summaries, the subspace ranking is computed based on the relevance score, which quantifies the degree of contrast in a subspace (cf. Algorithm 1). The *ComputeRanking* procedure computes the new ranking for each subspace and updates the rankings. In each iteration, the algorithm first creates a so-called *test space* TS , by selecting a random dimension D_j from within the current subspace and removing it (Line 5). A slice sl is randomly selected in TS (Line 6), and its CDF is calculated (Line 7). The slice must be selected such that the boundaries of sl are within the boundaries of at least one element in $RSList$. In order to compute the MDF, the CF summaries are projected on dimension D_j . For the CDF calculation, one has to query the $RSList$ for those RS that overlap sl and get the number of objects in the overlapping regions. Based on the width of sl , the corresponding number of objects within the $left_i$ and $right_i$ endpoints of the selected sl can be approximated.

Algorithm 1 Compute Ranking

```

1: procedure COMPUTERANKING(subspace_list,  $M$ )
2:   for all subspaces  $S$  in subspace_list do
3:     contrast  $\leftarrow 0$ ;
4:     for  $k \leftarrow 1 \dots M$  do
5:        $TS \leftarrow S \setminus D_j$  ( $D_j$  randomly selected);
6:       select random slice  $sl$  from  $TS$ 
7:       compute CDF and MDF by  $CFList$  and  $RSList$  projections
8:        $run\_k = deviation(MDF_{D_j}, CDF_{sl})$ 
9:       contrast  $\leftarrow contrast + run\_k$ 
10:    end for
11:    contrast  $\leftarrow \frac{1}{M} \cdot contrast$ 
12:    UpdateRankingList( $S$ , contrast);
13:  end for
14: end procedure

```

For a particular subspace, we consider the dissimilarity between its MDF and CDF the percentage of the total area represented by their difference, with

respect to the MDF. Larger dissimilarities will produce higher contrast scores:

$$contrast = \frac{1}{M} \sum_{k=1}^M \int_{sl_{min}}^{sl_{max}} MDF_{D_j}(x) - CDF_{sl}(x) dx$$

Here sl_{min} and sl_{max} are the multi-dimensional boundaries of the random slice. The values calculated at each iteration are summed up, and the final contrast score for subspace S is returned as the mean of the individual scores. To address the changing nature of the stream and to emphasize the current one, the new and old values of the final contrast score are combined using a decay function.

4.3 Tracking Relevant Subspaces over Time

Efficient generation of new subspace candidates is important. As noted in our challenges, the data distribution within the stream might change over time. New relevant subspaces might appear in the stream. If our algorithm does not assign top rank to these subspaces, important information might get lost. A further issue is that a superspace of a known high contrast subspace might yield a higher contrast score. This superspace would be more interesting and valuable. Traditional methods for generating patterns of higher dimensionality rely on Apriori-like algorithms to do so. This bottom-up approach conflicts with the challenge of instantly selecting high contrast projections. Thus, our algorithm generates new subspaces in two phases: probability-based generation and random generation of candidate subspaces. We will discuss this process in the following.

The algorithm implements a progressive tracking of subspaces, in the sense that higher-dimensional subspaces are always generated and assessed later. With each iteration, the subspaces grow in dimensionality. The new subspaces are created by adding an additional dimension. Since there are many permutations possible, considering all higher-dimensional subspaces as candidates would result in a long candidate list. Processing such a list would require a large amount of time and resources, rendering the algorithm impractical for stream processing. For this reason we propose a so-called *dimension selection function* (DSF) which selects and returns the dimensions with a high probability of being part of a high contrast subspace. Let DIM denote the set of dimensions available, and \bar{S} the set of the dimensions not contained in S : $\bar{S} = DIM \setminus S$. The dimensions used to create a new higher-dimensional subspace NS are generated based on the existing subspace and the most promising dimensions selected by the DSF : $NS = S \cup DSF(\bar{S}, gen_k)$.

Therefore, gen_k parameter specifies the number of dimensions selected by DSF . Please note that the algorithm does not check all possible subspaces at each time point, due to the potentially high number of new candidate subspaces. In contrast to such a complete breadth first search, our algorithm tracks the higher dimensional subspaces in a best-first search based on DSF . Figure 2 shows how the relevant subspaces are used as seeds for the generation of new subspaces. The elements of the set $S_2^{t_0}$ of 2-dimensional subspaces generated at time point t_0 are used as seeds for the generation of the set $S_3^{t_0}$ of 3-dimensional subspaces, which

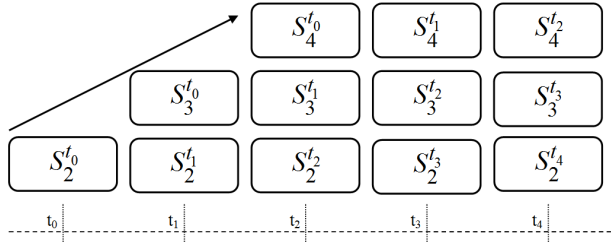


Fig. 2. Shifting the subspace dimensionality from a time point to the next.

is analyzed at time point t_1 . The process is applied to all subspaces at the top of the ranking list. We have implemented the *DSF* function as a *frequency score* which records the occurrences of the dimensions in the top-ranked subspaces. The *frequency score* is updated at each time point in the stream, based on the current subspace list.

Another set of *rand.k* new subspaces to check is randomly selected by choosing different dimensions. In this stage, we select the dimensions which have rarely been part of a high-contrast subspace so far. The rationale is that subspaces which have been less relevant in the past could become relevant in light of the new data. The final set of new candidate subspaces is given by the seeded and random candidates selected by our algorithm, which are then assessed by our contrast measure. Overall, this processing selects novel projections and refines the previously selected ones. We expect it to keep track of changes and to adapt to high-contrast projections in the data stream. We evaluate these properties in the following.

5 Experimental Results

We have conducted experiments on two different types of data: synthetic and real world data. We evaluated the quality of outlier detection based on synthetic data and compare our high contrast projections vs. the full dimensional space and other baseline solutions. Furthermore, we show the applicability of our method on a real world database.

5.1 Synthetic Data

We have generated multiple synthetic datasets with different dimensionalities (10, 20 and 30 dimensions). For each dataset we have inserted several high-contrast projections at random locations within the stream data. Synthetic outliers were hidden in the full space and in selected projections. The stream time-points when these outliers are set to appear have been chosen such that they materialize just before a subspace becomes relevant, during the time the subspace is considered relevant and much after a subspace has become irrelevant. We measure the accuracy of our method in comparison to several baseline solutions: (1) without subspace selection (full space), (2) in one-dimensional projections, (3) random projections, and (4) by using a traditional subspace search approach (HiCS [7]) which requires the entire stream as a static database. In all cases,

LOF [19] was used as outlier detection method to detect outliers in the selected subspaces. The overview of results is presented in Table 1. It contains the $F1$ score (i.e. the harmonic mean of precision and recall) as a quality measure. It shows that high contrast projections yield better results.

dimensions	full space	one dim.	random subspaces	HiCS (static)	HCP-StreamMiner
10	0.1905	0.3023	0.2250	0.7152	0.9256
20	0.1647	0.2353	0.1282	0.6887	0.9091
30	0.1205	0.2326	0.1013	0.6887	0.8943

Table 1. F1 score for outlier detection on synthetic data.

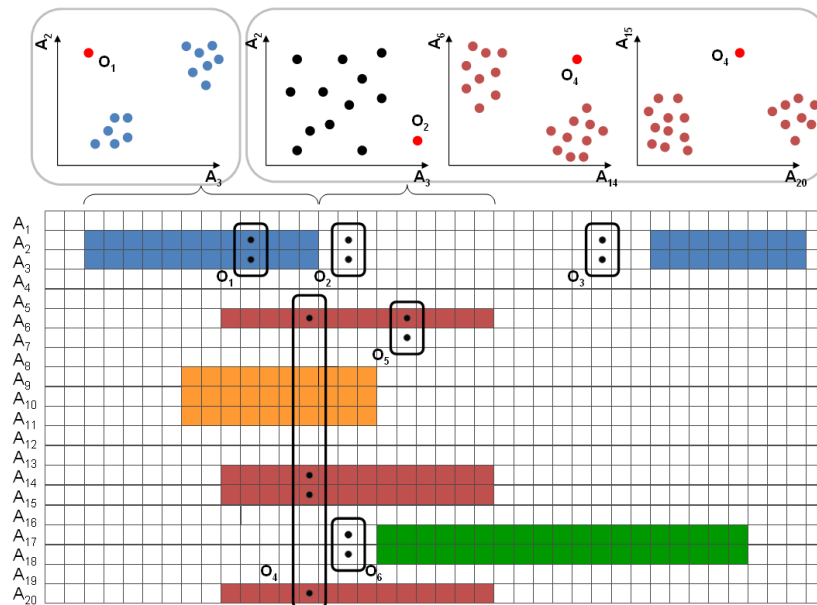


Fig. 3. *HCP-StreamMiner* tracking high contrast subspaces and outliers.

Let us discuss some details why *HCP-StreamMiner* outperforms the competitors. In Figure 3 we depict a down-scaled version of our synthetic data stream. Each block (column) represents 25 measurements within the 20 dimensions (rows). The dimensions belonging to the same hidden subspace are coded with the same color. Some of the hidden outliers are also listed, for demonstration purposes. In the first highlighted part of the stream, the subspace $\{A_2, A_3\}$ shows a high contrast; the outlier O_1 is clearly visible. This is also the case for O_4 in $\{A_6, A_{14}, A_{15}, A_{20}\}$ in the second snapshot of the stream. Such outliers can only be detected if the correct subspace is selected. They are only detected by *HCP-StreamMiner* and missed by other techniques. In contrast to this, accuracy of static selection (HiCS) is affected by the fact that static implementations do not take the transient nature of streams into account. A selected subspace will be kept selected for the entire data stream. This results in the detection of false positive outliers, and thus, lower F1 scores.

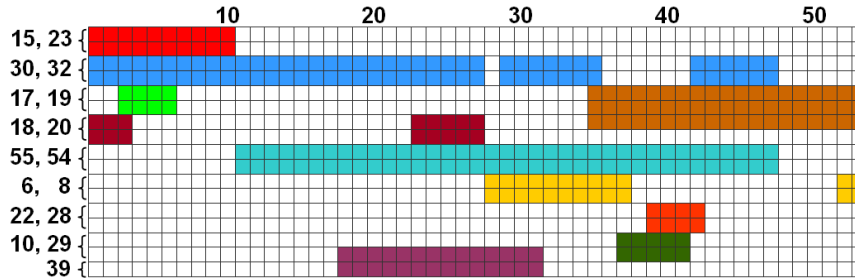


Fig. 4. Tracking high contrast subspaces in real-world data.

5.2 Real-World Data

We used a real world dataset containing energy consumption measurements. The records contain hourly sensor recordings (in MWh), from multiple building inside the KIT campus. A total of 62 buildings was selected as dimensions, for a time span of one year. The data was streamed with 24 objects in a single window – representing one day. Each object consists of all 62 buildings measured at the same moment of time. Therefore, each stream object represents the complete set of energy consumption in a multi-dimensional feature vector.

In Figure 4 we present an excerpt from the tracking of relevant subspaces in the real-world data. The numbers in the legend represent the building IDs. Each column corresponds to a calendar week. We observe that some of the selected subspaces tend to maintain their high contrast throughout the stream ($\{30, 32\}$, $\{55, 54\}$). However, some of them are considered relevant only in specific time points. For example, $\{15, 23\}$ and $\{17, 19, 18\}$ seem to be relevant only in the very early and late stage respectively. This real world example illustrates the change of high contrast subspaces over time. It shows that *HCP-StreamMiner* is not restricted to a single projection and finds multiple projections in the stream by adapting over time.

6 Conclusion and Future Work

In this paper we have presented a new data mining approach for the selection of high-contrast projections in multi-dimensional streams. The *HCP-StreamMiner* algorithm searches for dependent dimensions containing outliers in contrast to clustered data. The proposed algorithm assesses the contrast of subspaces in a one-pass solution using cluster features as stream summarization. It outputs a ranking of the most promising subspaces and updates this ranking incrementally to keep track of changes in the data distribution. Experiments show that our approach is able to detect outliers that are hidden in different projections of the data. Such outliers were not detectable by previous algorithms that search for outliers in the full dimensional space.

For future work we aim at an interactive exploration based on high contrast projections. To this end, we would like to integrate user feedback into our subspace selection process. Presenting a first rough approximation of high contrast

subspaces might give the user a first impression of dependent dimensions. For the following stream data, user feedback could be integrated as constraints into our refinement procedure. Hence, projections could be selected and refined based on both data distribution and user preferences.

Acknowledgements:

This work is supported by POSDRU/88/1.5/S/60078 co-funded by the European Social Fund, by the YIG program of KIT as part of the German Excellence Initiative, and by the German Research Foundation (DFG) within GRK 1194.

References

1. Angiulli, F., Fassetto, F.: Detecting distance-based outliers in streams of data. In: CIKM. (2007) 811–820
2. Yamanishi, K., Ichi Takeuchi, J., Williams, G.J., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Min. Knowl. Discov.* **8**(3) (2004) 275–300
3. Aggarwal, C.C.: On abnormality detection in spuriously populated data streams. In: SDM. (2005)
4. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online outlier detection in sensor data using non-parametric models. In: VLDB. (2006) 187–198
5. Assent, I., Kranen, P., Baldauf, C., Seidl, T.: Anyout: Anytime outlier detection on streaming data. In: DASFAA (1). (2012) 228–242
6. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is nearest neighbors meaningful. In: IDBT. (1999) 217–235
7. Keller, F., Müller, E., Böhm, K.: HiCS: High contrast subspaces for density-based outlier ranking. In: ICDE. (2012) 1037–1048
8. Furtado, P., Madeira, H.: *Vmhist*: Efficient multidimensional histograms with improved accuracy. In: DaWaK. (2000) 431–436
9. Muthukrishnan, S., Strauss, M.: Maintenance of multidimensional histograms. In: FSTTCS. (2003) 352–362
10. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: VLDB. (2003) 81–92
11. Thaper, N., Guha, S., Indyk, P., Koudas, N.: Dynamic multidimensional histograms. In: SIGMOD Conference. (2002) 428–439
12. Roh, Y.J., Kim, J.H., Son, J.H., Kim, M.H.: Efficient construction of histograms for multidimensional data using quad-trees. *Decision Support Systems* (2011) 82–94
13. Jolliffe, I.: *Principal Component Analysis*. Springer, New York (1986)
14. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. In: SIGMOD. (2001) 37–46
15. Kriegel, H.P., Schubert, E., Zimek, A., Kröger, P.: Outlier detection in axis-parallel subspaces of high dimensional data. In: PAKDD. (2009) 831–838
16. Müller, E., Schiffer, M., Seidl, T.: Statistical selection of relevant subspace projections for outlier ranking. In: ICDE. (2011) 434–445
17. Cheng, C.H., Fu, A.W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: KDD. (1999) 84–93
18. Kailing, K., Kriegel, H.P., Kröger, P., Wanka, S.: Ranking interesting subspaces for clustering high dimensional data. In: PKDD. (2003) 241–252
19. Breunig, M., Kriegel, H.P., Ng, R., Sander, J.: LOF: Identifying density-based local outliers. In: SIGMOD. (2000) 93–104