# Data Stream Trajectory Analysis Using Sequential Possibilistic Gaussian Mixture Model

Wenlong Wu, James M. Keller, Marjorie Skubic
Electrical Engineering and Computer Science Department
University of Missouri-Columbia
Columbia, MO 65211, USA
ww6p9@mail.missouri.edu

Mihail Popescu
Health Management and Informatics Department
University of Missouri-Columbia
Columbia, MO 65211, USA

**Abstract—Data stream processing has gained much attention lately, in the era of big data. Streaming clustering is an effective tool to recognize normal baseline and to detect outliers in sequentially presented data. Perhaps more importantly would be the ability to predict that incoming data indicates movement towards a likely anomaly. In this paper, a Gaussian Mixture Model (GMM) is employed to represent different patterns in the data stream. The Sequential Possibilistic One-Means (SP1M) is used for initialization, and is incorporated into the GMM framework to recognize new mixture components in the data stream. The new proposed algorithm is called Sequential Possibilistic Gaussian Mixture Model (SPGMM). Furthermore, two methods of trajectory analysis, the "maximum typicality decline" and the "trend value measurement," are used together with SPGMM to detect early signs of pattern changes before unusual pattern data arrive in the stream. The proposed SPGMM is tested on synthetic and real-world datasets, and is shown to have excellent performance on predicting early signs of pattern changes in these sequential streams.**

## I. Introduction

Multifaceted user behaviors in online social networks, monitoring systems, and the internet of things (IoT) are captured by the rapid development of technology. Many petabytes of data are being generated daily awaiting processing. However, these data streams may overwhelm the storage capacity or need to be processed in a timely manner. Thus, data stream processing techniques are required to analyze data immediately as they arrive in the system with little or no storage requirements. Pattern recognition is often conducted to determine if structure exists so that some of the redundant data can be removed.

Mixture models have been successfully applied for modeling data distributions, and they have been employed in many applications such as clustering. The Gaussian mixture model (GMM) [1] is one of the most effective methods that has been used to approximate distributions and detect outliers in the data. The GMM is often used when the distribution of the data is known. However, the distribution may be unknown if we do not have enough background or prior knowledge of the data. Clustering algorithms can be run on the data to find the Gaussian mixtures in this case.

Fuzzy C-Means (FCM) [2] is an effective method to find cluster structures in data. FCM works well on data that has overlapping clusters but it is sensitive to outliers. Possibilistic C-Means (PCM) [3] was then introduced by abandoning the membership sum-to-one constraint in FCM and has been shown to be robust against outliers. In [4, 5], PCM and a variant, Automatic Merging Possibilistic C-Means (AMPCM) [6] are employed to initialize a Gaussian mixture model and re-constructed to continuously adapt to the new incoming streaming data. The AMPCM that uses all data points as initial cluster centers is not suitable for big data and is very sensitive to noise. In recent work, we showed that a meaningful result in PCM can be obtained even when $c = 1$ due to the formulation of the PCM. The Sequential Possibilistic One-Means (SP1M) [7, 8, 9] is a sequential version of PCM that iteratively hunts for one possibilistic cluster at a time, which is an effective mechanism to handle coincident clusters in PCM. In this paper, the SP1M algorithm is incorporated into GMM to initialize the data distributions and detect pattern changes in the data stream. The resulting algorithm is called the Sequential Possibilistic Gaussian Mixture Model (SPGMM).

The SPGMM is designed to identify pattern changes in the data stream. Our goal is, aside of identifying new structures in the data stream, to detect early signs of pattern changes so that we can make an early and timely response to these changes. The early detection of changes requires us to analyze data trajectories in the incoming stream to determine trends or pattern changing signs.

The rest of the paper is organized as follows. Section II briefly reviews GMM, FCM, PCM and SP1M. Section III introduces SPGMM. Two methods of trajectory analysis in SPGMM are discussed in this section. Section IV shows our results on synthetic and real-world datasets, and Section V summarizes our conclusions and future work.

## II. Preliminary theory

### *II.A Gaussian Mixture Model*

Given a $d$-dimensional vector $x$, a Gaussian mixture probability density function can be written as:

$$p(x) = \sum_{i=1}^{c} w_i p_i(x) \quad (1)$$

where $c$ represents the number of mixture components, the mixture weights $w_i$ satisfy $\sum_{i=1}^{c} w_i = 1$ and $w_i > 0$. Each component density $p_i(x), i = 1,2,\ldots,c$ is the probability density function of Gaussian distribution parameterized by a $d \times 1$ mean vector $u_i$ and a $d \times d$ covariance matrix $\Sigma_i$. The probability density function is shown as follows,

$$p_i(x) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} exp\{\frac{1}{2}(x - u_i)^T(\Sigma_i)^{-1}(x - u_i)\} \quad (2)$$

It has shown that any continuous probability density function can be approximated with high accuracy by using a sufficient number of Gaussian mixture components and by adjusting their means and covariance as well as the weights [10]. For video background subtraction, it is suggested in [11] to use a fixed number of Gaussian mixture components, usually between 3 and 5, to model the data. However, the number of Gaussian components in different situations should change to describe different data patterns. Too many or too few Gaussian components may decrease the generalization or the accuracy of the algorithm. Thus, determining a suitable number of Gaussian components is of vital importance to the mixture modeling. Clustering algorithms are often used to help determine the number of Gaussian components in the data.

*II.B. Fuzzy C-Means*

The Fuzzy C-Means (FCM) [2] is defined as the minimization of the following objective function:

$$J_{FCM}(U, V; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m ||v_i - x_k||^2 \quad (3)$$

with the constraints

$$\sum_{i=1}^{c} u_{ik} = 1 \quad (4)$$

for all $k = 1, \dots, n$. Here, the fuzzifier parameter $m \in (1, \infty)$. Optimization of the FCM model can be performed by randomly initializing $V$ and then alternatively updating $U$ and $V$ (alternating optimization) using the necessary conditions for extrema of $J_{FCM}$,

$$u_{ik} = (\sum_{j=1}^{c} (\frac{||v_i - x_k||^2}{||v_j - x_k||^2})^{\frac{1}{m-1}})^{-1} \quad (5)$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \quad (6)$$

until a suitable termination criterion holds, for example when successive estimates of $V$ change less than a threshold $\varepsilon$.

$$\max_{i=1,\dots,c} ||v_i - v_i'|| < \varepsilon \quad (7)$$

In this paper, we will always consider the Euclidean distance $||\cdot||$, the fuzziness index $m = 2$, and the threshold $\varepsilon = 0.01$. FCM may work well on many general datasets but cannot perform well on outliers and noise due to the constraint given by equation (4).

*II.C Possibilistic C-Means*

The Possibilistic C-Means (PCM) algorithm [3] is a generalization of the Fuzzy C-Means (FCM) algorithm, which abandons the membership sum-to-one constraint in FCM (equation (4)). The objective function of PCM is as follows,

$$J_{PCM}(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m ||v_i - x_k||^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - u_{ik})^m \quad (8)$$

The PCM task is to minimize (8) subject to

$$u_{ik} \in [0, 1] \text{ for all } i \text{ and } k$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \text{ for all } i$$

$$\max_i u_{ik} > 0 \text{ for all } k$$

The typicality update in the PCM is:

$$u_{ik} = \frac{1}{1 + (\frac{d_{ik}^2}{\eta_i})^{\frac{1}{m-1}}} \quad (9)$$

The parameter $\eta_i$ determines the distance at which the typicality degree equals to 0.5. As was suggested in [3], $\eta_i$ can be determined as follows:

$$\eta_i = P \frac{\sum_{k=1}^{N} u_{ik}^m d_{ik}^2}{\sum_{k=1}^{N} u_{ik}^m} \quad (10)$$

Usually, $P$ is set as one. The cluster center update is still same as the cluster center update in equation (6).

PCM performs well against outliers but it may yield coincident cluster centers. Some PCM research has been done to deal with the coincident cluster center problem such as AMPCM [6] and PFCM [12].

*II.D. Sequential Possibilistic One-Means*

In PCM, each cluster is independent of the others and, effectively, is found separately. Hence PCM can be computed for $c = 1$. The Sequential Possibilistic One-Means (SP1M) algorithm [7, 8, 9] was created to overcome the coincident cluster drawback by generating one cluster at a time using Possibilistic One-Mean (P1M) and stopping when all the "dense" regions are found.

The pseudocode for the latest version of SP1M with dynamic $\eta$ is shown in Fig. 1, where $X$ is the dataset, $c$ is the input for cluster number, $\varepsilon$ is the threshold, $K$ is defined to be the number of points whose maximum typicality is smaller than 0.5. Note that the (*) details of dynamic $\eta$ computation in Fig. 1 is discussed in [9].

```
Input X, c, ε
Initialize U, V as empty, i = 1
Do {
    Repeat <loop to find a suitable cluster>
        Pick v ∈ X with probabilities            (11)
        Repeat <loop to execute P1M>
            Compute η(i) dynamically              (*)
            Compute u(v, X)                       (9)
            Compute v(u, X)                       (6)
        Until termination                         (7)
    Until termination                             (12)
    Append u to U
    Append v to V
} While (i ++ < c && # (P1M) < K)
Output U, V
```

Fig. 1. SP1M pseudocode

In SP1M, the cluster centers are not initialized randomly. They are initialized from probabilities based on the typicalities of the previously found clusters. The initial cluster centers are picked from dataset *X* with probabilities

$$p(x_k) = \begin{cases} \frac{1}{n} & if\ i = 1 \\ 0 & if\ \max_{j=1,\ldots,i} u_{jk} > 0.5 \\ \frac{1-\max_{j=1,\ldots,i} u_{jk}}{n-\sum_{s=1}^{n} \max_{j=1,\ldots,i} u_{js}} & otherwise \end{cases} \quad (11)$$

When P1M has found a new cluster center *v*, *v* may be very close to one of the cluster centers in *V* found so far. We only consider *v* a new cluster center if it has a distance of at least $2\eta$ from each cluster center in *V*, that is, if

$$\min_{w \in V} ||v - w|| \geq 2\eta \quad (12)$$

If this condition does not hold, then *v* is discarded and P1M runs again to find a non-coincident cluster center. SP1M terminates when *c* non-coincident clusters have been found. In [9], a termination criterion was inserted to avoid the algorithm being trapped in an endless loop when no more new clusters can be found. Consequently, less than *c* non-coincident clusters may have been found in this way. Note that $\#(\text{P1M}) < K$ means on the $k^{th}$ execution of P1M (the process of searching for the $k^{th}$ cluster), if the times of abandoning coincident cluster is greater than a threshold *K*, the algorithm stops, where *K* is the number of points whose maximum typicality is smaller than 0.5. The points whose maximum typicality is larger than 0.5 are likely to be those points strongly identified in an already found cluster. The value of *K* decreases at each P1M run.

## III. SEQUENTIAL POSSIBILISTIC GAUSSIAN MIXTURE MODEL

In this section, the new Sequential Possibilistic Gaussian Mixture Model (SPGMM) is introduced. The first part (III.A) incorporates the SP1M into GMM and presents the basic model on the data stream. The second part (III.B) adds two methods of trajectory analysis on the data stream and strengthens the flexibility of SPGMM to detect the early signs of the pattern changes in the data stream. The last part (III.C) briefly summarizes the SPGMM algorithm.

### *III.A. Basic Sequential Possibilistic Gaussian Mixture Model*

The heart of the basic SPGMM follows the MUSC algorithm in [4, 5], but instead of using PCM and AMPCM to initialize the GMM in [4, 5], we use SP1M that can automatically determine the number of clusters, that is, the number of Gaussian mixture components. When a new streaming data point $x_t$ comes in at time *t*, the Mahalanobis distance is calculated to each Gaussian mixture and then the algorithm determines if this new incoming point belongs to any of mixture model by comparing it with a pre-defined threshold.

If the new streaming data belongs to one of the Gaussian components, that is, one of the clusters is "firing" due to the activation of the new streaming data point, then the algorithm will update the Gaussian mixture model by updating the mean and covariance of the "firing" Gaussian by

$$u_{new} = u_{old} + \frac{x_t - u_{old}}{|u_{new}|} \quad (13)$$

$$\Sigma_{new} = \frac{(|u_{new}|-1)*\Sigma_{old} + (x_t - u_{old})^T (x_t - u_{old})}{|u_{new}|} \quad (14)$$

where $|u_{new}|$ is the cardinality of the "firing" cluster, $u$ and $\Sigma$ are mean and covariance of that Gaussian.

Otherwise, the system will make an alert and the new streaming data point will be marked as an outlier. All the data points that are marked as outliers will be recorded in an anomaly list for further analysis. We note that the outlier points may not be all necessarily indicative of pattern changes. We also need to check the anomaly list to see if a new cluster forms due to the new pattern possibility. If enough data marked as outliers form a cluster, then we should remove them from the anomaly list and build a new Gaussian component. SP1M is run in the anomaly list to hunt for one cluster at a time and return an empty set if it fails to recognize a cluster. The pseudo code of the SPGMM is shown in the last part of this section.

### *III.B. Trajectory analysis*

One characteristic of the data stream is that it forms a time series making neighboring data points related to each other. So, we can analyze the trajectory of the data stream as a time series to see if it has trends or an indication of forming an outlier (important to issue a preemptive alert). In this part, two methods of trajectory analysis in the data stream are introduced. The first method is to check the maximum typicality continuous decline in the data stream. The other method is to check if the data stream has the trend of going towards the cluster boundary.

#### *(a.) Maximum typicality continuous decline measurement*

In possibilistic clustering, the typicality value measures how "typical" a particular data point is to each cluster. If a data point has a large typicality value (maximum is 1), it means that data point strongly belongs to that cluster. A low typicality value indicates a weak cluster belonging (minimum is 0).

The continuous decline of the maximum typicality values in a data stream suggests that the data stream gradually decreases its belonging to any cluster. This is a measure of trajectory analysis that can predict an early sign of pattern changes in the data stream, and can be used in an early warning system where an alert is triggered if consecutive declines of the maximum typicality value are detected. For example, if '*Few_Num*' consecutive declines of the maximum typicality are detected, a "weak warning" will be triggered; if '*Med_Num*' consecutive declines of the maximum typicality are detected, a "medium warning" will be issued; if '*High_Num*' consecutive declines of the maximum typicality are detected, a "strong warning" will be sent. For this paper, the value of '*Few_Num*', '*Med_Num*' and '*High_Num*' are set to 3, 5 and 7 accordingly. The decline threshold could be adjusted for different situations or applications.

### *(b.) Trend value measurement*

In the data stream, the neighboring data points in time are correlated with each other. The data point in the current timestamp $x_t$ could be influenced by $x_{t-1}$, $x_{t-2}$, …, $x_{t-s}$ where $s$ is a local window size. By looking at the past streaming data's values, we may determine if the current data point has a trend of going outside of the cluster. Two new vectors are defined below to compute the trend value. The first vector is computed from the current steaming data point and the past data stream in a window size $s$. This vector $vec_1$ is defined as

$$vec_1 = \frac{1}{s}\sum_{i=1}^{s}(x_{t-i+1} - x_{t-i}) \tag{15}$$

The second vector is computed from the current streaming data point and its closest cluster center. This vector is defined as

$$vec_2 = v_{close} - x_t \tag{16}$$

The trend value is decided by these two defined vectors and computed from the cosine function of these two vectors, as shown below,

$$cos\ \alpha = \frac{vec_1 \cdot vec_2}{|vec_1| * |vec_2|} \tag{17}$$

To illustrate the idea, an example is shown in Fig. 3. The big red dot is the closest cluster center. The local window size $s = 3$ in this example. The blue dots are other data points inside of the cluster.

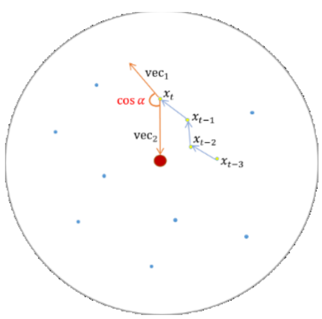



Fig. 2. The trend value illustration

The range of the trend value is between -1 and 1. As we can see above, the trend value $\cos\alpha$ is smaller than 0 in this example, which indicates the current time stamp value $x_t$ has a trend to go towards the boundary of the cluster. Thus, we can determine early signs of pattern changes by checking the symbol sign of the trend value. If the trend value $\cos\alpha$ is positive, $x_t$ has the trend to go towards the center of the cluster. Otherwise, $x_t$ has the trend to go towards the boundary of the cluster. The early signs of pattern changes usually happen when the trend value $\cos\alpha$ is negative, that is, when the current time stamp value $x_t$ has a trend to go towards the boundary of the cluster.

### *III.C. Final SPGMM Algorithm*

```
Input:
X - Existing dataset; T_d - distance threshold;
M - Minimum number of data points in one Gaussian mixture

Initialization:
Run Sequential Possibilistic One-Means (SP1M) on X;
Find all possible Gaussian mixtures (means and covariance) and anomaly list

Update:
For each data stream x_t:
        Calculate Mahalanobis distance to each Gaussian Mixtures and find the minimum
        If the minimal distance < T_d: <inside of the Gaussian mixtures>
                Incrementally update the mean and covariance of the "firing" Gaussian
                  -mixture using equations (13) and (14)
                Examine the anomaly list to check if any anomaly falls into the updated
                  -Gaussian mixtures and update them
                If the symbol sign of the trend value is negative:
                        Check the maximum typicality value continuous decline
                        If 'High_Num' continuous decline is detected:
                                Trigger "Strong warning"
                        Elf 'Med_Num' continuous decline is detected:
                                Trigger "Medium warning"
                        Elf 'Few_Num' continuous decline is detected:
                                Trigger "Weak warning"
                        Else:
                                "No warning"
                        End if
                End if
        Else: <outside of the Gaussian mixtures>
                Alert "This may be an outlier in the data stream!"
                Log x_t into anomaly list
                Examine the anomaly list for an emergent new behavior pattern
                  -by running SP1M on anomaly list
                If a new cluster is found && number of data points > M:
                        Update the Gaussian mixtures and delete data points of the
                          -new cluster from anomaly list
                End if
        End if
End for
```

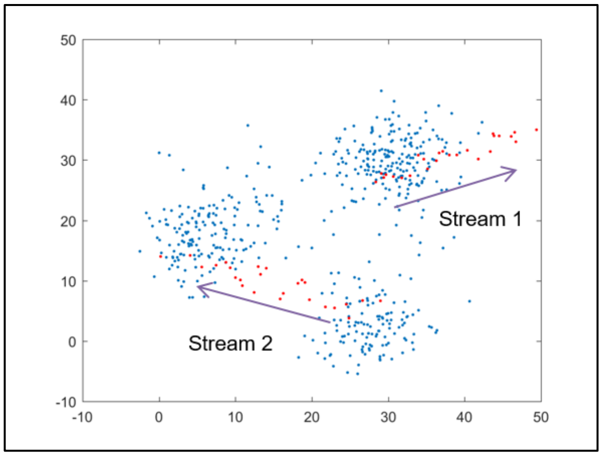
Fig. 3. SPGMM pseudo code

## IV. EXPERIMENTS

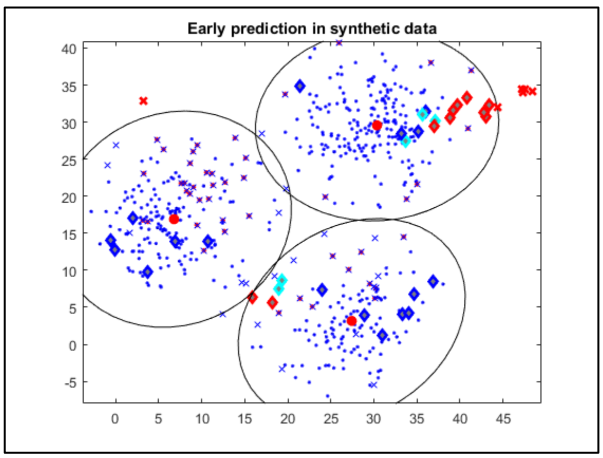
In this section, we used two datasets to test the SPGMM. The first dataset is a synthetic dataset that has three clusters, with vectors presented sequentially. Two trajectory paths were added in the data stream to test the algorithm. The second dataset is a real-world dataset, which is a collection of weather station nodes in the Le Genepi (LG) region in Switzerland [13]. We used two weeks of data at node 18 starting from October 10, 2007, to October 24, 2007. More details for each dataset are discussed in sections below. Our proposed SPGMM algorithm is tested on these datasets and has been shown to have excellent performance on predicting early signs of pattern changes in data streams.

### *IV.A. Experiment on the synthetic dataset*

In this experiment, we run the algorithm on a synthetic dataset that has three Gaussian clusters. Two trajectory paths were added in the data stream. The scatter plot of the full synthetic data is shown in Fig. 4 (a). One-third of the data is used to initialize the Gaussian mixture prototype. The top right cluster in Fig. 4 (a) is captured in initialization. The other two-thirds of the data is used as a data stream that is fed into the algorithm. The left cluster and bottom right cluster in Fig. 4 (a) are input into the system alternately. Stream 1 and stream 2 are two trajectory paths added in the data stream.




(a)



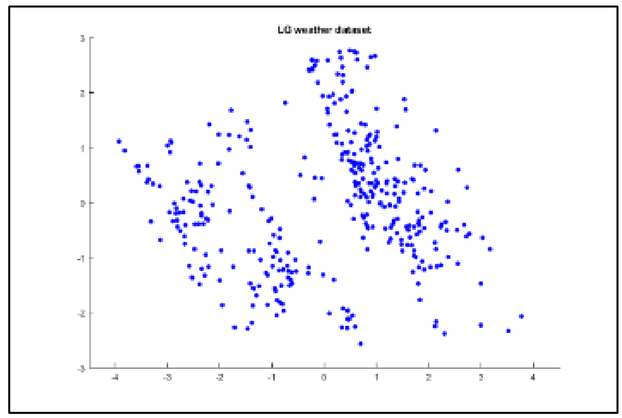

(b)

Fig. 4. (a) The synthetic dataset scatter plot and (b) the result of SPGMM where the red x denotes "outlier", the blue diamond denotes "weak warning", the cyan diamond denotes "medium warning", the red diamond denotes "strong warning" and the blue x denotes recovered normal point.

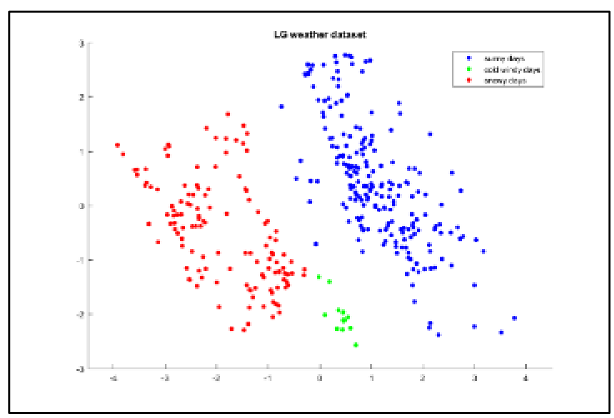
The SPGMM is run on this synthetic dataset. Three clusters were captured and the cluster centers are marked as red dots as shown in Fig. 4 (b). As we can see in the result, many warnings (marked with diamonds) were fired in both stream 1 and stream 2. For example, the data in stream 1 triggered the weak warning (blue diamond) at the beginning. As the data stream goes towards the outside of the cluster, the medium warning (cyan diamond) and the strong warning (red diamond) were triggered gradually. At the end, the data were marked as outliers (red x) because the data is "outside" of the cluster, which means this data point does not belong to any of the Gaussian components. For the trajectory in stream 2, the weak warning (blue diamond), the medium warning (cyan diamond) and the strong warning (red diamond) were triggered gradually. However the data stream becomes normal when it shifts to another cluster, which is a good indication since the data shifts to a different pattern instead of becoming an outlier. The SPGMM showed good performance in capturing the trajectory of moving towards the boundary of the cluster in this experiment.

*IV.B. Experiment on the LG weather dataset*

In August 2007, a small Sensor Scope network was deployed on the rock glacier located at 2500 meters on the top of Le Genepi above Martigny in Switzerland. A two week period starting from October 10, 2007, to October 24, 2007, was used in this experiment. Each data point's time gap is one hour and there are a total of 356 records in this dataset. This dataset contains five features: ambient temperature, surface temperature, relative humidity, rain meter and wind speed. To better visualize the clusters and see how the trajectory goes, Principal Component Analysis (PCA) was used to reduce the dimension of the dataset from $R^5$ to $R^2$. A scatter plot of reduced dimension LG weather dataset is shown in Fig. 5 (a).




(a)

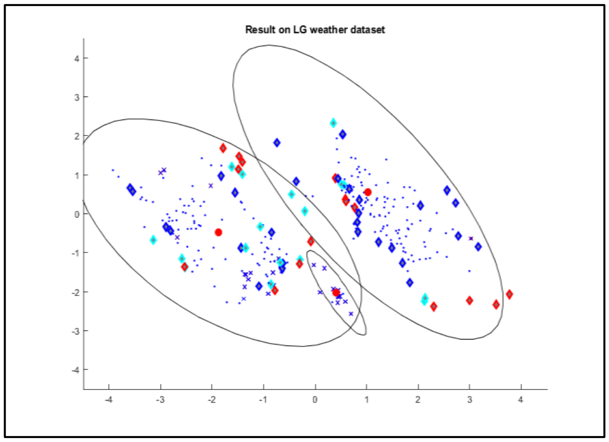



(b)

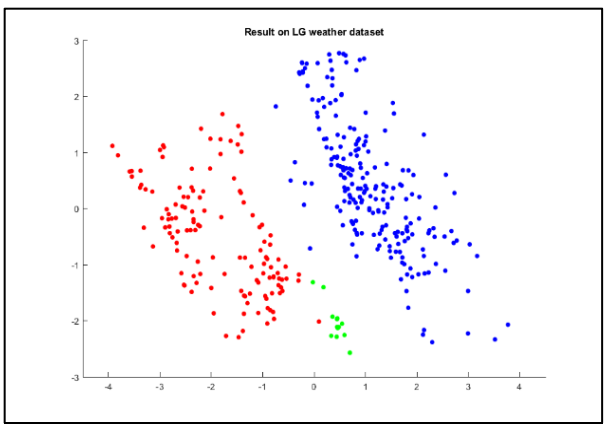
Fig. 5. The LG weather dataset scatter plot (a) of the original and (b) the labeled (Blue represents sunny days; green represents cold windy days; red represents snowy days)

By looking at the scatter plot, it does not provide clear visual evidence about how many clusters are in the LG dataset and where the clusters are located. Therefore, the imagery information from the site is used to show that there is a snowy day during the two weeks. A windy and cold day precedes the snow. So we consider the LG dataset to have three different types of weather: sunny days, cold windy days and snowy days as in the "ground truth" seen in Fig. 5 (b).

We run SPGMM on the LG weather dataset. One-fifth of the dataset is used to initialize the Gaussian mixture prototype. The other four-fifths of the dataset is used as a data stream that is fed into the algorithm. The final result of SPGMM on this dataset is given in Fig. 6 where Fig 6 (a) shows the warning system to predict the early signs of the pattern changes and (b) shows the final clustering results of each cluster.




(a)




(b)

Fig. 6. SPGMM results on the LG weather dataset: (a) warning system result where the blue diamond denotes "weak warning", the cyan diamond denotes "medium warning", the red diamond denotes "strong warning" and the blue x denotes recovered normal point; (b) the final clustering result

As we can see above, three types of warnings are still inside of the cluster but are detected to have trends that go towards the boundary of the cluster. There are many warnings that are not so close to the cluster boundary but they have trends of shifting out of the cluster in the data stream sequence. In Fig. 7, we analyze the trajectory path where the sunny days cluster shifts to the cold windy days cluster. On the current timestamp $x_t$, a strong warning (the red diamond in Fig. 7) is fired. Before $x_t$, weak warnings (the blue diamond in Fig. 7) and medium warnings (the cyan diamond in Fig. 7) are also fired. After $x_t$, the weather becomes cold and windy and shifts to the cold and windy days cluster, as created by SPGMM. Therefore, the strong warning at the current time stamp $x_t$ predicted the early sign of weather changes successfully.

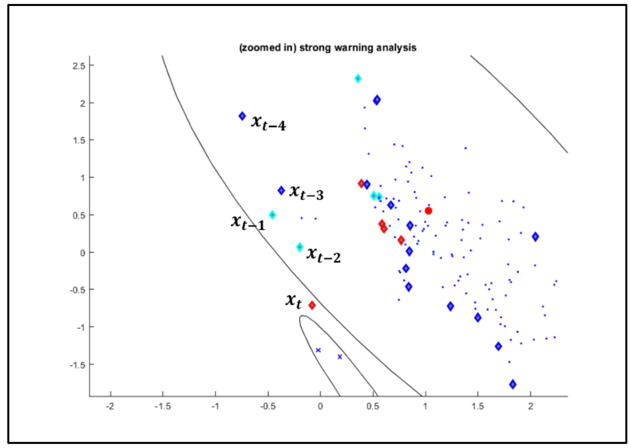



Fig. 7. The strong warning point analysis (zoomed in)

The successful prediction of the early sign of weather changes is due to the trajectory analysis that is introduced in the last section: the maximum typicality continuous decline measurement and the trend value measurement. The maximum typicality plot and the trend value plot of the data stream for this dataset are shown in Fig. 8. By looking at these two plots, we find that the maximum typicality value tends to decline continuously and the symbol sign of the trend value trends to be negative when one cluster stream shifts to the outliers or to other clusters.

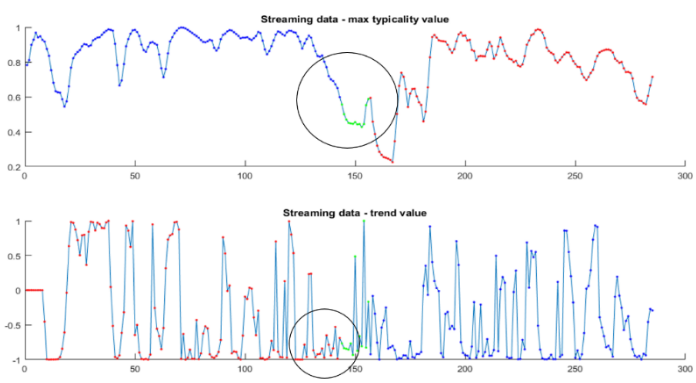



Fig. 8. The maximum typicality value plot and trend value plot of LG weather dataset

## V. Conclusions

Gaussian mixture model (GMM) is well known to have good performance for finding patterns in static data. However, it is hard to determine the number of Gaussian components when not enough information about the data is available. In this paper, we incorporated the Sequential Possibilistic One-Means (SP1M) together with the GMM to find patterns in the data and to detect outliers in the data stream. The resultant algorithm is called Sequential Possibilistic Gaussian Mixture Model (SPGMM). Each

Gaussian mixture adapts with every new incoming streaming data point. The new streaming data point will be marked as an outlier if it does not belong to any of the Gaussian mixtures. Furthermore, trajectory analysis is also conducted in SPGMM to detect the early signs of pattern changes in the data stream. The maximum typicality continuous decline measurement and the trend value measurement discussed in this paper are two effective methods to track the data stream's trajectory and are shown to have excellent performance on detecting the early signs of pattern changes in the data stream.

One issue in SPGMM is that it needs a proportion of data to initialize the Gaussian mixture components. This may be unrealistic sometimes if we want to deploy this method in a real-world application. In the future work, we are going to use fewer data points to do the initialization while striving to maintain the same accuracy and to decrease false alerts.

## VI. Acknowledgments

This work is supported by National Institutes of Health (NIH) under the project number R01NR016423. The authors would like to thank the continuing support of the MU Center for Eldercare and Rehabilitation Technology research team.

## References

[1] D. Reynolds, "Gaussian Mixture Models". *Encyclopedia of Biometrics*, pp.827-832, 2015.

[2] J. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm", *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984.

[3] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering", *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.

[4] O. A. Ibrahim, J. Shao, J. M. Keller and M. Popescu, "A temporal analysis system for early detection of health changes," *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 186-193, 2016.

[5] O. Ibrahim, Y. Du and J. Keller, "Robust On-Line Streaming Clustering", *Communications in Computer and Information Science*, pp. 467-478, 2018.

[6] M. Yang and C. Lai, "A Robust Automatic Merging Possibilistic Clustering Method", *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 26-41, 2011.

[7] T. Runkler and J. Keller, "Sequential possibilistic one-means clustering", *2017 IEEE International Conference on Fuzzy Systems*, pp. 1-6, 2017.

[8] T. Runkler, and J. Keller, "Sequential Possibilistic One–Means Clustering with Variable Eta," *27th Workshop on Computational Intelligence*, Dortmund, Germany, Dec. 2017 pp. 103-116, 2017.

[9] W. Wu, J. Keller and T. Runkler, "Sequential Possibilistic One-Means Clustering with Dynamic Eta", *IEEE International Conference on Fuzzy Systems*, pp. 1-8, 2018.

[10] C. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking", *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*.

[12] N. Pal, K. Pal, J. Keller and J. Bezdek, "A possibilistic fuzzy c-means clustering algorithm", IEEE Transactions on Fuzzy Systems, vol. 13, no. 4, pp. 517-530, 2005.

[13] Sensorscope. https://lcav.epfl.ch/page-145180-en.htm