A Multi-view Clustering Approach for Analysis of Streaming Data

Vishnu Manasa Devagiri, Veselka Boeva, Shahrooz Abghari

A Multi-View Clustering Approach for Analysis of Streaming Data*

Vishnu Manasa Devagiri, Veselka Boeva, and Shahrooz Abghari

Blekinge Institute of Technology, Karlskrona, Sweden
{vmd,vbx,sab}@bth.se

Abstract. Data available today in smart monitoring applications such as smart buildings, machine health monitoring, smart healthcare, etc., is not centralized and usually supplied by a number of different devices (sensors, mobile devices and edge nodes). Due to which the data has a heterogeneous nature and provides different perspectives (views) about the studied phenomenon. This makes the monitoring task very challenging, requiring machine learning and data mining models that are not only able to continuously integrate and analyze multi-view streaming data, but also are capable of adapting to concept drift scenarios of newly arriving data. This study presents a multi-view clustering approach that can be applied for monitoring and analysis of streaming data scenarios. The approach allows for parallel monitoring of the individual view clustering models and mining view correlations in the integrated (global) clustering models. The global model built at each data chunk is a formal concept lattice generated by a formal context consisting of closed patterns representing the most typical correlations among the views. The proposed approach is evaluated on two different data sets. The obtained results demonstrate that it is suitable for modelling and monitoring multi-view streaming phenomena by providing means for continuous analysis and pattern mining.

Keywords: Multi-View Clustering · Multi-Instance Learning · Closed Patterns · Streaming data · Formal Concept Analysis.

1 Introduction

In recent years, the amount of data being generated in areas such as web, social media, IoT, and smart monitoring applications is increasing rapidly. Data generated in most of these areas is usually heterogeneous as the data is generally collected at different locations using variety of devices (e.g., mobile devices, edge nodes, sensors in IoT networks) and/or streaming in nature as new data is continuously produced. Another common factor of the data generated in streaming scenarios is its evolving nature. Change of data characteristics over a period of

* This work is funded in part of the research project "Scalable resource efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden.

time, known as concept drift, is an important challenge to be addressed when dealing with streaming data.

Clustering techniques are well-known tools and broadly used for analysis and extraction of interesting patterns from unlabeled data sets. Traditional clustering algorithms however, are not suitable and cannot deal with the data generated in today's smart monitoring applications due to characteristics already mentioned above like heterogeneity, streaming nature, concept drift [7]. There is a need for new clustering algorithms that are able to address these challenges. Data stream mining is an area dealing with the challenges concerning analysis and understanding of streaming data scenarios. Multi-view clustering, a distributed clustering technique, is capable of analysing heterogeneous data that are generated by different sources and represents different views or perspectives about the studied phenomenon. In multi-view clustering scenarios different views, contexts or interpretations of the data bringing complementary information (e.g., numerical reports of a patient and reports like ECG), are analysed in order to extract meaningful correlations among the different views. Although many research studies have been conducted and published in both data stream mining and multi-view clustering fields, the area of multi-view stream clustering is still in its infancy and there is a need for clustering techniques addressing and analysing streaming data in a multi-view fashion [13,18]. Some of the major challenges of multi-view stream clustering techniques are data heterogeneity [22], incomplete views [16,18,23] and evolving nature of the data [13].

In this work, we propose a multi-view clustering algorithm, entitled MV Multi-Instance Clustering, that can be used for monitoring and continuous analysis of streaming data scenarios. The proposed algorithm allows for parallel monitoring of the individual view clustering models and analyzing the views' correlations revealed by the integrated (global) clustering model. The individual view clustering models at each data chunk are initially updated when new data arrives by applying multi-instance clustering. Then, a global model can be built at each data chunk as a formal concept lattice generated by a formal context. The latter consists of selected closed patterns presenting the most typical correlations among the different views. Such a hierarchical global model allows to analyse and compare the views' correlations derived by two consecutive data chunks. Note that the local models' data values are not needed in order to build the global model which supports data privacy and lowers the required memory for data processing. In addition, if there are missing data in some of the views the previously extracted correlations among the views could be used to reconstruct the missing values.

2 Related Work

Distributed clustering techniques can deal with large, unlabelled and heterogeneous data sets which cannot be gathered centrally [2,9,12,19]. Characteristics of distributed data like heterogeneity, scalability, security, etc., demand novel robust clustering algorithms to address these challenges [9]. While some

researchers [2] have tried to tackle various challenges in the field, others [9,12,19] have proposed an overview of the research being done. Gan et al. [9] discuss various challenges and provide a summary on the state-of-the-art distributed clustering techniques. The authors cover various important concepts in the field of data mining like frequent itemset mining, frequent sequence mining, frequent graph mining, clustering and privacy for distributed context. In [12,19], a comparative study on the various state-of-the-art distributed clustering techniques has been done. Bendechache and Kechadi [2] propose an algorithm, entitled Distributed Dynamic Clustering algorithm, which is based on k-means for spatial data that is distributed and heterogeneous.

Multi-view clustering deals with clustering techniques in which same data is available in different perspectives or views complementing each other [18]. Studies published in [7,22], provide an overview and analysis of different multi-view clustering techniques proposed. Fu et al. [7] evaluate the selected multi-view algorithms on seven real-world data sets using cluster validation metrics like accuracy, purity, and normalized mutual information. In [22], the authors have reviewed available multi-view clustering algorithms by grouping them into five categories. In series of papers [16,18,23], the authors address the challenges of incomplete views, where data in some views maybe missing. Shao et al. [18] develop an algorithm, entitled Online Multi-View clustering, based on non-negative matrix factorization for large scale incomplete distributed data sets.

It is interesting to note that in [14], the authors treat multi-view clustering as a multi-objective optimization problem. In [15], a multi-view clustering approach based on non-negative matrix factorization and probabilistic latent semantic analysis is proposed to obtain common consensus clustering across views. Research in [13,18] deals with streaming data in multi-view scenarios. Huang et al. [13] propose a novel multi-view clustering approach for streaming data.

In the current state-of-the-art algorithms for multi-view clustering there are not many solutions dealing with monitoring and analysis of streaming data and the challenges that come along with it. The proposed MV Multi-Instance Clustering algorithm address these challenges.

3 Background

3.1 Multi-Instance Clustering and Hausdorff Distance

Multi-Instance (MI) clustering is an unsupervised learning process, where the data objects are bags of instances and there is no information about the labels of bags [24]. This is a typical setting for many real world application scenarios in which, it is costly and even in many cases impossible to obtain labeled data.

Multi-Instance clustering algorithms are supposed to partition a set of unlabeled bags into a number of groups on the basis of a similarity measure. However, the task of distributing objects into clusters is more difficult in the multi-instance context, since the ambiguity due to the fact that the objects are bags of unlabeled often related instances. In this sense, the similarity measures used in

single-instance clustering may not be appropriate for multi-instance clustering scenarios. *Maximal Hausdorff distance* has been proposed in [5] to measure the distance between two bags and later successfully applied to the standard multi-instance learning problem [21]. However, in [24] the maximal Hausdorff distance has been found to not work well in the generalized multi-instance learning problems due to its sensitivity to outliers. Therefore, the authors have proposed another distance called *average Hausdorff distance*.

In this paper, we use average Hausdorff distance to measure the distance between two bags A and B, since the preliminary experiments with this distance have generated better results than the ones produced by the maximal Hausdorff distance. Formally, given two bags of data instances A and B, the *average Hausdorff distance* is defined by Eq. 1, where $dist(a, b)$ is the distance between instances $a \in A$ and $b \in B$, which usually takes the form of Euclidean distance, and $|\ .\ |$, represents the set cardinality.

$$H(A, B) = \frac{\sum_{a \in A} \min_{b \in B} dist(a, b) + \sum_{b \in B} \min_{a \in A} dist(a, b)}{|\ A\ | + |\ B\ |}. \tag{1}$$

3.2 Formal Concept Analysis

Formal Concept Analysis (FCA) [10] is a mathematical apparatus for deriving a concept hierarchy from a collection of objects and their properties. FCA allows to generate and visualize the concept hierarchies. FCA is used for data analysis, information retrieval, and knowledge discovery. In addition, it can be understood as conceptual clustering method, which clusters simultaneously objects and their descriptions.

FCA derives a concept lattice from a formal context constituted of a set of objects O, a set of attributes A, and a binary relation defined on the Cartesian product $O \times A$. The context is described as a table, the rows correspond to objects and the columns to attributes or properties and a cross in a table cell means that "an object possesses a property". The *concept lattice* is composed of formal concepts organized into a hierarchy by a partial ordering (a subsumption relation allowing to compare concepts). Intuitively, a concept is a pair (X, Y) where $X \subseteq O$, $Y \subseteq A$, and X is the maximal set of objects sharing the whole set of attributes in Y and vice-versa. Relying on the subsumption relation, the set of all concepts extracted from a context is organized within a complete lattice, which means that for any set of concepts there is a smallest super-concept and a largest sub-concept, called the *concept lattice*.

3.3 Closed Patterns

Sequential pattern mining is the problem of finding interesting frequent ordered patterns from a sequence database [1]. Given a sequence database $\mathcal{T}$ and a pattern α the support for α is the number of sequences in $\mathcal{T}$ that contain α as a sub-sequence. The pattern α is called frequent if its support is equal or greater than a user-specified support threshold. Mining frequent patterns in big

databases can lead to generating a large number of patterns. In order to mitigate this problem, one can only extract frequent closed sequential patterns. A pattern α is closed when none of its super patterns has the same support as α.

In this study, we apply BIDE [20], which is a famous frequent closed sequential pattern mining algorithm, to extract patterns. The Python implementation of BIDE is adopted from prefixspan library.

4 MV Multi-Instance Clustering using Closed Patterns

In [4], an extension of the Split-Merge Evolutionary Clustering algorithm (abbreviated Split-Merge Clustering) [3] for multi-view data streaming scenarios has been introduced. The introduced algorithm, MV Split-Merge Clustering, has been demonstrated to be able to integrate data from multiple views in a streaming manner. The algorithm can be applied for grouping distinct chunks of multi-view streaming data so that a global clustering model is built on each data chunk. Initially, an updated clustering solution (local model) is produced on each view of the current data chunk by applying the Split-Merge Clustering. In that way updated local models reflecting the information presented in the current and previous data chunks are obtained. FCA is then used in order to integrate information from the local clustering models and generate a global model that reveals the relationships among the local models.

We have recognized two main limitations of the MV Split-Merge Clustering [4]. First, the Split-Merge Clustering algorithm [3], used for updating the local clustering models, needs to find the cluster centroids in order to integrate the local models of two consecutive data chunks. Our proposed MV Multi-Instance Clustering algorithm overcomes this by interpreting the integration of two local models as a Multi-Instance clustering problem, i.e. each cluster (bag) is regarded as an atomic object. Evidently, by exploiting Multi-Instance clustering analysis, we enable to improve the performance of the algorithm and also handle the ambiguity which is typical for real-world streaming data. For example, we would be able to model semi-supervised learning scenarios where some bags may be labeled. Second, the MV Split-Merge Clustering [4] builds a global model by using all the identified correlation patterns among the views. This leads to the generation of a large and complex concept lattice that is not easy to be interpreted and analysed. In comparison, our MV Multi-Instance clustering algorithm uses closed patterns, which considers the most typical correlations among the views, to create a global clustering model. In this way, unimportant concepts are excluded and do not complicate the understanding and analysis of the built global model. In addition, there is an opportunity to obtain even a smaller set of the most frequent (top-ranked) patterns based on the frequency or support score associated with each closed pattern.

Let us formally describe our MV Multi-Instance Clustering algorithm. We consider a streaming scenario where a particular phenomenon (physical object, biological process, machine asset, patient etc.) is monitored under n different circumstances (views). We further assume that the data arrives over time in

chunks. Each chunk t can contain different number of data points and can be represented by a list of n different data matrices $D_t = \{D_{t1}, D_{t2}, \ldots, D_{tn}\}$, one per view. Each matrix D_{ti} $(i = 1, 2, \ldots, n)$ contains the information about the data points in the current chunk t with respect to the corresponding view i. Assume that chunk t contains N_t data points. In addition, n clustering models, one per view, can be built on each data chunk. Let $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tn}\}$ be a set of clustering solutions (local models), such that C_{ti} $(i = 1, 2, \ldots, n)$ represents the grouping of the data points in tth chunk with respect to ith view, i.e. a local model built on data set D_{ti}.

On each data chunk, the proposed algorithm conducts two main operations. They are described in Algorithms 2 and 3. The local models built on the current chunk C_t are first updated by analysing the newly arrived data D_{t+1}. Clustering solutions C_{t+1} are initially built on the new data chunk $t + 1$ and correlated with ones of chunk t in order to generate updated clustering models $C_t^{'}$ with respect to $t + 1$. Then, these local models $C_t^{'}$ are used to build a global model that consists of three parts providing information about different aspects of the studied phenomenon. Namely, the model includes the formal context, closed patterns and concept lattice. The latter two are generated based on the built formal context. The formal context F_t consists of the set of $(N_t + N_{t+1})$ data points, the set of K $(K = k_1 + k_2 + \ldots + k_n)$ clustering labels of $C_t^{'}$ and an indication of which data points are associated with which clusters. Thus the context is described as a matrix, with the data points corresponding to the rows and the cluster labels corresponding to the columns of the matrix, and a value 1 in cell (i, j) whenever data point i belongs to cluster $C_j^{'}$ $(j = 1, 2, \ldots, K)$. Evidently, the formal context F_t contains all view correlation patterns supported by the local clustering models. The set of closed patterns, denoted by F_t^c, contains the most typical correlations that exist among the views. Finally, the concept lattice provides description of the hierarchical organisation of the concepts it produces.

The operations for updating the local clustering models on data chunk t are given in Algorithm 1.

Algorithm 1: Use Bi-Correlation MI-Clustering to update the local clustering models on data chunk t

Input: local clustering models C_t and newly arrived data D_{t+1}
for *each view* i $(i = 1, 2, \ldots, n)$ **do**
  Build a clustering model $C_{(t+1)i}$
  Bi-Correlation MI-Clustering $(C_{ti}, C_{(t+1)i})$ (Algorithm 2)
end

Algorithm 2 describes Bi-Correlation MI-Clustering that is applied for updating the local clustering models on data chunk t. Average Hausdorff distance (see Section 3.1) is used to find the correlations between the two clustering solutions C_{ti} and $C_{(t+1)i}$ for each view i $(i = 1, 2, \ldots, n)$. Global threshold T_i (see Eq. 2) is calculated for each $\mid C_{ti} \mid \times \mid C_{(t+1)i} \mid$ adjacency matrix as follows:

$$T_i = \frac{\sum_{p \in C_{ti}} \min_{q \in C_{(t+1)i}} H(p,q) + \sum_{q \in C_{(t+1)i}} \min_{p \in C_{ti}} H(p,q)}{\mid C_{ti} \mid + \mid C_{(t+1)i} \mid}, \tag{2}$$

where $H(p,q)$ is the average Hausdorff distance (see Eq. 1) between a cluster $p \in C_{ti}$ and a cluster $q \in C_{(t+1)i}$. T_i averages the Hausdorff distances between each cluster in C_{ti} and its nearest cluster in $C_{(t+1)i}$ and vice-versa. Evidently, T_i measures the average Hausdorff distance between two clustering solutions.

Algorithm 2: Bi-Correlation MI-Clustering of C_{ti} and $C_{(t+1)i}$

Input: local clustering models C_{ti} and $C_{(t+1)i}$
Build a $\mid C_{ti} \mid \times \mid C_{(t+1)i} \mid$ adjacency matrix based on Hausdorff distance (Eq. 1)
Calculate global threshold T_i (Eq. 2)
Remove edges in the adjacency matrix for which $H(p,q) > T_i$
for *each uniformly random cluster* p *in* C_{ti} **do**
  Find average distance of adjacent nodes, denoted by T_i^p
  Remove edges in adjacency matrix for which $H(p,q) > T_i^p$
  Find neighbours of p in $C_{(t+1)i}$, denoted by $N^p_{(t+1)i}$
  Find neighbours of each $q \in N^p_{(t+1)i}$ in C_{ti}, denoted by N^q_{ti}
  Create cluster $C'_p = \{p\} \cup N^p_{(t+1)i} \cup_{q \in N^p_{(t+1)i}} N^q_{ti}$
  $C_{ti} = C_{ti} \setminus \{p\}$
end

The adjacency matrix can also be visualized as a bipartite graph to illustrate how the clusters are correlated. The nodes on the left side of the graph represent clustering solution of chunk t, i.e. C_{ti}, and those on the right hand side represents new clustering solution i.e. $C_{(t+1)i}$. T_i is used to filter out the edges between clusters which are far apart and thus avoiding considering too many clusters to decide which ones to merge. The average local distance T_i^p could be considered as the local threshold for each cluster p in C_{ti} and it is used to find its closest clusters in $C_{(t+1)i}$. The motivation of using T_i^p is that, it facilitates identifying new trends in the scenarios of concept drift, where a group of data points can form a new cluster by slowly moving away from their current cluster at each data chunk. By considering the average local distance as a merging condition, we avoid early merging which allows such new clusters to naturally form.

Algorithm 3: Use FCA and closed patterns to build a global model on data chunk t

Input: updated local clustering models $C_t^{'}$
Build a formal context, denoted by F_t. F_t is a $(N_t + N_{t+1}) \times K$ binary matrix that indicates for each data point belonging to $D_t \cup D_{t+1}$ which clusters of $C_t^{'}$ it is associated with
Derive closed patterns, denoted by F_t^c ($F_t^c \subset F_t$), from the set of all built patterns of F_t
Produce a formal concept lattice from F_t^c

5 Evaluation

5.1 Data Sets and Experimental Setup

Anthropometric data: Initial analysis is done on a comparatively small public data set [11] that describes the medical conditions of 399 undergraduate students based on their anthropometric data. Each student is described by the following features: age, obesity, body mass index (BMI), waist circumference (WC), hip circumference (HC), and waist hip ratio (WHR), Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), *preh* for women and *hyper* for men, where the *preh* and *hyper* are classification labels that show what kind of blood pressure the individual has (e.g., regular or hyper). In order to mimic the streaming data scenario required for the proposed algorithm, the data set is divided into historical and newly arriving data. The historical data set is composed of the 70% of total data and the remaining 30% is treated as the newly arriving data. The features of the data set are divided into three views, where view 1 (v_1) contains details about age and gender, view 2 (v_2) contains details about BMI, WC, HC, WHR, and view 3 (v_3) presents information about blood pressure (SBP, DBP). Initial grouping of data points in each view is given in Table 1.

Table 1. Cluster categories in the views of Anthropometric data set

Label	Cluster description	Size
v_{10}	Adolescence, male (age $<$ 20)	44
v_{11}	Adolescence, female (age $<$ 20)	63
v_{12}	Early adulthood, male (20 $\leq$ age $\leq$ 39)	124
v_{13}	Early adulthood, female (20 $\leq$ age $\leq$ 39)	157
v_{14}	Adulthood, male (age $>$ 39)	7
v_{15}	Adulthood, female (age $>$ 39)	4
v_{20}	underweight (BMI $\leq$ 18.49)	21
v_{21}	normal weight (18.50 $\leq$ BMI $\leq$ 24.99)	234
v_{22}	overweight (25.00 $\leq$ BMI $\leq$ 29.99)	113
v_{23}	obese (BMI $\geq$ 30.00)	31
v_{30}	Level 1 (SBP $<$ 120 and DBP $<$ 80)	141
v_{31}	Level 2 (120 $\leq$ SBP $\leq$ 129 and/or 80 $\leq$ DBP $\leq$ 84)	83
v_{32}	Level 3 (130 $\leq$ SBP $\leq$ 139 and/or 85 $\leq$ DBP $\leq$ 89)	67
v_{33}	Level 4 (140 $\leq$ SBP $\leq$ 159 and/or 90 $\leq$ DBP $\leq$ 99)	80
v_{34}	Level 5 (160 $\leq$ SBP $\leq$ 179 and/or 100 $\leq$ DBP $\leq$ 109)	23
v_{35}	Level 6 (SBP $\geq$ 180 and/or DBP $\geq$ 110)	5

Our objective is to use this data set to build controlled and easy to interpret experimental multi-view streaming scenarios for studying and comparing the two multi-view clustering algorithms described in Section 4. In this setup two experiments are conducted to evaluate the algorithms. Bi-Correlation MI-Clustering step of the proposed algorithm is initially compared with Split-Merge Cluster-

ing [4] for updating the local models. We also analyse how different views are related to each other using the closed patterns derived from the global model.

Real-world sensor data: The potential of the proposed approach is also demonstrated on a real-world data set from a company in the smart building domain. The data has been used in [6] for analysing and monitoring the control valve system behaviour. In smart building domain different types of metrics are collected from a wide range of sensors available for systems such as heating, ventilation, air conditioning, and refrigeration. Data covering a year period (Jan 1^{st} 2019 till Dec 27^{th} 2019) is used in the current study. The eight features listed in Table 2, seven of which also considered in [6], are used in our experiments.

Table 2. Features included in the real-world sensor data set

View	Id	Acronyms	Feature name	Units
Operation	1	SST	Secondary Supply Temperature	°C
	2	SRT	Secondary Return Temperature	°C
	3	PHL	Primary Heat Load	kW
Performance	4	VOM	Valve Openness Mean	%
	5	VOS	Valve Openness Standard Deviation	%
	6	SE	Sub-station Efficiency	%
Context	7	OTM	Outdoor Temperature Mean	°C
	8	OTS	Outdoor Temperature Standard Deviation	°C

The available data features are analysed and partitioned in three distinctive views: system operational behaviour parameters, performance indicators and contextual factors. The features SST, SRT, and PHL are selected to model the system typical operational behaviour. The system performance can be evaluated by these three indicators: VOM, VOS, and SE. Finally, the contextual factors are represented by the features: OTM and OTS.

5.2 Results and Discussion

Anthropometric data: This data can be used to study and associate different age categories with the patients' anthropometric measurements to identify patients with increased risk for cardiovascular disease, e.g., hypertension. The data set is used to generate 10 test data set couples by randomly separating the individual profiles into two sets, as it was explained in Section 5.1. Thus the first set (279 patients) of each couple presents the current data chunk of individual profiles, and the other one (120 individuals) is the new chunk of patients' profiles. In that way, we have created 10 test data set couples.

MV Split-Merge Clustering and MV Multi-Instance Clustering are applied and compared on the built 10 test data sets. For MV Multi-Instance Clustering, we have additionally studied and conducted the experiments with two different (maximal Hausdorff versus average Hausdorff) distance measures in order to

select the better one, i.e. we have done 20 experiments in total. The average Hausdorff distance has outperformed the maximal Hausdorff distance on all the 10 test data sets. This confirms the discovery in [24], hence we have chosen to use this distance measure in the definition of Algorithm 2 and discuss its experimental results further in this section.

Out of the ten experimental iterations of the proposed approach, we have selected the results of one of the iterations (same as the one in [4]) to be presented and discussed in detail further in this section. The lattice produced in this iteration by MV Split-Merge Clustering has a total of 160 non-empty concepts. Out of these, 82 concepts link clusters from all the three views. The lattice size generated by MV Multi-Instance Clustering on the built formal context is very similar, namely it has 165 non-empty concepts, out of which 83 concepts link clusters from all three views. In the considered iteration, the local models generated in the three views have $6, 5$ and 7 clusters, respectively. In view 1, the clusters presented in Table 1 are retained, i.e. the same six age categories. In view 2, the cluster presenting all individuals with obese weight (v_{23}) has been split into two different clusters and similarly with the individuals having blood pressure Level 5 (v_{34}) in view 3. It can be observed that most of the original clustering structure is retained with the proposed MV Multi-Instance Clustering.

Table 3. Closed patterns showing correlations between all three views (support 10)

Blood Pressure Level	S/N	Concept	Size	Blood Pressure
Level 1	1	v_{13}, v_{21}	53	Regular
(v_{30})	2	v_{11}, v_{21}	26	Regular
	3	v_{13}, v_{22}	15	Regular
	4	v_{12}, v_{21}	12	Regular
Level 2	5	v_{13}, v_{21}	22	Regular, Pre
(v_{31})	6	v_{12}, v_{21}	15	Regular
Level 3	7	v_{12}, v_{21}	18	Regular
(v_{32})	8	v_{12}, v_{22}	12	Regular
	9	v_{13}, v_{21}	10	Pre, 1 Regular
Level 4	10	v_{12}, v_{22}	18	Regular, Hyper
(v_{33})	11	v_{13}, v_{21}	15	Pre
	12	v_{12}, v_{21}	14	Regular, Hyper

We compare MV Split-Merge Clustering [4] and MV Multi-Instance Clustering algorithms with respect to the purity of the produced clustering solutions. For this purpose, we first consider how the four main classes (Regular male, Regular female, Hyper and Pre) are distributed among the clusters. The average value calculated on the ten conducted iterations of MV Split-Merge Clustering algorithm is 0.76. While the corresponding value generated by the proposed MV Multi-Instance Clustering is 0.895. We have also evaluated the two algorithms with the six blood pressure levels (Levels 1 to 6) as main classes, where the

score generated by the MV Multi-Instance Clustering is 1.0 versus 0.65 for the MV Split-Merge Clustering algorithm. The MV Multi-Instance Clustering has demonstrated a better performance in the both evaluation scenarios, i.e. it is able to detect more efficiently the correlations between the current and new incoming data chunks. We have further used the adjusted Rand Index [17] to determine the similarity between the partitions generated by MV Multi-Instance clustering algorithm and benchmark clustering (used in [4]) as a function of positive and negative agreements in pairwise cluster assignments. The average score produced by the proposed algorithm is 0.99 versus 0.44 for the MV Split-Merge Clustering.

We are interested in discovering the relationships among the views. Hence we have specially studied patterns with length 2 or 3 in order to reveal correlations that exist between two or three views. Closed patterns (see Section 3.3) have been used for this purpose. We have generated closed patterns with support 10 (patterns that cover $\approx$ 2.5% of the data set) which resulted in 43 concepts of which 12 patterns show the relationship between all the three views. Table 3 lists all 12 derived concepts from the retrieved closed patterns as examples to study the relations among the views. The generated closed patterns do not contain concepts where the blood pressure levels are either 5 or 6. This could be due to the less number of instances in these clusters which are 23 and 5, respectively (see Table 1).

It is interesting to notice that the first three top frequent patterns among the three views (see rows 1, 2 and 5 in Table 3) represent typical categories in female population: females of age between 20 and 39 (early adulthood) with Level 1 blood pressure and normal weight; females in the same blood pressure and weight group, but in adolescent age category (age less than 20); and females again in early adulthood and normal weight category, but Level 2 blood pressure. In comparison with these categories, the three least frequent concepts (see rows 4, 8 and 9 in Table 3) can also be considered. For example, rows 8 and 9 represent respectively, overweight males and normal weight females in early adulthood age category with Level 3 blood pressure. One can get further insight into the discussed concepts by analyzing frequent concepts that connect two views (not included in Table 3). For example, it can be observed that females in their early adulthood typically have normal weight. This is demonstrated by a concept with support 104. Females in this age group are less likely to be overweight (31 individuals) or underweight (11 individuals) since only small size concepts supports this. In addition, the individuals in this female age category are less likely to be obese as there is no concept with size above 10 to support this.

Real-world sensor data: This data can be used for modelling, understanding and monitoring the control valve system behaviour. For example, it would be useful if one can link or trace back certain performance to specific operational modes by taking into account the influence of contextual factors (e.g., outdoor temperature). Initially, the available data features are partitioned in three distinctive views as it is explained in Section 5.1. For each view averaged daily values of the corresponding features are calculated to build daily profiles. The

created daily profiles (361 in total) are then split into two parts in order to simulate two data chunks: the initial one with 243 daily profiles (January - August) used to build the system behaviour model and the new data chunk used for the model update contains 118 daily profiles (September - December).

Table 4. Closed patterns correlating all three views after the new data chunk is added

S/N	PHL	SST	SRT	VOM	VOS	SE	OTM	OTS	Month	Size
1	2.42	26.24	26.01	0.03	±0.06	58	21.34	±0.48	6, 7, 8	36
2	4.39	27.41	26.61	3.25	±1.13	68	17.77	±0.46	6, 7, 8	55
3	5.54	28.13	26.93	4.76	±1.16	76	16.13	±0.35	9	12
4	7.45	29.70	28.15	6.34	±0.99	77	15.45	±0.50	5	14
5	**3.00**	31.35	29.30	7.05	±1.09	86	13.48	±0.56	4	9
6	13.26	35.90	32.36	11.87	±0.65	87	10.65	±0.44	9	11
7	15.65	37.01	33.58	12.18	±0.41	92	9.48	±0.30	10, 11	13
8	16.74	37.81	33.61	12.85	±0.60	91	9.17	±0.49	5	12
9	**3.36**	41.61	35.33	14.99	±0.63	95	6.19	±0.44	3, 4	40
10	20.93	43.13	37.86	13.26	±0.45	95	5.30	±0.34	10, 11	32
11	20.75	43.68	38.36	13.23	±0.45	95	4.81	±0.30	12	16
12	37.45	47.36	38.29	17.37	±0.42	96	1.17	±0.40	3	11
13	42.54	48.20	38.49	18.04	±0.54	96	0.46	±0.33	1,2	54
									Total	315

Note. The unit for PHL is kW and for SST, SRT, OTM, and OTS is °C. VOM, VOS, and SE are expressed in %. For the full form of each feature see Table 2. Row enumerations in bold italic represent patterns repeated from the initial chunk. The bold in PHL column represents deviating behavior.

Initial clustering in views 1 and 2 is done by applying k-means. Silhouette index and elbow method are used to find the optimal number of initial clusters. In the initial data chunk, the optimal number of clusters in these two views are 3 and 4, respectively. In the second data chunk, the optimal number of clusters for view 1 is 4 while for view 2 is the same as in the initial chunk, i.e. 4. In view 3, the data points are grouped into 4 clusters according to the yearly seasons based on [8], i.e. the context view has the following four clusters: December to February (winter); March, April, October and November (early spring, late autumn); May and September (late spring, early autumn); June to August (summer). After applying MV Multi-Instance clustering algorithm to update local models the number of clusters in the three views are $5, 5$ and 6, respectively.

In order to analyse how the correlations among the views are updated, the global model built on the initial data chunk is compared with the one produced on the updated clustering solutions when the new data chunk is added. The lattice built on the initial local models generated 32 non-empty concepts and 14 concepts connecting all the three views. The new global model produced on the updated local models, after the new data chunk has arrived, contains 59 non-empty concepts from which 26 concepts connect all three views. We fur-

ther compare the sets of closed patterns produced by the corresponding formal contexts using one and the same support ($\approx$ 2.5% of the data set). The latter gives 18 concepts (support 6) connecting two or three views for initial data chunk and 32 concepts (support of 9), respectively on the second formal context. Table 4 lists all 13 concepts linking three views extracted after adding the new data chunk. Each concept is presented by its mean vector and additionally, the concepts are grouped into two groups with respect to the contextual view, i.e. average outdoor temperature above and below 10 °C. It is interesting to notice that 8 (rows 1, 2, 4, 5, 8, 9, 12 and 13) of these 13 concepts have been discovered by analysing the initial data chunk and they are the only discovered concepts with the same support linking the three views. By considering the concepts linking two views we observe that they are retained in the global model built on the new data chunk and are further expanded with data points from its. Evidently, the integration procedure of our algorithm demonstrates to have a stable behaviour with respect to discovered patterns. Five new patterns presented in the new data chunk have also been extracted, i.e. the proposed algorithm can be used as a continuous data mining technique. The newly discovered patterns may be labelled with the expected performance under a particular context. In addition, our results are comparable to the ones reported in [6], where 49 days in March and April have been marked as having deviating behaviour. Our algorithm presents those with two different concepts (5 and 9 in Table 4), both of which show sudden drop in PHL with respect to the other concepts in the same contextual group.

6 Conclusion and Future Work

In this study, we have proposed a novel multi-view clustering approach, entitled MV Multi-Instance Clustering, that uses average Hausdorff distance, closed patterns and Formal Concept Analysis for analysis of streaming data. The MV Multi-Instance Clustering allows for parallel monitoring of the individual view clustering models and analysing view correlations in the global model generated at each data chunk.

The proposed algorithm has been evaluated on two different data sets. In addition its performance has been benchmarked to MV Split-Merge Clustering. The MV Multi-Instance Clustering has outperformed the latter algorithm in the studied evaluation scenarios. In general, the obtained results have demonstrated that the proposed algorithm is a robust technique for modelling and continuous analysis and mining of streaming data.

The potential of the MV Multi-Instance Clustering has been demonstrated on real-world data from smart building domain. Our future aim is to pursue further evaluation and study whether the proposed approach is fit for other real-world distributed streaming scenarios.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th Int. Conf. on Data Engineering. pp. 3–14. IEEE (1995)
2. Bendechache, M., Kechadi, M.T.: Distributed clustering algorithm for spatial data mining. 2015 2nd IEEE ICSDM (2015)
3. Boeva, V., et al.: Bipartite split-merge evolutionary clustering. In: van den Herik, J., et al. (eds.) Agents and AI. pp. 204–223. Springer (2019)
4. Devagiri, V.M., Boeva, V., Tsiporkova, E.: Split-merge evolutionary clustering for multi-view streaming data. Procedia Computer Science **176**, 460 – 469 (2020)
5. Edgar, G.: Measure, Topology, and Fractal Geometry, 3rd. edn. Springer, Berlin (1995)
6. Eghbalian, A., et al.: Multi-view data mining approach for behaviour analysis of smart control valve. In: Proc. of 19th IEEE ICMLA. pp. 1238–1245 (2020)
7. Fu, L., Lin, P., Vasilakos, A.V., Wang, S.: An overview of recent multi-view clustering. Neurocomputing **402**, 148–161 (2020)
8. Gadd, H., Werner, S.: Heat load patterns in district heating substations. Applied Energy **108**, 176–183 (2013)
9. Gan, W., et al.: Data mining in distributed environment: a survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **7**(6) (2017)
10. Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis: Foundations and Applications. LNAI, no. 3626, Springer-Verlag (2005)
11. Golino, H.F., et al.: Predicting increased blood pressure using machine learning. Journal of Obesity (2014)
12. Hai, M., et al.: A survey of distributed clustering algorithms. In: 2012 Int. Conf. on Industrial Control and Electronics Engineering. pp. 1142–1145 (2012)
13. Huang, L., et al.: Mvstream: Multiview data stream clustering. IEEE Transactions on Neural Networks and Learning Systems **31**(9), 3482–3496 (2020)
14. Jiang, B., et al.: Evolutionary multi-objective optimization for multi-view clustering. In: 2016 IEEE CEC 2016. pp. 3308–3315 (2016)
15. Liu, J., et al.: Multi-view clustering via joint non-negative matrix factorization. In: Proceedings of the 2013 SIAM International Conference on Data Mining, SDM 2013. pp. 252–260 (2013)
16. Liu, X., et al.: Late fusion incomplete multi-view clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence **41**(10), 2410–2423 (2019)
17. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association **66**(336), 846–850 (1971)
18. Shao, W., et al.: Online multi-view clustering with incomplete views. In: 2016 IEEE Int. Conf. on Big Data (Big Data). pp. 1012–1017 (2016)
19. Singh, D., Gosain, A.: A comparative analysis of distributed clustering algorithms: A survey. In: 2013 Int. Symp. on Comp. and Business Intellig. pp. 165–169 (2013)
20. Wang, J., Han, J.: BIDE: efficient mining of frequent closed sequences. In: Proceedings of the 20th International Conference on Data Engineering. pp. 79–90 (2004)
21. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: a lazy learning approach. In: Proc. of the 17th ICML. p. 1119–1125 (2000)
22. Yang, Y., Wang, H.: Multi-view clustering: A survey. Big Data Mining and Analytics **1**(2), 83–107 (2018)
23. Ye, Y., et al.: Incomplete multiview clustering via late fusion. Computational Intelligence and Neuroscience pp. 1–11 (2018)
24. Zhang, M., Zhou, Z.: Multi-instance clustering with applications to multi-instance prediction. Applied Intelligence **31**, 47–68 (2009)