Event Streams Clustering Using Machine Learning Techniques

Hanen Bouali and Jalel Akaichi

Institut Supérieur de Gestion, BESTMOD, Tunisia

hanene.bouali@gmail.com, j.akaichi@gmail.com

DOI: 10.20470/jsi.v6i4.224

Abstract. *Data streams are usually of unbounded lengths which push users to consider only recent observations by focusing on a time window, and ignore past data. However, in many real world applications, past data must be taken in consideration to guarantee the efficiency, the performance of decision making and to handle data streams evolution over time. In order to build a selectively history to track the underlying event streams changes, we opt for the continuously data of the sliding window which increases the time window based on changes over historical data.*

In this paper, to have the ability to access to historical data without requiring any significant storage or multiple passes over the data. In this paper, we propose a new algorithm for clustering multiple data streams using incremental support vector machine and data representative points' technique. The algorithm uses a sliding window model for the most recent clustering results and data representative points to model the old data clustering results. Our experimental results on electromyography signal show a better clustering than other present in the literature.

Keywords: machine Learning, clustering, Incremental SVM, data representative points, Healthcare

1 Introduction

In recent years advances in hardware technology have allowed us to automatically record transactions and other pieces of information of everyday life at a rapid rate. Such processes generate huge amounts of online data which grow at an unlimited rate. These kinds of data are referred to data streams.

Hence, an event is an object in time. The event occurring in a data streams constitute an event stream and an event stream has the same characteristics as a data stream. It's continuous and only a window of the streams can be seen at a time. Eventually, an event stream is a collection of events that are collected from a data stream over a period of time.

The issue of management and analysis of event streams have been researched extensively in recent years because of its emergency, imminent and broad application. The classification has been widely studied in the data mining literature. In this paper we are interested to this problem. Event streams show special challenges not only because of the huge volume of online event streams but also because that the event stream may temporally shows new correlation between events.

Once archived, event streams are difficult to query efficiently because of their rich semantics and large volume, forcing applications to sacrifice either performance or accuracy in addition to ignore existing correlation between events.

To cope with those problems, we need to extract flexible and comprehensible knowledge from data of material behavior. To reach this objective, we opt for machine learning techniques. Machine learning techniques are less complicated to use than traditional statistical techniques and are more comprehensible to users. ML algorithms are described either supervised or unsupervised. The distinction is drawn from how the learner classifies data. In supervised algorithms, the classes are predetermined. These classes can be conceived of as a finite set, previously arrived at by a human. Unlike unsupervised, learning task is to develop classification labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming groups. These groups are termed clusters.

In practice, event streams are partially labeled, this is the reason of unsupervised learning techniques' uses in our method.

In this paper, we are interested in solving the task of predicting Bell's palsy behavior. To do that, there are several steps we need to follow in order to build model from the stream event which are represented by EMG signals captured in sensors located in facial nerve ending (glands and muscles).

First, we have to formulate the learning problem in the task context. Second, a learning technique is selected. This selection is based on the properties of the learning problem created after simplifying the task context. Third, we prepare input for machine learning technique. Fourth, operational parameters for the learning technique used are selected. Fifth, the results of models are analyzed. Most often, the quality of learning is evaluated based on the reduction of errors in arriving at correct solution.

In this paper, in addition to the literature review, we aim to present in details the steps presented before in order to improve data quality and show new correlation between EMG signals.

The remainder of this paper is organized as follows.

In part 2, we discuss clustering algorithm evolution ranging from stream to SPECluster while highlighting their objectives, features, complexity and limits. In part 3, we detailed machine learning steps and illustrates our solution. Part 4 introduces the data analysis in this study; the results are shown and discussed. Finally, an overview of our work and opportunities that has been opened up are provided in part 5.

2 Related Works

Scientists use classification system to help them make sense of the world around them. They use classification to organize information and objects. When things are stored into groups, it makes them easier to understand and it makes it easier to see the relationships between them.

A related development is the increased demand for mining useful information from data streams. Examples of such increased demand are summarization, similarity searches, classification and clustering of data streams.

Current clustering techniques can be broadly classified into several categories: partitioning methods: k-means (Wang, 2014), k-medoids (Lai, 2011), hierarchical methods BIRCH (Labroche, 2014) (Alam, 2014), density-based methods DBSCAN (Andrade, 2013, Jinag, 2011) and grid based methods CLIQUE (Zhang, 2011). However, these methods are designed only for static data sets and cannot be directly applied to data streams.

In 2002, researches on clustering data in one data streams have emerged, O'Callaghan et al. presented an algorithm called STREAM (O'Callaghan, 2002) that was based on k-means, which adopts divided and conquer technique to process buckets and obtain cluster.

Later on, one of the first data stream clustering methods to consider the history cluster information in real time was CluStream (Toshniwal, 2012) for clustering large and evolving data streams. The method has clear advantages over recent techniques which try to cluster the whole stream at one time rather than viewing the stream as a changing process over time. The CluStream model provides a wide variety of functionality in characterizing data stream clusters over different time horizons in an evolving environment. This is achieved through a careful division of labor between the online statistical data collection component and an online analytical component. Thus, the process provides considerable flexibility to an analyst in a real-time and changing environment. These goals were achieved by a careful design of the statistical storage process. The CluStream algorithm uses micro-cluster as an on-line statistical data collection. This process is independent of any user input such as the time horizon of the required granularity of the clustering process. The aim is to maintain statistics at sufficiently high level of temporal and spatial granularity so that it can be effectively used by the off-line components such as horizon specific macro clustering as well as evolution analysis.

In this algorithm, we need first to create the initial *q* micro-cluster using an off-line process. They store the first InitNumber points on disk and use a standard k-means clustering algorithm. Wherever a new data point X_{ik} arrives, the Micro Clusters are updated in order to reflect the changes. Each data point can be observed by an existing micro-cluster or putted in a cluster of its own.

The micro-clusters generated by the algorithm serve as an intermediate statistical representation which can be maintained in an efficient way even for a data stream of large volume. On the other hand, the macro-clustering process does not use the voluminous data stream, but the compactly stored summary statistics of the micro-clusters.

Therefore, it is not constrained by one-pass requirements. It is assumed, that as input to the algorithm, the user supplies the time-horizon h, and the number of higher level clusters k which he wishes to determine.

The high-dimensional case presents a special challenge to clustering algorithms even in the traditional domain of static data sets. This is because of the sparsity of the data in the high-dimensional case. In high-dimensional space, all pairs of points tend to be almost equidistant from one another. However, CluStream does not show a good quality with these types of data. To do that, authors in (Aggarwal, 2004) develop an algorithm which is a modification of CluStream for high dimensional projected stream clustering (Alzghoul, 2011) by continuous refinement of the set of projected dimensions and data points during the progression of the stream. The updating of the set of dimensions associated with each cluster is performed in such a way that the points and dimensions associated with each cluster can effectively evolve over time. In order to achieve this goal, we utilize a condensed representation of the statistics of the points inside the clusters. These condensed representations are chosen in such a way that they can be updated effectively in a fast data stream. At the same time, a sufficient amount of statistics is stored so that important measures about the cluster in a given projection can be quickly computed. HPStream offers two main advantages:

- HPStream introduces the concept of projected clustering to data streams. Since a lot of stream data is high-dimensional in nature, it is necessary to perform high quality high-dimensional clustering.
- HPStream can reach consistently high clustering quality due to its adaptability to the nature of real data set, where data shows its tight clustering behavior only at different subsets of dimension combinations. HPStream explores a linear update philosophy in projected clustering, achieving both high scalability and high clustering quality.
- HPStream employs data projection methods to reduce the dimensionality of the data streams to a sub set of dimensions that minimize the radius of cluster grouping.

As with CluStream, however, the underlying assumption remains that cluster in the projected space remains spherical in nature.

BIRCH is a well-known hierarchical clustering algorithm (Lühr, 2009) that incrementally updates summary cluster information for off-line analysis. Clusters suitable for classification are then extracted using the summary information via a second pass over the data.

Later on 2005, DucStream (Letrou 2013) was a grid based technique that seeks dense units or regions in a multidimensional space. Clusters are discovered by applying CLIQUE algorithm to regions that are considered to be dense. The algorithm adopts the change in data stream by disregarding regions whose density fades over times.

DBSCAN is also known for data streams clustering (Chen, 2012). Many versions of DBSCAN as the incremental version, the OPTICS Algorithm (Letrou, 2013), multi-density clustering techniques and DenStream (Alzghoul, 2011).

The DenStream algorithm extends DBSCAN by adopting the original density based connectivity search to micro-cluster approach. The micro-cluster allows DenStream to efficiently summarize the overall shape of cluster as the data evolves without requiring the complete set of points to be retained in memory.

None of the mentioned algorithm provides a means to selectively archive historical information (Chen, 2012)

To cope with this problem, authors in (Chen, 2012) present an incremental graph based clustering algorithm whose design was motivated by a need to extract and retain meaningful information data stream produced by applications such as large scale surveillance… to this end, the method they propose utilizes representative points to both incrementally cluster new data and to selectively retain important cluster information within a knowledge repository.

Generally, the algorithm requires expert assistance in the form of the number of partitions expected or the expected density of clusters. The aims of this work have been to develop an algorithm that can handle all types of clusters with minimal expert help and to provide a graph based description of changes and pattern observed in the stream in order to enable a detailed analysis of the acquired knowledge. In this paper, authors present RepStream, a sparce-graph-based stream clustering approach that employs representative cluster points to incrementally process incoming data. The graph-based description is used because it allows us to model the spatio-temporal relationships in a data stream more accurately than is possible via summary statistics; each cluster is defined by using two types of representative points: exemplar points that are used to capture the stable properties of the cluster and predictor points which are used to capture the evolving properties of the cluster. A critical aspect of this research has been to avoid having to re-discover previously learned patterns by maximizing the reuse of previously useful cluster information. For this reason, a repository of knowledge is used to capture the history of the relevant changes occurring in the clusters over time. The use of the repository offers two major benefits:

The algorithm can handle recurrent changes in the clusters more effectively by storing a concise representation of persistent and consistent cluster features. These feature assist in the classification of new data points belonging to historical cluster distributions within an evolving data stream.

The repository provides a concise knowledge collection that can be used to rebuild a cluster's overall shape and data distribution history.

Researchers also study the clustering of multiple and parallel data streams (Chen, 2012). In the problem of clustering multiple data streams, each data is an element to be considered; this approach must focuses on the similarity between the data streams. The existing algorithms are based on Euclidean distance between data records. Because similar trends in data streams cannot be described by their Euclidean Distance, these methods may discard important trend information that is contained in the data streams. Similarity measure such as Euclidean Distance or correlation coefficients cannot be helpful for revealing the log similarity between the data stream. Authors in this paper propose a clustering algorithm for multiple data stream based on an auto-regressive modeling technique (Hory 2012) to measure the correlation between data streams. This algorithm uses frequency spectrum to extract the essential features of the data streams. Each stream is represented as the sum of the spectral components and the correlation is measured component wise. The SPECluster proposed algorithm assume a discrete time stamp model, where the time stamp are labeled by positive integers. The algorithm consists of on-line and off-line processing components.

3 Background

Given an event stream X of time ordered events X= $\{x_1, x_2, \dots, x_n\}$, we wish to find groups of events sharing similar attributes. We define a cluster c to be a set of events c = $\{x_1, x_2, \dots, x_c\}$ where each event x_i is a vector x_i= $\{a_1, a_2, \dots, a_m\}$ of m attributes.

Let C be the set of clusters c = $\{c_1, c_2, \dots, c_c\}$.

An event type E of an instance x_i describes the essential feature associated with the event instance denoted by x_i. type.

Each event type is associated a set of attributes. Each attribute has a corresponding domain of possible values:

- Attribute modeling time
- Attribute modeling location
- Attribute modeling threshold (domain specific attribute)
- Host
- Source (muscle name)

Let's say we have an extracted field called muscle which refers to the event source. We can make this attribute useful by tagging each portion failure based on its muscle. So a muscle identified with a name and a threshold.

4 Machine Learning Technique

4.1 Problem Formulation

The first step is to formulate the learning problem in the task context. The task context is the prediction whether Bell's palsy will occur under certain conditions or not.

Let us now formalize the unsupervised learning method.

Given:

- A geo spatial temporal framework E consisting of event Stream e_{ij} ordered as an C*SE, where C is the set of clusters and SE is the sub- events and 1<= i <=C, 1<=j<=SE.
- A finite set of events attributes (labels) A= $\{a_1, a_2, \dots\ a_n\}$ where n is the total number of attributes.
- Training Data Set D=D_l U $D_{ul}$where D_l contains labeled training EMG signals and D_{ul} contains unlabeled training EMG signals.

Find:

- Estimate classification parameters (Distance)

Objectives:

- Minimize clustering error ratio.
- Maximize intra-cluster distance and minimize inter-cluster distance.
- Assign a class label to each signal.

Assumptions:

- A1: the size of the labeled training data set is smaller than the number of unlabeled training data set.
- A2: D_l and D_{ul} samples are generated by the same function
- A3: Clusters are unseparable
- A4: Feature vectors are independent and identically distributed, but features are highly correlated in feature.
- A5: Input data are a combination of continuous and discrete random variable. In case of multisource data (different muscles), we have to deal with both continuous and discrete random variables.

4.2 Learning Technique Selection

The selection of the learning technique is based on the properties of the learning problem created after simplifying task context.

The SVM developed by Vapnick in 1995 is an emerging machine learning technique to classify and do regression.

SVM is used for a wide variety of problems and it has already been successful in pattern recognition and bio-informatics. SVM can produce clustering function from a set of training data set. The choice of SVM is based on its advantages which are essentially:

- Cores flexibility: similitude measure
- Reasonable computational time: find a hyper plane which minimizes the error ratio.
- Support vector: parsimonious representation and easily interpretable.

In the domain application, the SVM steps (Rubinstein, 2010) consist of core construction, parameter adaptation, efficient implementation and results analysis. Otherwise, in application and to apply the SVM we have to deeply understand the problem generalization, the complexity and the parsimony. The basic procedure for applying SVM to a clustering model can be stated briefly as follows: Map the input vectors into a feature space, which is possible with a higher dimensions. The mapping is either linear or nonlinear. In our algorithm, we aim for a higher accuracy. The higher accuracy is achieved by minimizing the noise in data which explain the choice of the non-linear SVM. To meet changes of event streams which can grow continuously at a rapid rate, we opt for incremental SVM to support such events change and to dynamically update schema changes

4.2.1 Incremental non-linear SVM algorithm (INSVM)

INSVM is an incremental learning algorithm which can meet the requirement of online learning algorithm to update the existing clustering schemas.

Let us consider a training dataset T of N pairs(x_i, y_i) where i= 1,…, N

x_i € R is the input data,

y_i € {1,k} is the output class label.
The SVM classifier used for data classification is defined by

x_i € C_{k_i} ; k argmax f_i (x_i)

Each decision function f_i is expressed as:

f_i (x)= W_i^T ϕ (x) +b_i

Where ϕ (x) maps the original data x_i to a higher dimensional space to solve non-linear problems.

In multi class, the margin between classes i and j is $\frac{2}{|w_i - w_j|}$ where $|w_i - w_j|$ is the distance measure in order to get the largest intra-classes distance between any two different classes i and j, minimization of the sum of $|w_i - w_j|^2$ for all i,j = 1..k is computed.

Among different sub classes of multi-class SVM classifier, we adopt the One Against All Approach and takes advantages from the schema which determines the class label of x using the majority voting strategy for the rest-class.

The main idea of INSVM is to train a SVM with a partition of the dataset, reserve only the support vectors at each training step and create the training support vectors for the next step with those vectors (fig.1).

In (Syed, 1999) authors showed that the decision function of an SVM depends only on its support vectors, and achieve the same results as the whole dataset.

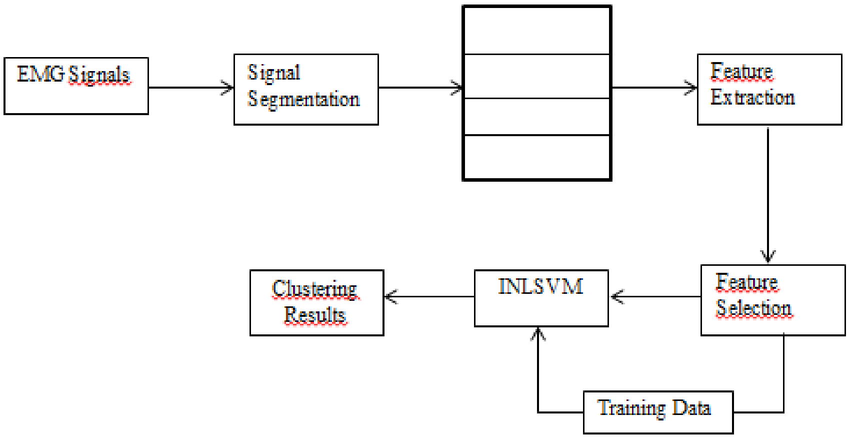

Fig. 1. INSVM Algorithm

4.3 Clustering Algorithm

We propose an algorithm for clustering incoming data stream based on events type (attributes). It divides the clustering process into on-line and off-line components. The algorithms assumes discrete time step model.

The algorithm adopts a pipelined online-offline processing (Figure2) method that supports a one-scan schema for detecting clusters. To do that we adopt the dynamic SVM so the algorithm can adoptively recognize the evolving behavior of data streams.

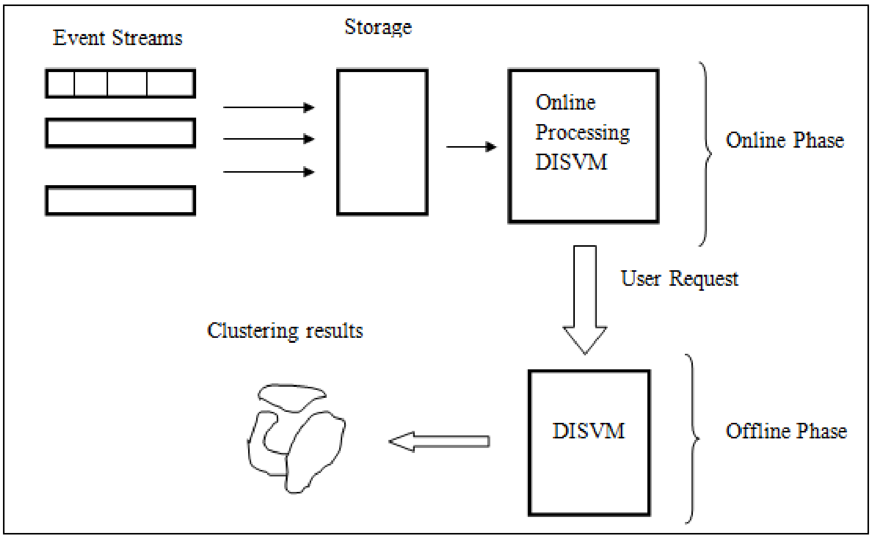

Fig. 2. Algorithm Processing

The adoption of a pipelined online and offline components in the clustering algorithms raise several important issues (Aggarwal, 2003)

- Maintain clusters at a sufficiently high level of temporal and spatial granularity.
- The summary clustering should provide sufficient temporal and spatial information for a horizon specific offline clustering process, which being prove to an efficient online update process.

4.3.1 Online Phase

The online process of the algorithm is not dependent on any user input such as the time horizon on the required granularity of the clustering process. We first create an initial clusters using Support Vector Machine. Whenever, a new data point X_i arrives, the INSVM updates clusters in order to reflect the changes. Each point can be observed by an existing cluster or putted in a cluster of its own.

***Clustering Multiple Data Streams under a Sliding Window*.**

Limitation imposed by available computational resources typically the ability to process continuously and huge amount of data. The above example (clustering of muscle signal) already suggests that one will usually not be interested in the entire data streams, which are potentially of unbounded length. Instead, it is reasonable to ensure that recent observations are more important than past data. Therefore, one often concentrates of a time window, that is subsequently of a continuously data streams.

But, to guarantee the efficiency and the performance of the algorithm in detecting changes over time, this total removal of previous data can affect it. Hence, if not completely remove; we would like to have the ability to access historical data without requiring any significant storage, processing or multiple passes over the data.

We aim in the part of online algorithm to build a selective history to track the underlying changes in the cluster observed.

In some real world applications such as sensor network, recent data are more important than old data. Hence, users are interested in the current data within a fixed time period; the sliding window also reduces memory requirements because only a small portion of data is stored. But, in our work and to detect changes over the time, in addition to the continuously data of the sliding window, old data must be taken into consideration. To do that, we store data representative points of previous sliding window.

Notation

Symbol	Meaning
X_i	The ith event stream
x_{ij}	The jth event of the ith stream
L	Sliding window length
l	Length of basic window
T	Time of the sliding window
M	Number of basic windows
N	Number of event stream(number of sensor network)

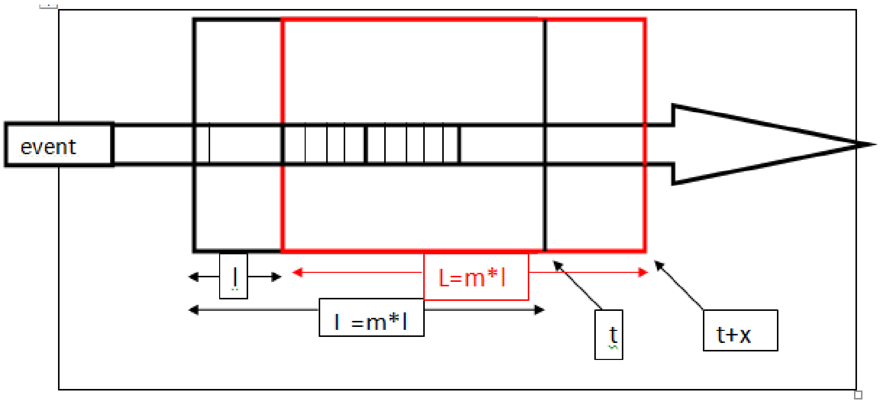

Fig. 3. A slide window of length L is divided into m basic window of size l

Given a length L of a slide window, at any tile t, we report the clustering results for the data stream in the time horizon [t-L,t]. To support efficient processing, we partition the slide window into m blocks of size l (m= L/l). This time segment is called basic window (Fig3).
Whenever a new segment (basic window) of length l arrives, the algorithm builds a new sliding window by inserting the new segment into the end and removing the old segment from the beginning.

The removal of the old segment does not means eliminates the basic window but keeping a data representative points of the basic window and such processes guarantee detecting changes in muscles' behavior over the time.

```
Algorithm Online (Storage)

Input: events from n streams$X_1$, $X_2$,…, $X_n$
Output: Slide window
Begin
1.   t=0
2.   while not the end of streams do
         a.   read $x_{ijt}$ from each $X_i$ ($x_1$, $x_2$,…, $x_j$), $X_2$,…, $X_n$;
                  i.   t= t+1;
                  ii.  if ( t mod l=0) then
                           1.   form a new basic window
                           2.   if ( the number of basic window = m) then
                                    a.   form a new slide window
                           3.   if (the number of basic window > m ) then
                                    a.   reduce the oldest basic window using data representative
                                         points
                           end if
                  end if
     end while
end.
```

Algo1. Online processing

At each time step, the online processing continuously reads a new data record from each event stream. This component consists of constituting the newest slide window. But to detect changes over the time, we need to store all data. To do that and before removing old data, we find the newest slide window and we summarize old events. This summarization is done through the technique of data representative points; this guarantees the storage of all events without requiring big space storage.

***Clustering using based representative points*.**

The representative points' technique retain only important event and incrementally store new data within a knowledge repository. The repository can then be subsequently used to assist in the processing of new data, the archival of critical features for the offline processing.

Two types of representative points exist:

- Exemplar points that are used to capture the stable properties of the stream event.
- Predictor points which are used to capture the evolving properties of the stream event.

This technique maximizes the reuse of previously useful events. For this reason, we store the representative points to handle recurrent changes in the event more efficiently. Also, when a recall of historical changes is desired. In some real world application, especially in our context, the medical decision, historical changes are crucial for some decision. The stream storage via representative points allows to rapidly adapting to significant pattern previously observed.

This algorithm tries to find representative cluster incrementally by intensively assigning new objects to partitions and re-computing representatives. Even though there are many approaches for clustering data that were proposed recently, representative based clustering is still widely used in many emerging application areas.

In representative based algorithms we met five sub-problems:

- Initialize representatives
- Measure distance
- Update representatives
- Evaluate clusters
- Stop criterion

The first question in connection with the clustering of active data streams concerns the concept of distance or, alternatively, similarity between streams. What does similarity of two streams means, and why should they fall into one cluster.

That is to say, two streams are considered similar if their evolution over time shows similar characteristics. As an example consider two muscles' signal both of which continuously increased compare to the threshold between 9.00 an and 11.00 am but then stated to decrease until 12.00 am.

The aim of this part is to extract common and interesting solutions for sub problems mentioned before and encapsulate them as a representative based clustering algorithm.

Input: Event from n Streams X_1,… X_n

Output: Clustering result

```
Begin
    t=0
    while not the end of streams do
    read $x_{ijt}$ from each $X_i$ ($x_1$, $x_2$,…, $x_j$), $X_2$,…, $X_n$;
    t=t+1;
    if (t mod l =0) then
    form a new basic window
    if (the number of basic window = m) then
     form a new slide window
    clustering representative points
    adjust clustering results
    output clustering result
    end if
```

```
13. end if
14. end while

End
```

At each time step, the online component of our proposed algorithm needs a new event from each event stream. In our case, event streams are parallel because of the existence of many body area networks.

If a new basic window is formed, it is stored in a stream engine until a new sliding window is formed. Once this latter is formed, the oldest slide is cluster using clustering representative points and we adjust clustering results with the previous clustering result in order to obtain one clustering results of all event stream as output. In line 10, employs procedure adjust to dynamically adjust clusters with the new cluster of the new slide window.

Clustering representative points.

The main idea of our algorithm, when clustering old slide window and when using clustering representative points, we obtain only two cluster which are:

- Exemplar cluster that are used to capture the stable properties of the stream event.
- Predictor cluster which are used to capture the evolving properties of the stream event.

This is a binary clustering which explain the reason of using Support Vector Machine (SVM) as clustering technique.

Algorithm. Clustering representative points

Input: slide window S(X_i, T) 1=<i<=n

Output: binary clustering results (exemplar cluster, predictor cluster)

```
Begin
    Read the event in the slide window and create two initial cluster
    t= $T_i$
    While not the end of the slide window do
    t=t+1;
    INSVM
    Output clustering results
    End while
End
```

Adjust Clustering Results Algorithm.

This algorithm consists of adjusting the clustering schema of the old slide window with the clustering schema of the new slide window in order to obtain one and only one clustering schema. In this part resides the difference between our algorithm and the SPE Cluster algorithm.

Algorithm. Adjust

Input: current clustering schema, new slide window

Output: updated clustering schema

```
Begin
    Compute support vectors for new slide window
    Set a new cluster with X as a center
    Incremental Non Linear SVM
    Recompute support vectors for the current clustering schemas
    While cluster center are closest to each other
        a. Merge the two cluster and compute new clusters center
    end While

End
```

4.3.2 Offline Phase.

In some real world application, old and recent data are both important. Therefore, sliding window is a model where mostly recent event may be important. To cope with this issue, an offline algorithm is needed to output clustering schemas of both recent and old data in order to have higher accurate model.

The offline phase is triggered when a user request is sent. To do that, a clustering schema is obtained as output. This schema obtained using INSVM.

In the offline phase, we read the whole event streams composed of the most recent slide window and data representative clusters with represent the old event. In line 3, algorithm INSVM is used to create the clustering results to recognize the evolving behavior of the data streams.

```
Algorithm offline
Input: Representative points and slide window
Output: clustering schemas
Begin
    1. Read representative points and the newest slide window
       t=0;
    2. While not the end of the streams
         t= t+1
         3. INSVM
    4. Output Clustering Results
    5. End while

End.
```

5 Experimental Evaluation

5.1 Testing Data

We tested our algorithm on a data set with data streams. This data set is for EMG signals recorded from different persons. To generate real-time and incremental data aspect, we made an algorithm that reads the data in a dynamic way. Each unit of time, the algorithms read a record which represents the EMG signal.

5.2 Clustering Quality

The quality of the proposed algorithm is measured by the correct rate and the ratio of the number of the streams that are correctly labelled to the total number.

Figure 4 shows the quality scores of the clustering results using different lengths of sliding windows. We can see that the quality of the clustering results improves with the increase in the length of the sliding window.

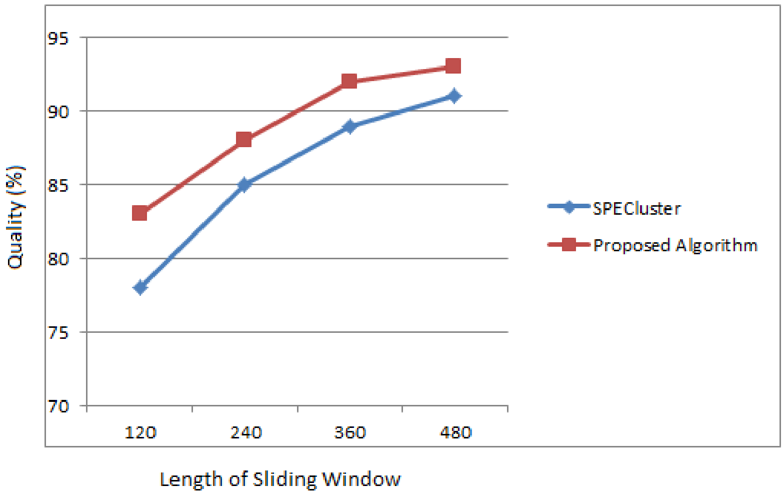

Fig. 4. Clustering Qualities under different length of sliding window

Then, to compare our algorithm to SPEcluster algorithm, we fix a sliding window and test the quality of the clustering results on our dataset. Figure 5 illustrates the quality of the clustering on the first 8 sliding window with size of 480. It shows higher quality of our algorithm due to the storage of old and new data. In addition to that, the clustering schemas are obtained by the adjust algorithm which adjust oldest clustering schema with the newest one.

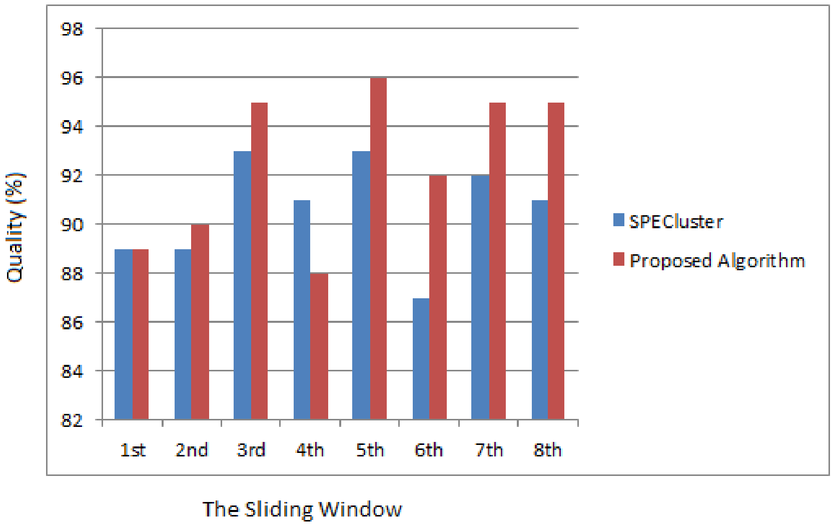

Figure 5. Clustering Quality in the sliding window

5.3 Temporal complexity

To test the processing speed of the proposed algorithm, we use the same data set to each SPEcluster and our algorithm. We set the sliding window to 480. The experimental results show that SPEcluster requires less executing time than our algorithm. Because the time of online processing is longer than offline processing, we list only computational time of the online processing in the algorithms. The difference between two algorithms for online process is minor. This is explained by the inclusion of old data for classification and the use of data representative points' technique to cluster old data. (Figure 6)

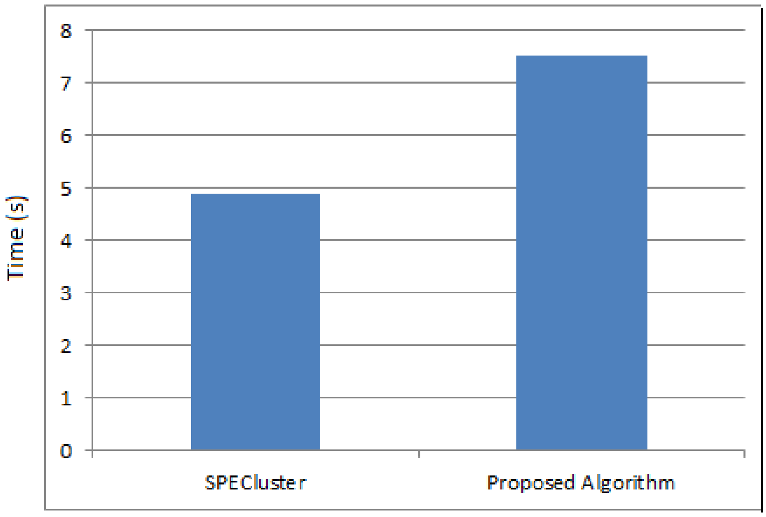

Fig. 6. Computational Time for SPEcluster and the proposed algorithm

6 Conclusion

In this paper, we propose an algorithm which adopts a pipelined on-line off-line processing method that supports a one scan schema for generating clustering schema and detecting changes over time. The algorithm takes in consideration old and recent data. To do that and minimize storage complexity, we use the points' representative clustering technique for clustering old data and for each new sliding window, the algorithm adjust the clustering schema with the oldest one. Experimental results on healthcare data set show that our algorithm has higher clustering quality comparing to SPECluster but needs more computational time due to the old data storage.

References

Aggarwal, C. C., Han, J., Wang, J., & Yu, P. S., 2004: A framework for projected clustering of high dimensional data streams. In *Proceedings of the Thirtieth international conference on Very large data bases, Vol. 30* , pp. 852-863

Aggarwal, C. C., Han, J., Wang, J., & Yu, P. S., 2003: A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases, Vol.29,* pp. 81-92 VLDB Endowment

Alam, S., Dobbie, G., Koh, Y. S., Riddle, P., & Ur Rehman, S., 2014: Research on particle swarm optimization based clustering: a systematic review of literature and techniques. *Swarm and Evolutionary Computation*

Alzghoul, A., & Löfstrand, M., 2011: Increasing availability of industrial systems through data stream mining. *Computers & Industrial Engineering*, vol.*60* (2), pp.195-205

Andrade, G., Ramos, G., Madeira, D., Sachetto, R., Ferreira, R., & Rocha, L., 2013: G-DBSCAN: A GPU Accelerated Algorithm for Density-based Clustering. *Procedia Computer Science*, vol.*18*, pp.369-378

Chen, L., Zou, L. J., & Tu, L., 2012: A clustering algorithm for multiple data streams based on spectral component similarity. *Information Sciences*, vol.*183* (1), pp.35-47

Greene, D., Cunningham, P., & Mayer, R., 2008: Unsupervised learning and clustering. In *Machine learning techniques for multimedia*, pp. 51-90. Springer Berlin Heidelberg

Hory, C., Bouillaut, L., & Aknin, P., 2012: Time–frequency characterization of rail corrugation under a combined auto-regressive and matched filter scheme.*Mechanical Systems and Signal Processing*, vol.*29*, pp.174-186

Jaehn, F., & Pesch, E., 2013: New bounds and constraint propagation techniques for the clique partitioning problem. *Discrete Applied Mathematics*, vol.161(13),pp. 2025-2037

Jiang, H., Li, J., Yi, S., Wang, X., & Hu, X., 2011: A new hybrid method based on partitioning-based DBSCAN and ant clustering. *Expert Systems with Applications*, vol. *38*(8),pp. 9373-9381

Labroche, N., 2014: Online fuzzy medoid based clustering algorithms. Neurocomputing, Vol. 126, pp. 141-150

Lai, P. S., & Fu, H. C., 2011: Variance enhanced K-medoid clustering. *Expert Systems with Applications*, vol. *38*(1), pp.764-775

Letrou, C., Khaikin, V., & Boag, A., 2012: Analysis of the RATAN-600 radiotelescope antenna with a multilevel Physical Optics algorithm. *Comptes Rendus Physique*, vol.13(1),pp. 38-45

Lühr, S., & Lazarescu, M., 2009: Incremental clustering of dynamic data streams using connectivity based representative points. *Data & Knowledge Engineering*, vol. *68*(1), pp.1-27

O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S., Motwani, R., 2002: Streaming data algorithms for high-quality clustering. In *ICDE '02 Proceedings of the 18th international conference on data engineering.* Pp. 685-694, Washington, DC: IEEE Computer Society

Rubinstein, R., Zibulevsky, M., & Elad, M., 2010: Double sparsity: Learning sparse dictionaries for sparse signal approximation. *Signal Processing, IEEE Transactions on*, vol.*58*(3),pp. 1553-1564

Sancho-Asensio, A., Navarro, J., Arrieta-Salinas, I., Armendáriz-Íñigo, J. E., Jiménez-Ruano, V., Zaballos, A., & Golobardes, E., 2014: Improving data partition schemes in Smart Grids via clustering data streams. *Expert Systems with Applications*, vol.41(13), pp.5832-5842

Syed, N. A., Liu, H., & Sung, K. K., 1999: Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 317-321

Toshniwal, D., 2012: A Framework for Outlier Detection in Evolving Data Streams by Weighting Attributes in Clustering. *Procedia Technology*, vol. *6*, pp.214-222

Wang, L., & Pan, C., 2014: Robust level set image segmentation via a local correntropy-based K-means clustering. *Pattern Recognition*, vol.47(5), pp 1917-1925

Zhang, H., & Liu, X., 2011: A CLIQUE algorithm using DNA computing techniques based on closed-circle DNA sequences. *Biosystems*, vol.105(1), pp.73-82

JEL Classification: C32, C38