Local Modes-based Free-Shape Data Partitioning

Plamen Angelov, *Fellow*, *IEEE* and Xiaowei Gu
School of Computing and Communications,
Lancaster University
Lancaster, LA1 4WA, UK
{p.angelov,x.gu3}@lancaster.ac.uk

***Abstract*—In this paper, a new data partitioning algorithm, named "local modes-based data partitioning", is proposed. This algorithm is entirely data-driven and free from any user input and *prior* assumptions. It automatically derives the modes of the empirically observed density of the data samples and results in forming parameter-free *data clouds*. The identified focal points resemble Voroni tessellations. The proposed algorithm has two versions, namely, offline and evolving. The two versions are both able to work separately and start "from scratch", they can also perform a hybrid. Numerical experiments demonstrate the validity of the proposed algorithm as a fully autonomous partitioning technique, and achieve better performance compared with alternative algorithms.**

Keywords— data partitioning; evolving clustering; parameter-free; data cloud; data- driven.

I. INTRODUCTION

Clustering has long been considered as one of the most effective tools for recognizing the underlying patterns within the data. As a supervised machine learning method, clustering currently is a very hot topic in the field of data analytics.

Established clustering algorithms [1-6] require different kinds of user inputs. For instance, the k-means algorithm [2] requires the number of clusters to be known beforehand, the DBScan algorithm [4] requires the radius and the minimum number of data samples within the radius to be predefined, etc. Although, these algorithms can achieve relatively high accuracy, their performances heavily rely on the user- and problem-specific parameters, which require clear *prior* knowledge. However, in most real cases, the *prior* knowledge is too limited for pre-defining the user input, which, in turn, influences the efficiency and accuracy of the clustering algorithms.

The concept of a *data cloud* was introduced in [7] as a collection of data samples entirely based on the mutual distribution and ensemble properties. Thus, *data clouds* are nonparametric and they do not have a specific shape. The *data clouds* directly represent the distribution of the observed data samples instead of giving some desirable/excepted (often, subjectively) pre-defined smooth functions. A number of focal points (not necessarily to be the centres or means of the *data clouds*) representing the modes of the data density. They attract the nearest data samples to them and then form *data clouds* forming Voroni tessellations [8] in the data space.

This work was supported by The Royal Society (Grant number IE141329/2014)

In this paper, we propose a new, fully autonomous algorithm named local modes-based data partitioning. This novel algorithm is entirely driven by the observed data samples and their mutual distribution in the data space. It can automatically identify the focal points representing the main modes of the data pattern merely based on the empirically observed data samples and then uses them to form *data clouds.* Thus, there is no need for any kind of user- or problem-specific parameters or assumptions.

This algorithm has two versions, *i)* offline and *ii)* evolving. The offline version is designed to partition the offline dataset while the evolving version is for streaming data processing. They have the ability of starting "from scratch", thus, they can both work independently. While a hybrid between the two versions is also possible. In this paper, we will introduce the two versions of the proposed method as two independent algorithms without loss of generality.

The remainder of this paper is organised as follows. Section II describes the theoretical basis of the proposed algorithm. The offline and evolving versions are separately introduced in sections III and IV. Section V summarizes main procedures of the proposed algorithm and section VI presents the numerical experiments and a discussion. The paper is concluded by section VII.

II. THEORETICAL BASIS

In this paper, the theoretical basis of the proposed local modes-based data partitioning algorithm will be introduced.

Frist of all, let us assume the data set/stream in the Hilbert data space $\mathbf{R}^d$ as $\{\boldsymbol{x}\}_k=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_k\}$, $\boldsymbol{x}_i=\left[x_{i,1},x_{i,2},...,x_{i,d}\right]^{\mathrm{T}}$, $i=1,2,...,k$ indicates the time instance that the i^{th} data sample arrives. To be more general and realistic, we assume that some data samples within the data set/stream repeat more than once, namely, $\exists \boldsymbol{x}_i=\boldsymbol{x}_j,\ \ i\neq j$. The set of unique data samples can be denoted as $\{\boldsymbol{u}\}_k=\{\boldsymbol{u}_1,\boldsymbol{u}_2,...,\boldsymbol{u}_{l_k}\}$ ($\boldsymbol{u}_i=\left[u_{i,1},u_{i,2},...,u_{i,d}\right]^{\mathrm{T}}$, $\{\boldsymbol{u}\}_k\subseteq\{\boldsymbol{x}\}_k$, $1<l_k\leq k$) and the corresponding frequencies of occurrence $\{f\}_k=\{f_1,f_2,...,f_{l_k}\}$ ($\sum_{i=1}^{l_k}f_i=1$).

In the rest of this section, the main operators of the recently introduced Empirical Data Analytics (EDA) [9]-[12] and their

corresponding recursive expressions disclosing the ensemble properties of the observed data samples will be introduced. At the end, the well-known *Chebyshev inequality* and its new, simpler form within the EDA framework will be briefly presented. The most widely used Euclidean type of distance is used in this paper for derivation clarity, but we have to stress that, the proposed algorithm can work with various types of distance as well.

A. Cumulative proximity

The *cumulative proximity* π of a particular data sample $\boldsymbol{x}_i$ ($i=1,2,...,k$) is defined as the sum of square distances between this data sample to all other data samples existing in the data space [9],[10]:

$$\pi_k\left(\boldsymbol{x}_i\right)=\sum_{j=1}^{k}\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2=k\left(\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_k\right\|^2+X_k-\left\|\boldsymbol{\mu}_k\right\|^2\right) \quad (1)$$

where $\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2=\sum_{l=1}^{d}\left(x_{i,l}-x_{j,l}\right)^2$. $\boldsymbol{\mu}_k$ is the mean of $\{\boldsymbol{x}\}_k$ and X_k is the average scalar product, and they can be updated recursively as:

$$\boldsymbol{\mu}_k=\frac{k-1}{k}\boldsymbol{\mu}_{k-1}+\frac{1}{k}\boldsymbol{x}_k;\quad \boldsymbol{\mu}_1=\boldsymbol{x}_1 \quad (2)$$

$$X_k=\frac{k-1}{k}X_{k-1}+\frac{1}{k}\left\|\boldsymbol{x}_k\right\|^2;\quad X_1=\left\|\boldsymbol{x}_1\right\|^2 \quad (3)$$

B. Standardized eccentricity

Standardized eccentricity is a very important measure for anomaly/fault detection [11]. The *standardized eccentricity* ε of $\boldsymbol{x}_i$ ($i=1,2,...,k$) is defined in [11] as:

$$\varepsilon_k\left(\boldsymbol{x}_i\right)=\frac{2\pi_k\left(\boldsymbol{x}_i\right)}{\frac{1}{k}\sum_{j=1}^{k}\pi_k\left(\boldsymbol{x}_j\right)}=\frac{2k\sum_{l=1}^{k}\left\|\boldsymbol{x}_i-\boldsymbol{x}_l\right\|^2}{\sum_{j=1}^{k}\sum_{l=1}^{k}\left\|\boldsymbol{x}_j-\boldsymbol{x}_l\right\|^2} \quad (4)$$

The sum of the *cumulative proximity* of $\{\boldsymbol{x}\}_k$ can be expressed using $\boldsymbol{\mu}_k$ and X_k instead [11],[12]:

$$\sum_{i=1}^{k}\pi_k\left(\boldsymbol{x}_i\right)=2k^2\left(X_k-\left\|\boldsymbol{\mu}_k\right\|^2\right) \quad (5)$$

Combining equations (1) and (5), the recursive expression of the *standardized eccentricity*, ε is expressed as follows:

$$\varepsilon_k\left(\boldsymbol{x}_i\right)=1+\frac{\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_k\right\|^2}{X_k-\left\|\boldsymbol{\mu}_k\right\|^2};\quad i=1,2,...,k \quad (6)$$

C. Unimodal density

Unimodal density [12] is defined as the inverse of *standardized eccentricity* ε. The *unimodal density* D of the data sample $\boldsymbol{x}_i$ ($i=1,2,...,k$) is expressed as follows [12]:

$$D_k\left(\boldsymbol{x}_i\right)=\varepsilon_k^{-1}\left(\boldsymbol{x}_i\right)=\frac{\sum_{j=1}^{k}\pi_k\left(\boldsymbol{x}_j\right)}{2k\pi_k\left(\boldsymbol{x}_i\right)}=\frac{1}{1+\frac{\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_k\right\|^2}{X_k-\left\|\boldsymbol{\mu}_k\right\|^2}} \quad (7)$$

D. Multimodal density

The *multimodal density*, D^{MM} of a specific unique data sample $\boldsymbol{u}_i$ ($i=1,2,...,l_k$) is a weighted sum of its *unimodal density* by the corresponding frequency f_i, expressed as [12]:

$$D_k^{MM}\left(\boldsymbol{u}_i\right)=\frac{f_i}{\sum_{j=1}^{l_k}f_j}\frac{\sum_{j=1}^{k}\pi_k\left(\boldsymbol{x}_j\right)}{2k\pi_k\left(\boldsymbol{u}_i\right)}=\frac{f_i}{1+\frac{\left\|\boldsymbol{u}_i-\boldsymbol{\mu}_k\right\|^2}{X_k-\left\|\boldsymbol{\mu}_k\right\|^2}} \quad (8)$$

E. Chebyshev inequality

The well-known *Chebyshev inequality* [13] describes the probability of the distance between a certain data sample $\boldsymbol{x}_i$ and the mean value to be larger than *n* times the standard deviation, namely, "$n\sigma$". We introduced earlier a more elegant and general form of *Chebyshev inequality* in terms of *standardized eccentricity*, ε, expressed as [11]:

$$P\left(\varepsilon_k\left(\boldsymbol{x}_i\right)\le 1+n^2\right)\ge 1-\frac{1}{n^2} \quad (9)$$

The *inequality* (9) can be applied to anomaly detection directly regardless of the distribution of data samples [11].

III. Offline Local Mode-Based Partitioning

The proposed offline version of the local modes-based partitioning algorithm employs the *multimodal density*, D^{MM} as the main operator. The main procedure of the offline algorithm is summarized as follows:

Stage 1: Identifying the global maximum

For every unique data sample within the dataset $\{\boldsymbol{x}\}_k$, its *local density*, $D_k^L\left(\boldsymbol{u}_i\right)$ ($i=1,2,...,l_k$) can be calculated using the following equation:

$$D_k^L\left(\boldsymbol{u}_i\right)=f_i\frac{\sum_{\left\|\boldsymbol{x}-\boldsymbol{u}_i\right\|^2<\bar{d}^2}\pi_k^L\left(\boldsymbol{x}\right)}{2k\pi_k^L\left(\boldsymbol{u}_i\right)} \quad (10)$$

where $\bar{d}$ is the average distance between the data samples of $\{\boldsymbol{x}\}_k$ and is derived from equation (5):

$$\bar{d}^2 = \frac{\sum_{i=1}^{k} \pi_k(\boldsymbol{x}_i)}{k^2} = 2\left(X_k - \|\boldsymbol{\mu}_k\|^2\right) \quad (11)$$

The unique data sample with the highest D^L is selected as the reference sample in the ranked unique data samples set $\{\boldsymbol{u}^*\}_k$:

$$\boldsymbol{u}^{*(1)} = \arg\max_{j=1,2,...,l_k}\left(D_k^L(\boldsymbol{u}_j)\right) \quad (12)$$

where $\boldsymbol{u}^{*(1)}$ is the unique data sample with the global maximum of the *local density* D^L and assign $\boldsymbol{u}^{*r} \leftarrow \boldsymbol{u}^{*(1)}$. If there are more than one maximum, choose any one of them to be $\boldsymbol{u}^{*r}$.

Stage 2: Re-ordering the local density

Then, we find the unique data sample which is nearest to $\boldsymbol{u}^{*r}$. Assuming there are q unique data samples holding the smallest distance to $\boldsymbol{u}^{*r}$ at the same time, they all will be selected and put into $\{\boldsymbol{u}^*\}_k$ together as $\boldsymbol{u}^{*(2)}$, $\boldsymbol{u}^{*(3)}$,…, $\boldsymbol{u}^{*(q+1)}$, and their order will be decided by their *local density* D^L in descending order. Once the unique data samples are put into $\{\boldsymbol{u}^*\}_k$, they will be removed from $\{\boldsymbol{u}\}_k$.

This process continues with the rest of the data samples remaining in $\{\boldsymbol{u}\}_k$, and the last unique data sample, $\boldsymbol{u}^{*(q+1)}$, in $\{\boldsymbol{u}^*\}_k$ is used as the reference data sample ($\boldsymbol{u}^{*r} \leftarrow \boldsymbol{u}^{*(q+1)}$) to find its closest unique data samples, and then put them into $\{\boldsymbol{u}^*\}_k$ and remove from $\{\boldsymbol{u}\}_k$. By repeating this procedure till $\{\boldsymbol{u}\}_k$ becomes $\boldsymbol{\phi}$, we can finally get the ranked unique data samples, denoted as $\{\boldsymbol{u}^*\}_k = \{\boldsymbol{u}^{*(1)} \mid i = 1,2,...,l_k\}$ and their corresponding ranked *local density* collection: $\{D_k^L(\boldsymbol{u}^*)\} = \{D_k^L(\boldsymbol{u}^{*(1)}), D_k^L(\boldsymbol{u}^{*(2)}),..., D_k^L(\boldsymbol{u}^{*(l_k)})\}$.

Stage 3: Detecting all local maxima

At this stage, we need to derive all local maxima of the ranked *local density* $\{D_k^L(\boldsymbol{u}^*)\}$ using the sign function:

$$\begin{aligned} &IF\left(\operatorname{sgn}\left(D_k^L(\boldsymbol{u}^{*(j-1)}) - D_k^L(\boldsymbol{u}^{*(j)})\right)\cdot \right. \\ &\qquad \left. \operatorname{sgn}\left(D_k^L(\boldsymbol{u}^{*(j)}) - D_k^L(\boldsymbol{u}^{*(j+1)})\right) = -1\right) \\ &AND\left(\operatorname{sgn}\left(D_k^L(\boldsymbol{u}^{*(j-1)}) - D_k^L(\boldsymbol{u}^{*(j)})\right) = -1\right) \\ &THEN\left(\boldsymbol{u}^{*(j)} \text{ is a local } \max\text{imum of } D^L\right) \end{aligned} \quad (13)$$

where $\operatorname{sgn}(\cdot)$ is the sign function: $\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$. We denote the set of the local maxima of D^L as the set $\{\boldsymbol{u}^{**}\}_k = \{\boldsymbol{u}^{**(j)} \mid j = 1,2,...,l_k^*\}$ ($l_k^* < l_k$).

Stage 4: Forming data clouds

Each local maxima, $\boldsymbol{u}^{**(i)}$ is then set as a focal point/prototype of a *data cloud*. All other data samples are then assigned to the nearest focal point (local maximum) forming *data clouds* according to the following rule:

$$cloud\ label = \arg\min_{j=1,2,...,l_k^*}\left(d\left(\boldsymbol{x}_i, \boldsymbol{u}^{**(j)}\right)\right) \quad (14)$$

After all the data samples within $\{\boldsymbol{x}\}_k$ are assigned to the *data clouds*, the actual center (mean) $\boldsymbol{\mu}_{k,j}$ and the standard deviation $\sigma_{k,j}$ ($j = 1,2,...,l_k^*$) of each data cloud can be calculated. Then we consider the set of centers, $\boldsymbol{\mu}_{k,j}$ ($j = 1,2,...,l_k^*$) of the *data clouds* as if this is our dataset and calculate their *multimodal density*, D^{MM} using equation (8). The respective frequency of each *data cloud* is determined based on the sum of frequencies of the unique data samples associated with it.

The centers $\boldsymbol{\mu}_{k,j}$ ($j = 1,2,...,l_k^*, l_k^* \le l_k$) are ranked again in the same way as described in stages 1 and 2, denoted as $\{\boldsymbol{\mu}^*\}_k$ and the corresponding standard deviations $\{\sigma^*\}_k$ as well. Then, the set of centers $\{\boldsymbol{\mu}^*\}_k$ is filtered according to the following rule:

$$\begin{aligned} &IF\left(d\left(\boldsymbol{\mu}_k^{*(j-1)}, \boldsymbol{\mu}_k^{*(j)}\right) > \left(2\left(\sigma_k^{*(j-1)} + \sigma_k^{*(j)}\right)\right)\right); \quad j = 1,2,...,l_k^* \\ &\quad THEN\left(\boldsymbol{\mu}_k^{*(j)} \text{ is deleted}\right) \end{aligned} \quad (15)$$

From the *Chebyshev inequality* (equation (5)) we can see that $P(\varepsilon_k(\boldsymbol{x}_i) \le 5) \ge \frac{3}{4}$, which means more than $\frac{3}{4}$ of the data samples are expected to be within 2σ from the mean.

After the filtering operation, the collection of the filtered *data cloud* centers denoted as $\{\boldsymbol{\mu}^{**}\}_k = \{\boldsymbol{\mu}_k^{**(j)} \mid j = 1,2,...,l_k^{**}\}$ ($l_k^{**} \le l_k^*$) is obtained, and they are passed to stage 4 for another round of filtering until the *data cloud* centers do not change any more.

Finally, we can get the filtering result, re-named as $\{\boldsymbol{\mu}^o\}_k$, and use the $\{\boldsymbol{\mu}^o\}_k$ as the focal points/prototypes to build *data clouds* using equation (14).

IV. Evolving Local Mode-Based Partitioning

The evolving version of the proposed local modes-based data partitioning algorithm works with the *eccentricity*, ε and *unimodal density*, D of the streaming data. The evolving algorithm has three stages as follows.

Stage 1: Identification of the evolving local modes

For each new data sample, denoted as $\boldsymbol{x}_{k+1}$, that arrives, the global $\boldsymbol{\mu}_k$ and X_k are updated to $\boldsymbol{\mu}_{k+1}$ and X_{k+1} firstly using equations (2) and (3).

After the *unimodal densities* of $\boldsymbol{x}_{k+1}$ and all the existing local modes, $D_{k+1}\left(\boldsymbol{x}_{k+1}\right)$ and $D_{k+1}\left(\boldsymbol{\mu}_{k,i}\right)$ ($i=1,2,...,C_k$) are obtained using equation (7), the following condition [14] is checked to see whether $\boldsymbol{x}_{k+1}$ is associated with a new local mode:

$$\begin{aligned} &IF\left(D_{k+1}\left(\boldsymbol{x}_{k+1}\right)>\max_{i=1}^{C_k} D_{k+1}\left(\boldsymbol{\mu}_{k,i}\right)\right)\\ &OR\left(D_{k+1}\left(\boldsymbol{x}_{k+1}\right)<\min_{i=1}^{C_k} D_{k+1}\left(\boldsymbol{\mu}_{k,i}\right)\right)\\ &THEN\left(C_{k+1}\leftarrow C_k+1\right)\end{aligned} \tag{16}$$

where we use C_k as the number of existing local modes at the k^{th} time instance.

If this condition is met, a new local mode is added in the data space ($C_{k+1}\leftarrow C_k+1$). On the contrary, if the condition is not satisfied, $\boldsymbol{x}_{k+1}$ is associated with the nearest existing local mode and $C_{k+1}\leftarrow C_k$.

Assuming $\boldsymbol{x}_{k+1}$ is assigned to the i^{th} ($i=1,2,...,C_{k+1}$) local mode which is decided by equation (14), the support of the i^{th} local mode is updated as $S_{k+1,i}\leftarrow S_{k,i}+1$; $\boldsymbol{\mu}_{k,i}$ and $X_{k,i}$ are updated to $\boldsymbol{\mu}_{k+1,i}$ and $X_{k+1,i}$ using equations (2) and (3). For other local modes, their parameters stay the same for the next processing cycle.

Stage 2: Filtering main local modes

Once there are no new data samples available, the *data clouds* will automatically be formed based on the existing local modes as focal points or poles of attraction around which to form *data clouds*. However, because *data clouds* do not have specific shapes, they may overlap with each other, thus, the redundant local modes need to be removed first.

The filtering stage begins from the *data cloud* with the smallest support and ends with the one with the largest support. For each *data cloud*, we check the following principle according to the *Chebyshev inequality* using the *standardized eccentricity* (equation (9)) [11]:

$$\begin{aligned} &IF\left(\varepsilon_{k,i}\left(\boldsymbol{\mu}_{k,j}\right)\le\varepsilon_o\right)OR\left(\varepsilon_{k,j}\left(\boldsymbol{\mu}_{k,i}\right)\le\varepsilon_o\right)\\ &THEN\left(Merge\ the\ i^{th}\ and\ j^{th}\ data\ clouds\ together\right)\end{aligned} \tag{17}$$

Here, we use $\varepsilon_o=5$, which corresponds to 2σ as described earlier. $\varepsilon_{k,i}\left(\boldsymbol{\mu}_{k,j}\right)$ and $\varepsilon_{k,j}\left(\boldsymbol{\mu}_{k,i}\right)$ are the *standardized eccentricities* calculated per *data cloud* expressed as follows ($i\neq j$) [12]:

$$\varepsilon_{k,p}\left(\boldsymbol{\mu}_{k,q}\right)=1+\frac{S_{k,p}^2\left\|\boldsymbol{\mu}_{k,q}-\boldsymbol{\mu}_{k,p}\right\|^2}{\left(S_{k,p}+1\right)\left(S_{k,p}X_{k,p}+\left\|\boldsymbol{\mu}_{k,q}\right\|^2\right)-\left\|\boldsymbol{\mu}_{k,q}+S_{k,p}\boldsymbol{\mu}_{k,p}\right\|^2} \tag{18}$$

where $p,q=i\ or\ j$ and $p\neq q$.

If the principle in equation (17) is met, the two *data clouds* merge:

$$\begin{cases} C_k\leftarrow C_k-1\\ S_{k,i+j}\leftarrow S_{k,i}+S_{k,i}\\ \boldsymbol{\mu}_{k,i+j}\leftarrow\frac{S_{k,i}}{S_{k,i+j}}\boldsymbol{\mu}_{k,i}+\frac{S_{k,j}}{S_{k,i+j}}\boldsymbol{\mu}_{k,j}\\ X_{k,i+j}\leftarrow\frac{S_{k,i}}{S_{k,i+j}}X_{k,i}+\frac{S_{k,j}}{S_{k,i+j}}X_{k,j}\end{cases} \tag{19}$$

Stage 3: Forming data clouds

After the main local modes were filtered out, the output is generated. The remaining local modes, re-named as $\left\{\boldsymbol{\mu}^o\right\}_k$, are used as the focal points/prototypes to build *data clouds* using equation (14).

V. Algorithm Summary

In this section, the overall procedure of the two versions (offline and evolving) of the proposed local modes-based partitioning algorithm are presented in the form of pseudo-code in two separate subsections.

A. *Offline local modes-based partitioning algorithm*

i. Calculate $D_k^L\left(\boldsymbol{u}_i\right)$ ($i=1,2,...,l_k$) using eq. (10);

ii. Find the unique data sample $\boldsymbol{u}^{*(1)}$ with global maximum of D^L using eq. (12);

iii. Send $\boldsymbol{u}^{*(1)}$ into $\left\{\boldsymbol{u}^*\right\}_k$ and $D_k^L\left(\boldsymbol{u}^{*(1)}\right)$ into $\left\{D_k^L\left(\boldsymbol{u}^*\right)\right\}$ and delete $\boldsymbol{u}^{*(1)}$ from $\left\{\boldsymbol{u}\right\}_k$;

iv. $\boldsymbol{u}^{*r}\leftarrow\boldsymbol{u}^{*(1)}$;

v. **While** $\left\{\boldsymbol{u}\right\}_k\neq\boldsymbol{\phi}$

* Find the unique data sample(s) which is/are nearest to $\boldsymbol{u}^{*r}$;

* Send the data sample(s) and the corresponding D^L to $\left\{\boldsymbol{u}^*\right\}_k$ and $\left\{D_k^L\left(\boldsymbol{u}^*\right)\right\}$, respectively;

* Delete these data sample(s) from $\left\{\boldsymbol{u}\right\}_k$;

* Set the latest element in $\{\boldsymbol{u}^*\}_k$ as $\boldsymbol{u}^{*r}$;

vi. **End While**

vii. Filter $\{\boldsymbol{u}^*\}_k$ and $\{D_k^L(\boldsymbol{u}^*)\}$ using eq. (13) and obtain $\{\boldsymbol{u}^{**}\}_k$ as focal points of the *data clouds*;

viii. **While** $\{\boldsymbol{\mu}^o\}_k$ are not fixed

* Use the focal points to form the *data clouds* from $\{\boldsymbol{x}\}_k$ using eq. (14);

* Obtain the parameters $\{\boldsymbol{\mu}\}_k$ and $\{\sigma\}_k$ of the *data clouds*;

* Calculate $D_k^{MM}(\boldsymbol{\mu}_j)$ ($j=1,2,...,l_k^*$) using eq. (8);

* Find the $\boldsymbol{\mu}^{*(1)}$ with maximum D^{MM} using eq. (12);

* Send $\boldsymbol{\mu}^{*(1)}$ into $\{\boldsymbol{\mu}^*\}_k$ and $D_k^{MM}(\boldsymbol{\mu}^{*(1)})$ into $\{D_k^{MM}(\boldsymbol{\mu}^*)\}$ and delete $\boldsymbol{\mu}^{*(1)}$ from $\{\boldsymbol{\mu}\}_k$;

* $\boldsymbol{\mu}^{*r} \leftarrow \boldsymbol{\mu}^{*(1)}$;

* **While** $\{\boldsymbol{\mu}\}_k \neq \boldsymbol{\phi}$

- Find the element(s) of $\{\boldsymbol{\mu}\}_k$ which is/are nearest to $\boldsymbol{\mu}^{*r}$;
- Send the element(s) and the corresponding D^{MM} into $\{\boldsymbol{\mu}^*\}_k$;
- Delete the element(s) from $\{\boldsymbol{\mu}\}_k$;
- Set the latest element in $\{\boldsymbol{\mu}^*\}_k$ as $\boldsymbol{\mu}^{*r}$;

* **End While**

* Apply eq. (15) to filter $\{\boldsymbol{\mu}^*\}_k$ and obtain new focal points $\{\boldsymbol{\mu}^{**}\}_k$;

* $\{\boldsymbol{\mu}^o\}_k \leftarrow \{\boldsymbol{\mu}^{**}\}_k$

ix. **End While**

x. Build the *data clouds* with $\{\boldsymbol{\mu}^o\}_k$ using eq. (14).

B. Evolving local modes-based partitioning algorithm

i. **While** the new data sample $\boldsymbol{x}_{k+1}$ of the data stream is available (or until interrupted)

* **If** ($k=0$) **Then**

- $\boldsymbol{\mu}_1 \leftarrow \boldsymbol{x}_1$
- $C_1 \leftarrow 1$;
- $S_{1,1} \leftarrow 1$;
- $\boldsymbol{\mu}_{1,1} \leftarrow \boldsymbol{x}_1$;
- $X_{1,1} \leftarrow \|\boldsymbol{x}_1\|^2$;

* **Else**

- Update $\boldsymbol{\mu}_k$ and X_k to $\boldsymbol{\mu}_{k+1}$ and X_{k+1} using eqs. (2) and (3);
- **If** (Condition (eq. (16)) is met) **Then**

1. $C_{k+1} \leftarrow C_k + 1$;
2. $S_{k+1,C_{k+1}} \leftarrow 1$;
3. $\boldsymbol{\mu}_{k+1,C_{k+1}} \leftarrow \boldsymbol{x}_{k+1}$;
4. $X_{k+1,C_{k+1}} \leftarrow \|\boldsymbol{x}_{k+1}\|^2$;

- **Else**

1. Use eq. (14) to decide the i^{th} local modes $\boldsymbol{x}_{k+1}$ is assigned to;
2. Update $\boldsymbol{\mu}_{k,i}$ and $X_{k,i}$ to $\boldsymbol{\mu}_{k+1,i}$ and $X_{k+1,i}$ using eqs. (2) and (3);
3. $S_{k+1,i} \leftarrow S_{k,i} + 1$;

- **End If**

* **End If**

ii. **End While**

iii. **While** *data clouds* exhibit overlap

* **If** (Condition (eq. (17)) is met) **Then**

- merge the two overlapping *data clouds* into one using eq. (19);

* **End If**

iv. **End While**

v. Obtain $\{\boldsymbol{\mu}^o\}_k$ from the remaining local modes;

vi. Build the *data clouds* with $\{\boldsymbol{\mu}^o\}_k$ using eq. (14).

VI. Numerical Examples

In this section, a number of benchmark problems were considered to evaluate the performance of the proposed algorithm.

For better analysis, we also consider a number of performance measures:

1) Inp: the parameters that have to be predefined (user input);

2) NoC: number of clusters/*data clouds* in the processing results;

3) AvP: the average purity of the clusters/*data clouds*, but may disguise poor results [5];

4) MaP: the maximum cluster/*data cloud* purity [5];

5) MiP: the minimum cluster/*data cloud* purity [5];

6) T: the execution time (in seconds).

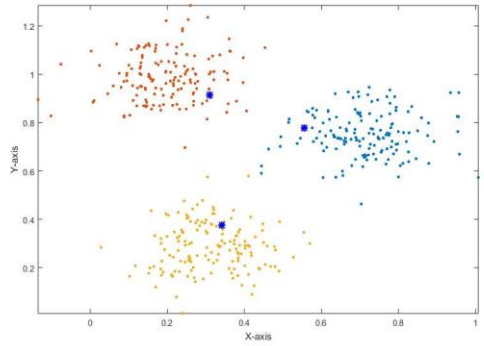
A. Evaluation of the offline version

In this subsection, we will test the performance of the

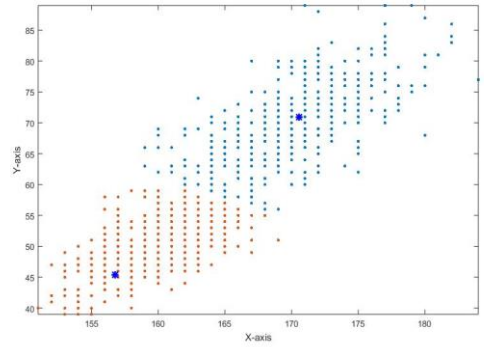
TABLE I. Dataset Description-part 1

Dataset	Details		
	Number of Data Samples	*Number of Clusters*	*Number of Attributes*
G1[a]	450	3	2
G2[b]	800	2	2
WW[c]	750	5	3

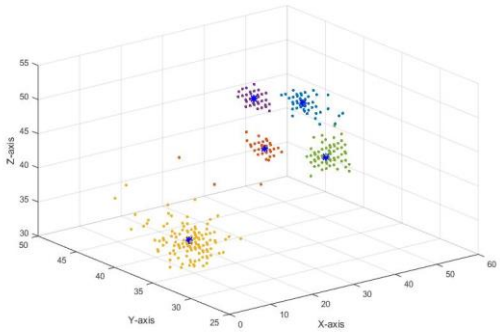
[a.] *Gaussian dataset 1;* [b.] *Gaussian dataset 2;* [c.] *Wrist-worn accelerometer dataset.*



(a) *Gaussian dataset 1*



(b) *Gaussian dataset 2*



(c) *Wrist-worn accelerometer dataset*

Fig. 1 Offline partitioning results (Blue "*" stands for the focal points, if "· " is in different colours, then it stands for different *data clouds*)

offline version of the proposed local modes-based partitioning algorithm on two synthetic Gaussian problems and a real problem (wrist-worn accelerometer dataset [15]) on a benchmark. The datasets used in the evaluation are tabulated in Table I. The partitioning results are presented in Fig. 1. The detailed experimental results are tabulated in Table II.

For further discussion of the performance, the offline version of the proposed algorithm is compared with two well-known offline clustering algorithms: mean shift [1] (needs the kernel size. σ to be predefined) and k-means [2] (needs the number of clusters to be predefined). The experimental results of the two comparative algorithms are tabulated together in the Table II as well.

As we can see from Table II, the mean shift algorithm [1] is the fastest one from all the three, however, it is less accurate.

TABEL II . Comparison between offline algorithms

	Inp	Data set	Measures				
			NoC	***AvP***	***MaP***	***MiP***	***T***
LM [a]		G1	**3**	**0.9978**	**1.0000**	**0.9934**	**12.13**
MS [b]	0.3		3	0.9911	1.0000	0.9740	0.02
	0.4		3	0.9844	1.0000	0.9554	0.02
KM [c]	3		3	0.9978	1.0000	0.9934	0.17
LM		G2	**2**	**0.9475**	**0.9661**	**0.9303**	**10.07**
MS	5		4	0.9413	1.0000	0.7647	0.03
	8		2	0.9325	0.9832	0.8914	0.03
KM	2		2	0.9475	0.9661	0.9303	0.14
LM		WW	**5**	**0.9973**	**1.0000**	**0.9868**	**5.11**
MS	10		5	0.9947	1.0000	0.9804	0.02
	12		3	0.6000	1.0000	0.5000	0.02
KM	5		5	0.9973	1.0000	0.9868	0.14

[a.] *Local Modes-based Partitioning Algorithm;* [b.] *Mean Shift Algorithm;* [c.] *K-means Algorithm.*

The k-means algorithm [2] is somehow comparable to the proposed local modes-based partitioning algorithm in terms of accuracy, but one has to keep in mind that the high accuracy of the k-means algorithm relies heavily on the properly pre-defined user input. In real cases, the number of clusters within a dataset is often hard to be decided because of the very limited *prior* knowledge. Comparatively, the proposed algorithm is not as fast as the other two algorithms, but it has the high accurate performance, and the most important point is that, it does not require any kind of user input.

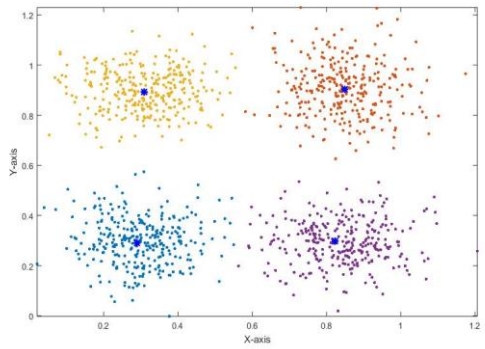
B. Evaluation of the evolving version

The performance of the evolving version of the proposed local modes-based partitioning algorithm will be evaluated in this subsection. Similarly, we will test it on two synthetic Gaussian problems and a real problem (climate dataset [16]) as well as on a benchmark. The datasets used in the evaluation are tabulated in Table III. The partitioning results are presented in Fig. 2. In addition, we compare the proposed evolving version with the well-known DBScan [4] (the radius and the minimum number of data samples within the radius need to be predefined) and ELM [5] (initial radius needs to be predefined) algorithms. The detailed experimental results of the three evolving algorithms are tabulated in Table IV.

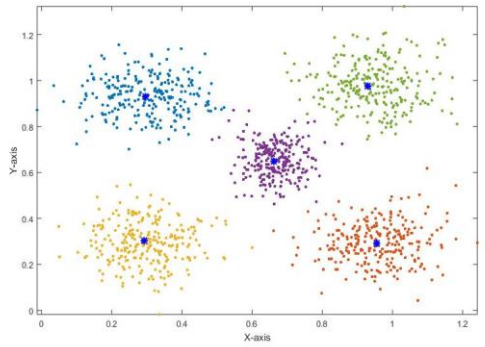
TABLE III. DATASET DESCRIPTION-PART 2

Dataset	Details		
	Number of Data Samples	***Number of Clusters***	***Number of Attributes***
G3[a]	1200	4	2
G4[b]	1250	5	2
C[c]	938	2	2

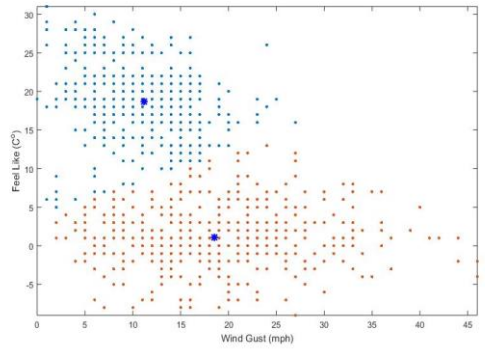
[a.] *Gaussian dataset 3;* [b.] *Gaussian dataset 4;* [c.] *Real climate dataset*



(a) *Gaussian dataset 3*



(b) *Gaussian dataset 4*



(c) *Real climate dataset*

Fig. 2 Evolving partitioning results (Blue "*" stands for the focal points; if " · " is in different colours, then it stands for different *data clouds*)

The comparison results as depicted in Table IV demonstrate that, the proposed algorithm is the most accurate one of the three, and the most importantly, it is entirely data-driven and free from any pre-defined parameters or assumptions. In contrast, the DBScan algorithm is the fastest, but it is less accurate and needs two parameters to be decided by the user. ELM algorithm exhibits comparably accurate performance with the proposed algorithm; however, this also depends on the user input. With improperly defined radius, its performance can be significantly degraded.

C. Hybrid example between the offline and evolving versions

We use the same real climate dataset [16] as tabulated in Table II to study a hybrid between the offline version to start and the evolving version afterwards. The offline version is applied to the first 300 data samples of the climate dataset [16] and the evolving version then is used to process the remaining 638 data samples as a data stream. The results of using the hybrid are shown in Fig. 3. From the figure we can see that the offline version builds two *data clouds* based on the 300 data samples. Then, the evolving version takes over the task and continues to process the rest of the data samples. The density changes with more data samples arriving and more clusters being formed based on the new data samples. Once, there are no anymore new data samples, the algorithm performs a filtering operation of the main modes and successfully identifies the two focal points representing the two main modes of the data density. Finally, the two focal points are used to form the two *data clouds*.

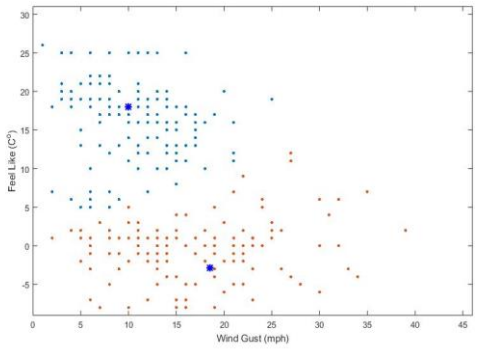
TABEL IV . Comparison between evolving algorithms

	Inp	Data set	Measures				
			NoC	***AvP***	***MaP***	***MiP***	***T***
LM		G3	**4**	**0.9958**	**1.0000**	**0.9900**	**2.00**
DB [a]	0.06, 10 [c]		4	0.9325	1.0000	0.9965	0.10
	0.08, 10		3	0.7350	1.0000	0.5017	0.16
ELM [b]	0.04		10	0.9967	1.0000	0.9870	1.06
	0.06		3	0.7425	0.9802	0.5000	0.41
LM		G4	**5**	**0.9920**	**1.0000**	**0.9689**	**2.24**
DB	0.06, 10		5	0.9464	1.0000	0.9957	0.11
	0.08, 10		3	0.5928	1.0000	0.3360	0.17
ELM	0.03		9	0.9912	1.0000	0.9689	1.18
	0.04		6	0.8896	1.0000	0.6507	0.87
LM		C	**2**	**0.9691**	**0.9747**	**0.9634**	**1.58**
DB	1.6, 10		7	0.7985	1.0000	0.9966	0.04
	2,10		3	0.8902	1.0000	0.9908	0.06
ELM	3		4	0.9670	1.0000	0.9592	0.42
	5		3	0.6215	1.0000	0.5639	0.24

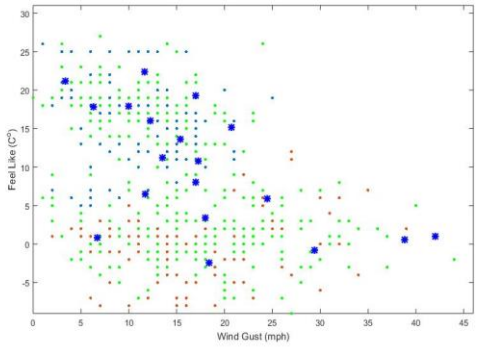
[a] DBScan Algorithm; [b] ELM Algorithm; [c] [Radius, minimum number].

VII. CONCLUSION

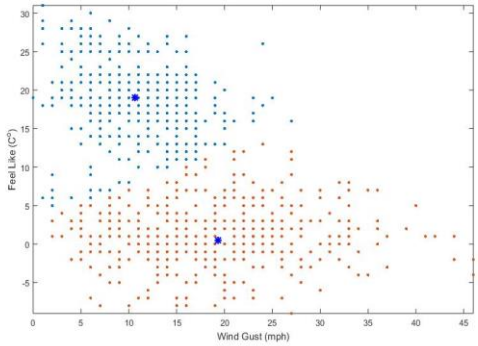
In this paper, a novel "local modes-based" partitioning algorithm is introduced as a fully autonomous technique to partition the data space into parameter-free *data clouds* according to the modes of the distribution of the data pattern. The proposed algorithm is driven entirely by the observed data samples and is free from any kind of user- and problem-specific parameters. The algorithm has two versions, offline and evolving; each one of them can operate independently, and they can perform a hybrid as well. Numerical experiments demonstrate the validity of the proposed algorithm as a fully autonomous data partitioning technique and also shows its advantages as compared with the well-known comparative algorithms.



(a) *Offline result*



(b) *300 new data samples processed as a data stream* (*green "· "*)



(c) *Final results*

Fig. 3 An example of a hybrid method (Blue "*" stands for the focal points; if "· "is in different colours, then it stands for different *data clouds*)

REFERENCES

[1] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, 1975, vol. 21, no. 1, pp. 32–40.

[2] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, no. 233, pp. 281–297.

[3] S. L. Chiu, "Fuzzy model identification based on cluster estimation.," *Journal of intelligent and Fuzzy systems*, 1994, vol. 2, no. 3. pp. 267–278.

[4] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, 1996, vol. 96, pp. 226–231.

[5] R. Dutta Baruah and P. Angelov, "Evolving local means method for clustering of streaming data," in *IEEE International Conference on Fuzzy Systems*, 2012, pp. 10–15.

[6] R. R. Yager and D. P. Filev, "Approximate clustering Via the mountain method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 8, pp. 1279–1284, 1994.

[7] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *International Journal of General Systems*, vol. 41, no. 2, pp. 163–185, 2011.

[8] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, Spatial tessellations: concepts and applications of Voronoi diagrams, 2nd ed. Chichester, England: John Wiley & Sons., 1999.

[9] P. Angelov, "Outside the box: an alternative data analytics framework," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 8, no. 2, pp. 53–59, 2014.

[10] P. Angelov, "Typicality distribution function – a new density- based data analytics tool," in *IEEE International Joint Conference on Neural Networks* (IJCNN), 2015, pp. 1–8.

[11] P. P. Angelov, "Anomaly detection based on eccentricity analysis," in *IEEE Symposium Series in Computational Intelligence, IEEE Symposium on Evolving and Autonomous Learning Systems, EALS, SSCI*, 2014, pp. 1–8.

[12] P. P. Angelov, X. Gu, J. Principe, and D. Kangin, "Empirical data analysis - a new tool for data analytics," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2016, p. in press.

[13] J. Saw, M. Yang, and T. Mo, "Chebyshev inequality with estimated mean and variance," *The American Statistician*, 1984, vol. 38, no. 2, pp. 130–132.

[14] P. Angelov, Autonomous Learning Systems: From Data Streams to Knowledge in Real Time. John Willey, 2012.

[15] https://archive.ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+with+Wrist-worn+Accelerometer

[16] http://www.worldweatheronline.com