ViMEC: Interactive Application for Micro-Cluster Visualizations

Florian Schmidt[1] and Yannick Ehrenfeld[1]

[1]TU Berlin, Complex and Distributed IT Systems, Germany

Abstract

Digitalization increases the opportunity to collect vast amounts of data in a large scale manner. In order to handle the information overload, data mining techniques like online clustering are performed. A lot of online clusterers are based on the concept of micro-clusters in order to represent the given data stream. Based on its definition, micro-clusters can be represented as an n-sphere. Online clustering algorithms like BIRCH or DenStream use different strategies for maintaining the micro-clusters in evolving time series, but using the same underlying key concept storing a summarized version of the data stream in their models. We propose ViMEC, an application for multidimensional micro-cluster visualization, giving the user the opportunity to gain understanding of the internal behaviour of the clustering model. For a given time frame, ViMEC gives the user three different types of visualizations presenting different levels of details: Overview, Pair-view and Detail-view. These views combine not only a summary and detail representations for the different dimensions, but also aim to show different relations between dimensions. Preliminary results show, that large data sets with up to 20,000 data points can be visualized within less than 20 seconds.

CCS Concepts

•***Human-centered computing*** → *Information visualization; Visualization toolkits;*

1. Introduction

The rapid development of digitalization in the public and private sector creates the opportunity to collect large amounts of data. In order to cope with such an information overload, automatic data analytics is performed, providing insights by filtering and summarizing valuable information. However, valuable information might be hidden in the high-dimensional, complex structures within the data. Thus, analyses like clustering are performed grouping similar data and providing more structure. As large amounts of data need to be processed, iterative updateable approaches are applied. Such online clustering algorithms often use as key strategy micro-clusters in order to aggregate the incoming data stream into a summarized format called micro-cluster (e.g. BIRCH [ZRL96], BICO [FGS*13], CluStream [AR13], DenStream [CEQZ06], HDDStream [NZP*12]).

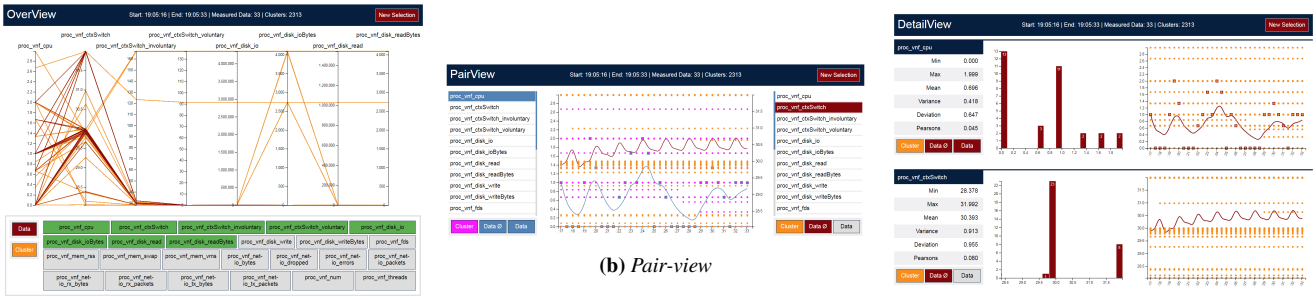
Micro-clusters consists of a tuple containing three entries: $\{N, L, S\}$, where N is the number of data points x_i, L the linear sum $\sum_{i=1}^{N} x_i$ of data points and S the squared sum $\sum_{i=1}^{N} x_i^2$. This notation makes it possible to maintain an aggregated summary of a large set of data points in an online manner by just adding new data points to the three components. Furthermore, clusters can be easily combined by adding the single components. Based on these three values, it is possible determining a centroid ($\frac{L}{N}$) and radius ($\sqrt{\frac{N \cdot \frac{L}{N}^2 + S - 2 \cdot \frac{L}{N}^2 \cdot L}{N}}$) of the micro-cluster representing an n-sphere.

In this paper, we propose an interactive visualization, called ViMEC, for micro-cluster based algorithms. Through ViMEC, the user can gain insights about changes within the clustering model depending on the underlying data stream. Further, we present how this visualization helps to select better configuration of algorithms, e.g. for BICO ,with the use case of anomaly detection for web-service monitoring. Besides finding better configurations, the user can gain deep understanding of the behaviour of the algorithms and their limitations. Based on this knowledge, the algorithms can be adapted or improved. This especially gives the opportunity to use this tool in both academic and productive environments, wherever time series data is analysed using micro-clustering based techniques (e.g. medical ECG anomaly detection [ASG*17], intrusion detection [LZ17], video analysis [JGE*17], etc.).

2. ViMEC

ViMEC is a web-application written in Javascript using D3js [BOH11]. The user has to provide two types of input data as CSV-files. One represents the data stream, which assumes numerical values (e.g. monitoring values like CPU, memory-usage, etc.). The second represents the pretrained micro-clusters over time given by the tuple information (e.g. while training BICO over time). Given both sets of data, ViMEC first summarizes the data sets as they might be too large to be visualized at once. Therefore, the user can select a time period (e.g. training phase), while ViMEC gives a prediction whether the plots can be generated within 2 seconds to ensure an interactive behaviour based on historic loading times. Thus, the user is warned and may select a smaller amount of values.

DOI: 10.2312/eurp.20181122



(a) *Overview* **(b)** *Pair-view* **(c)** *Detail-view*

Figure 1: *Representation of ViMECs three different visualization modes.*

For anomaly detection, an administrator may select a training set as time frame, where the service behaves normally. Such that a set of micro-clusters is trained using e.g. BICO representing the normal behaviour. Data points not fitting into any micro-cluster are stated as anomaly. Concrete questions, which ViMEC addresses are:

- Does the micro-clusters cover the behaviour of the data stream?
- How does the set of micro-clusters behave over time? When do new clusters appear or disappear? Does specific metrics influence the creation of new clusters?
- Which metrics differentiate the several micro-clusters most?
- How do algorithmic parameters (like number of clusters within BICO) influence the appearance of micro-clusters?

Thus, ViMEC provides the user the following visualizations for a preferred time frame:

Overview: This part illustrates a summary for multiple dimensions at the same time (Figure 1a). A parallel coordinates diagram is used showing both cluster centroids and the data points aggregated from the selected time frame. Micro-clusters are also represented in its cluster size, described by the radius, provided by larger bullets on the parallel coordinates. Thus, the user can get first impressions whether the trained model reflects the current data stream or not. Additionally, the visualization can show in which dimensions (e.g.monitoring metrics) normal clusters group, which might be helpful to see if the model is overfitted. Furthermore, the system processes the data first and automatically decides to show only dimensions, which change over the given time period. Of cause, the user can select any set of dimension at the bottom. Also, the user can chose whether illustrating micro-clusters, time series data points or both.

Pair-view: The Overview does not contain details about the behaviour over time. Therefore, we introduce the Pair-view providing a visualization to show both micro-clusters and data over time (Figure 1b). On the right and left side of the plot, dimensions of interest can be selected. By choosing those, the plot adapts to the new projection. On the x-axis, the time frame is shown, while the characteristics of dimensions are represented on the y-axis. Below the dimension selection, the user can decide to show the micro-clusters, the data points from the time series and an exponential moving average smoothing of the time series presented as curve. Also, those buttons function as legend as they are created in the same colour like presented in the plot. Pair-view is useful for comparing the influence of two recorded dimensions influencing the behaviour of the micro-clusters. For example, the variables network-io and CPU usage may correlate for a certain webservice. When more requests are sent to the webservice, the CPU usage might increase due to computation. These dependencies can be nicely investigated, while exploring corresponding micro-clusters. Since algorithms like CluStream adapts the set of clusters over time, micro-clusters may appear or disappear, which is also visualized. Overall, Pair-view covers a lot of details and includes the possibility to investigate dependencies of two dimensions over time.

Detail-view: Figure 1c illustrates an example of the Detail-view. Administrators may request further information about the current data stream and the micro-clusters like statistical information. These are important to verify whether representative values are used in the training phase for example. Each individual dimension is therefore represented by a left part, containing statistical information (values like min, max, mean, standard deviation, Pearson correlation between centroids and the time series data and a histogram of time series values) and a right part containing the time based behaviour. The second part is illustrated by a similar plot as the Pair-view, just for a single dimension. Through the second part, the user can directly verify next to the statistical values the behaviour over time of the data stream and clusters, without switching to the Pair-view.

We evaluated ViMEC by testing its time to visualize the three parts. This should give us a first impressions of limitations for the data set sizes, which the web application can cope with. For this purpose, we created eleven data sets with up to 20k time series points, with 24 dimensions. The online clustering algorithm BICO, configured with 200 micro-clusters, is applied by using the MOA library [BHKP10]. We selected the whole duration as the time frame to be visualized within ViMEC. The results indicate, that in average, the system scales linear in the number of time series points. Even the largest file with 20k data points needs a reasonable time of less than 20 seconds in average.

In future, we like to evaluate additional extensive benchmarks regarding further impact factors like dimensions or alternating number of micro-clusters and integrate the opportunity to directly stream in data for real-time micro-cluster behaviour analysis.

References

[AR13] AGGARWAL C. C., REDDY C. K.: *Data clustering: algorithms and applications*. CRC press, 2013. 1

[ASG*17] ACKER A., SCHMIDT F., GULENKO A., KIETZMANN R., KAO O.: Patient-individual morphological anomaly detection in multi-lead electrocardiography data streams. In *Big Data (Big Data), 2017 IEEE International Conference on* (2017), IEEE, pp. 3841–3846. 1

[BHKP10] BIFET A., HOLMES G., KIRKBY R., PFAHRINGER B.: Moa: Massive online analysis. *Journal of Machine Learning Research 11*, May (2010), 1601–1604. 2

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE transactions on visualization and computer graphics 17*, 12 (2011), 2301–2309. 1

[CEQZ06] CAO F., ESTERT M., QIAN W., ZHOU A.: Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining* (2006), SIAM, pp. 328–339. 1

[FGS*13] FICHTENBERGER H., GILLÉ M., SCHMIDT M., SCHWIEGELSHOHN C., SOHLER C.: Bico: Birch meets coresets for k-means clustering. In *European Symposium on Algorithms* (2013), Springer, pp. 481–492. 1

[JGE*17] JANSEN A., GEMMEKE J. F., ELLIS D. P., LIU X., LAWRENCE W., FREEDMAN D.: Large-scale audio event discovery in one million youtube videos. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on* (2017), IEEE, pp. 786–790. 1

[LZ17] LI S., ZHOU X.: An intrusion detection method based on damped window of data stream clustering. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2017 9th International Conference on* (2017), vol. 1, IEEE, pp. 211–214. 1

[NZP*12] NTOUTSI I., ZIMEK A., PALPANAS T., KRÖGER P., KRIEGEL H.-P.: Density-based projected clustering over high dimensional data streams. In *Proceedings of the 2012 SIAM International Conference on Data Mining* (2012), SIAM, pp. 987–998. 1

[ZRL96] ZHANG T., RAMAKRISHNAN R., LIVNY M.: Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record* (1996), vol. 25, ACM, pp. 103–114. 1