A New Evolving Clustering Algorithm for Online Data Streams

Clauber Gomes Bezerra*, Bruno Sielly Jales Costa†, Luiz Affonso Guedes‡ and Plamen Parvanov Angelov§
*Campus EaD
Federal Institute of Rio Grande do Norte - IFRN
Natal, Brazil
Email: clauber.bezerra@ifrn.edu.br
†Campus Natal - Zona Norte
Federal Institute of Rio Grande do Norte - IFRN
Natal, Brazil
Email: bruno.costa@ifrn.edu.br
‡Department of Computer Engineering and Automation
Federal University of Rio Grande do Norte - UFRN
Email: affonso@dca.ufrn.br
§Lancaster University, Data Science Group, School of Computing and Communications,
Lancaster LA1 4WA, United Kingdom
Chair of Excellence, Carlos III University, Madrid, Spain
Email: p.angelov@lancaster.ac.uk

***Abstract*—In this paper, we propose a new approach to fuzzy data clustering. We present a new algorithm, called TEDA-Cloud, based on the recently introduced TEDA approach to outlier detection. TEDA-Cloud is a statistical method based on the concepts of typicality and eccentricity able to group similar data observations. Instead of the traditional concept of clusters, the data is grouped in the form of granular unities called *data clouds*, which are structures with no pre-defined shape or set boundaries. TEDA-Cloud is a fully autonomous and self-evolving algorithm that can be used for data clustering of online data streams and applications that require real-time response. Since it is fully autonomous, TEDA-Cloud is able to "start from scratch" (from an empty knowledge basis), create, update and merge data clouds, in a fully autonomous manner, without requiring any user-defined parameters (e.g. number of clusters, size, radius) or previous training. Moreover, TEDA-Cloud, unlike most of the traditional statistical approaches, does not rely on a specific data distribution or on the assumption of independence of data samples. The results, obtained from multiple data sets that are very well known in literature, are very encouraging.**

***Index Terms*—clustering, data streams, evolving systems, autonomous learning, real-time, TEDA, typicality, eccentricity.**

I. Introduction

Nowadays, data clustering techniques are widely used in many different fields of application, such as image processing [1], pattern recognition [2], data mining [3], biological data analisys [4] and so on. Due to the high number of existing applications, many different approaches to data clustering have been proposed in literature. Usually, different techniques are more or less suitable for different types of application. For instance, for applications that require real-time data acquisition, the input to the clustering algorithm is in the form of online data streams. Thus, n-dimensional data observations are obtained, one by one, at a specific sampling rate.

Among the most traditional clustering algorithms, one can mention k-means [5] and k-NN (k-Nearest Neighbor) [6]. Both techniques are very easy to understand and implement, which make them very popular and applicable to many different problems [7], [8], [9], [10]. However, as most of the traditional clustering approaches, they are not suitable for applications based on online data streams.

For example, k-means algorithm requires an offline batch data processing. Moreover, it has a few restrictions regarding the shape of clusters and it is not very suitable for noisy data. On the other hand, k-NN requires previous training before the actual clustering. Both techniques also require that the number of clusters be known in advance. This value is one of the inputs to the algorithm.

There are several different algorithms in literature that try to solve many of these problems, as DBSCAN [11], for example. However, they are still not suitable for online data streams. A proper approach to this problem should be able to handle large amounts of data, in a continuous and (theoretically) infinite flow, being able to cope with time and memory constraints [12], [13], [14].

Furthermore, concept-drift and concept-evolution are, frequently, ignored by many of the state-of-the-art clustering techniques. In the first case, a clustering algorithm should be able to continuously adapt considering that the underlying concept of the data changes over time. In the second case, it should not assume that the number of clusters/data structures in the data stream is fixed, since novel structures might emerge when new data samples are available [15].

978-1-5090-2583-1/16/$31.00 ©2016 IEEE

2016 IEEE Conference on
Evolving and Adaptive Intelligent Systems

We propose in this paper a new fuzzy clustering algorithm for online data stream. This algorithm is called TEDA-Cloud, builds upon the family of the work proposed by [16], [17] and its core relies on TEDA, a recently introduced algorithm for outlier detection. TEDA-Cloud uses statistical measures, such as mean and variance, to decide when and how to create, develop, update and merge clusters/data clouds according the input data stream. It meets the main mentioned requirements for online and dynamic data clustering, such as low computational cost, and is fully autonomous and able to self-evolve, without needing any user-defined parameters or previous training.

The remainder of this paper is organized as follows: Section II briefly describes TEDA algorithm. In Section III, TEDA-Cloud algorithm is introduced in details. In Section IV, the obtained results from popular clustering data sets are presented. Finally, Section V presents the conclusions and future directions.

II. TEDA

TEDA (Typicality and Eccentricity Data Analytics) is an algorithm proposed by [18] to anomaly detection in data streams. It is a statistical method based on the concepts of typicality and eccentricity of data. These concepts are complementary and based on the distance from a particular data observation to the entire data set. Moreover, TEDA is a recursive and non-parametric algorithm, which makes it suitable for online and real-time applications. Among different applications, TEDA was recently applied to industrial fault detection problems [19].

Figure 1 presents an illustration of the idea of typicality and eccentricity. The typicality at point A is low and, thus, the eccentricity is high, while the typicality at point B is high and, thus, the eccentricity is low.

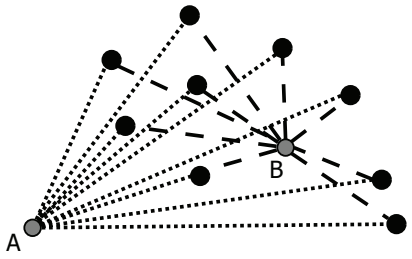

Fig. 1. Illustration of the concepts of typicality and eccentricity

The eccentricity ξ of a data sample x obtained at the discrete time instant k is defined by [18]

$$\xi(x_k) = \frac{1}{k} + \frac{(\mu_k - x_k)^T(\mu_k - x_k)}{k\sigma_k^2} \quad (1)$$

where μ_k is the mean and σ_k^2 is the variance of the data set after k samples. Both values are recursively updated by [18]

$$\mu_k = \frac{k-1}{k}\mu_{k-1} + \frac{1}{k}x_k, \quad \mu_1 = x_1 \quad (2)$$

$$\sigma_k^2 = \frac{k-1}{k}\sigma_{k-1}^2 + \frac{1}{k-1}\|x_k - \mu_k\|^2, \quad \sigma_1^2 = 0 \quad (3)$$

The typicality τ is defined as a complement to the eccentricity as [18]

$$\tau(x_k) = 1 - \xi(x_k) \quad (4)$$

The normalized eccentricity is defined as [18]

$$\zeta(x_k) = \frac{\xi(x_k)}{2} \quad (5)$$

For outlier detection under any data distribution, but, assuming a representative large amount of independent data samples, it is possible to use the well known Chebyshev inequality [20], which states that no more than $1/m^2$ of the data observations are more than $m\sigma$ away from the mean, where σ represents the standard deviation of the data. The authors in [21] show that the condition that provides exactly the same result (but without making any assumptions on the amount of data, their independence and so on) as the Chebyshev inequality and can be used as threshold for outlier detection is

$$\zeta(x_k) > \frac{m^2+1}{2k}, \quad m > 0 \quad (6)$$

where m represents the number of standard deviations (e.g. for $m = 3$, the threshold for classifying a data sample x_k as an outlier is $\zeta(x_k) > 5/k$)).

III. Proposed Method

The proposed method for clustering, as previously mentioned, is based on TEDA algorithm. It is called TEDA-Cloud and it is suitable for applications where the input is in the form of an online data stream.

The first major difference from TEDA-Cloud to most of the clustering approaches in literature is that, the granular data structures, here called *data clouds*, do not have predefined shapes or boundaries as the traditional data clusters. Data clouds are sets of previous data samples with common properties – closeness in terms of input mapped in the n-dimensional feature space. They directly represent all previous data samples. An example of two data clouds on a 2-dimensional feature space is presented in Figure 2 [22].

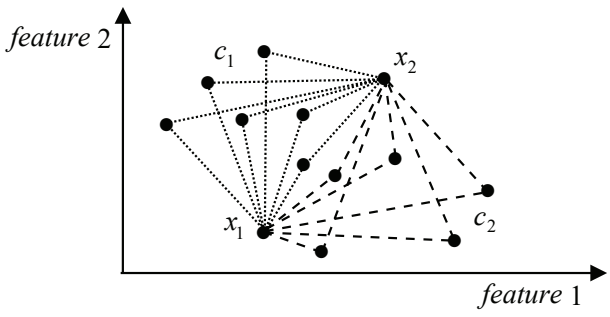

Fig. 2. Data clouds. Data points from cloud c_1 are linked with the data item x_1 using dotted lines and, respectively, the data points from cloud c_2 are linked with the data item x_2 using dashed lines. It is obvious that data clouds (unlike clusters) have no specific shape.

In contrast to this, the traditional clusters (e.g. ellipsoidal) do not represent the true data distributions; instead, they represent some desirable/expected/estimated (often subjectively) preferences [23]. In terms of fuzzy membership, fuzziness of the proposed method is guaranteed in the sense that a particular data sample can belong to all data clouds with different membership degrees, $\gamma \in [0; 1]$.

Basically, TEDA-Cloud derives each equation from TEDA to a generalized form, where each data cloud is handled, independently, as a distinct data set. TEDA-Cloud, then, determines the membership of each read data sample to each existing data cloud, based on equation 6.

Since TEDA is a recursive algorithm, the data are not required to be stored in memory. Instead, only three main statistical features are required for each data cloud c_i: the number of samples that belong to the i-th cloud, s_k^i, the mean μ_k^i, and variance, $[\sigma^2]_k^i$, of its samples, after k observations. As an example, Figure 3 illustrates two data clouds (c_1 and c_2) after k observations. Note that they are represented by circular structures for easier representation, however, it is important to stress that data clouds do not have specific shapes since they represent the true distribution of the data.

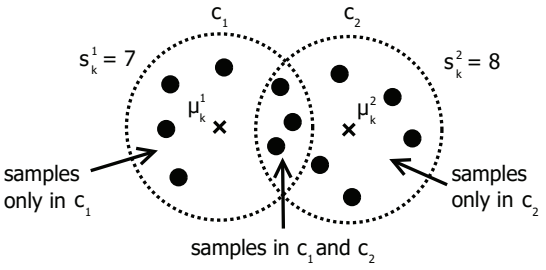

Fig. 3. Data clouds c_1 and c_2 after k observations.

Note that, the number of data points that belong to data clouds c_1 and c_2 is $s_k^1 = 7$ and $s_k^2 = 8$, respectively. It is important to highlight that, due to the fuzzy aspect of TEDA-Cloud, one data sample can simultaneously belong to more than one data cloud, often creating an intersection region between two or more clusters. The mean of c_1 and c_2, μ_k^1 and μ_k^2, respectively, graphically represent the center of the data clouds. Finally, the variances of c_1 and c_2, $[\sigma^2]_k^1$ and $[\sigma^2]_k^2$, respectively, represent the spread of the data clouds.

A. Data cloud updating

For each read data sample, TEDA-Cloud checks if the eccentricity $\zeta(x_k)^i$ of the data sample x_k in relation to the data cloud c_i is high. In that case, we assume that x_k is considerably *distinct* from the data belonging to c_i, causing no effect to the cloud structure. On the other hand, if x_k is *similar* to c_i, the number of points, s_k^i, mean, μ_k^i, and variance, $[\sigma^2]_k^i$, are updated. Finally, if x_k is considerably distinct from all existing data clouds, a new cloud is created.

The equations used in TEDA-Cloud are generalized versions of the equations presented in Section II. Thereby, the eccentricity, $\xi(x_k)^i$, and normalized eccentricity, $\zeta(x_k)^i$, of a data sample x_k, at the discrete time instant k, in relation to the i-th data cloud is given by

$$\xi(x_k)^i = \frac{1}{[s_k^i]'} + \frac{([\mu_k^i]' - x_k)^T([\mu_k^i]' - x_k)}{[s_k^i]'[[\sigma^2]_k^i]'} \tag{7}$$

$$\zeta(x_k)^i = \frac{\xi(x_k)^i}{2} \tag{8}$$

where $[s_k^i]'$, $[\mu_k^i]'$ e $[[\sigma^2]_k^i]'$ are temporary values and respectively the number of samples, the mean and the variance of the i-th data cloud in the event of x_k belonging to c_i (i.e. equation 6 does not hold). These values are calculated by

$$[s_k^i]' = s_{k-1}^i + 1 \tag{9}$$

$$[\mu_k^i]' = \frac{[s_k^i]' - 1}{[s_k^i]'}\mu_{k-1}^i + \frac{1}{[s_k^i]'}x_k \tag{10}$$

$$[[\sigma^2]_k^i]' = \frac{[s_k^i]' - 1}{[s_k^i]'}[\sigma^2]_{k-1}^i + \frac{1}{[s_k^i]' - 1}\|x_k - [\mu_k^i]'\|^2 \tag{11}$$

From a similar point of view, the condition to define if a data point x_k belongs to a data cloud c_i is given by

$$\zeta(x_k)^i <= \frac{m^2 + 1}{2s_k^i} \tag{12}$$

For each data cloud c_i, with $i = [1..N]$, where N is the number of existing clouds and, if equation 12 holds, the values of s_k^i, μ_k^i and $[\sigma^2]_k^i$ of the i-th data cloud are updated by the values of $[s_k^i]'$, $[\mu_x^i]'$ and $[[\sigma^2]_k^i]'$, respectively, previously calculated by equations 9, 10 and 11. If the condition does not hold, no action is taken regarding the data cloud c_i.

Figure 4 presents an illustration of that idea by showing three data clouds, c_1, c_2 and c_3, and an input data sample x_k at the k-th time instant (Figure 4(a)). TEDA-Cloud calculates the normalized eccentricities of x_k in relation to all three data clouds, $\zeta(x_k)^1$, $\zeta(x_k)^2$ and $\zeta(x_k)^3$, respectively. For this specific example, x_k belongs to c_1 and c_3, but fail to meet the requirement of equation 12 for c_2. Therefore, only c_1 and c_3 are updated, while no action is taken regarding c_2 (Figure 4(b)).

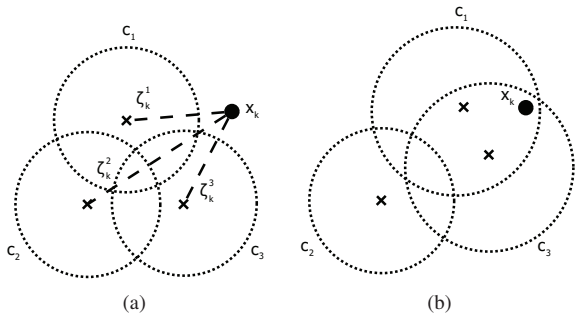

Fig. 4. Data cloud updating: (a) clouds c_1, c_2, c_3 and a newly arrived data sample x_k (b) clouds after updating.

If the equation 12 does not hold for any of the N existing clouds, a cloud c_{N+1} is created. The initialization values are

$s_k^{N+1} = 1$, $\mu_k^{N+1} = x_k$ and $[\sigma^2]_k^{N+1} = 0$. As an illustration example, consider Figure 5(a) showing three data clouds c_1, c_2 e c_3 and a newly arrived data observation x_k. It is easy to see that x_k is considerably distant from all existing data clouds and, thus, a new cloud c_4 is created, as shown in Figure 5(b).

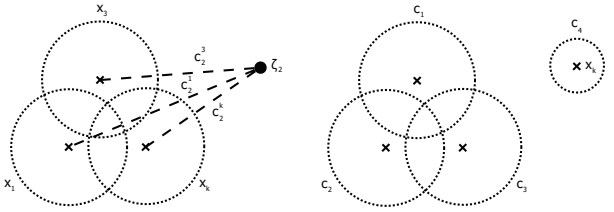

Fig. 5. Creation of a new cloud: (a) three existing clouds, c_1, c_2, c_3 and a newly arrived sample x_k and (b) a new cluster c_4

B. Data cloud merging

In order to limit the number of data clouds and, at the same time, preserving the evolving features of the approach, TEDA-Cloud is able to perform merge operation between two similar data clouds. Merge is performed when the number of intersection points between two data clouds (i.e. points that belong to both clouds simultaneously) is considerable high. The operation is fully autonomous and non-parametric.

Given two data clouds c_i and c_j, the merge is performed if at least one of the following conditions holds:

$$s_k^i \cap s_k^j > s_k^i - s_k^i \cap s_k^j \tag{13}$$

$$s_k^i \cap s_k^j > s_k^j - s_k^i \cap s_k^j \tag{14}$$

In detail, two clouds are merged when the amount of samples that belong to both clouds is higher than the amount of samples that exclusively belong to one of them.

Figure 6 illustrates the merging process in TEDA-Cloud. Figure 6(a) shows two clouds, c_1 and c_2, where the number of intersection samples is lower than the number of points that exclusively belong to one of them. Therefore, there is no merging. On the other hand, Figure 6(b) presents an example where merge is performed, since the number of intersection samples is higher than the number of samples that exclusively belong to c_2. The result of the operation, cloud c_3, is presented in Figure 6(c).

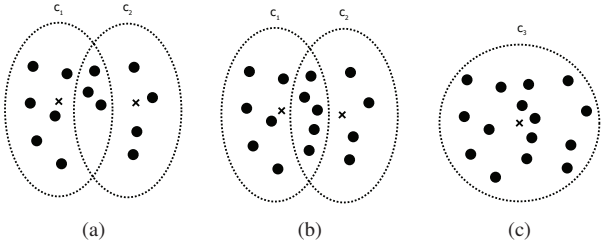

Fig. 6. Cloud merging: (a) clouds c_1 and c_2, (b) c_1 and c_2 instantly before merging and (c) resulting cloud c_3 after merging.

The properties of the resulting cloud c_l are defined as

$$s_k^l = s_k^i + s_k^j - s_k^i \cap s_k^j \tag{15}$$

$$\mu_k^l = \frac{s_k^i \mu_i + s_k^j \mu_j}{s_k^i + s_k^j} \tag{16}$$

$$[\sigma^2]_k^l = \frac{(s_k^i - 1)\sigma_i^2 + (s_k^j - 1)\sigma_j^2}{s_k^i + s_k^j - 2} \tag{17}$$

Finally, Figure 7 presents in detail TEDA-Cloud algorithm. At the end of the online processing of each data sample, the output of TEDA-Cloud is a list containing the number of points, mean and variance of each existing data cloud.

IV. Results

In order to validate the proposal, we applied TEDA-Cloud to four data sets that are openly available and very well known in literature [24]. They are synthetic data sets and were used many times for validation of clustering algorithms [25], [26]. They are called A_1, A_2, S_1 and S_2 and each of them has n data samples and N clusters. Data sets A_1 and A_2 are bi-dimensional with circular clusters, while S_1 and S_2 are also bi-dimensional, however, with complex and non-symmetric shapes. The choice for bi-dimensional data sets was based on the fact that they are easy to visualize, however, TEDA-Cloud can also be easily applied to high-dimensional data.

Although traditionally used for batch processing, the data sets were made available in the form of data streams, one sample x_k at a time, where $k = [1..n]$. Table I presents, for each data set, the number of samples, n, the number of existing clusters (provided by the data set manual), N, and the number of data clouds returned by TEDA-Cloud, $\overline{N}$.

TABLE I
Results obtained using TEDA-C.

Data Set	n	N	$\overline{N}$
S_1	5000	15	15
S_2	5000	15	15
A_1	3000	20	20
A_2	5250	35	35

It is easy to observe that, after reading all available data samples, the number of data clouds returned by TEDA-Cloud is exactly the same as the number of actual clusters, for all tested data sets.

Figure 8 presents the data sets used for validation after TEDA-Cloud processing. The blue points are the data samples of each set, while the mean of each data cloud returned by TEDA-Cloud is highlighted as a red 'x'.

Finally, Figure 9 illustrates the creation and merging of data clouds over time, from the very first data sample ($k = 1$) to the end of the data set ($k = n$). Again, the data samples were made available in the form of a data stream, in a ordered sequence, as provided by the data sets. The number of detected data clouds in all examples in Figure 9 increases as the number of clusters grow. The overlapping regions represent the creation and (nearly) instant merge of data clouds during

while $x_k \leftarrow$ *read next data sample* **do**
  if $k = 1$ **then**
    // create cloud 1 and add x_1
    $s_1^1 \leftarrow 1$
    $\mu_1^1 \leftarrow x_1$
    $[\sigma^2]_1^1 \leftarrow 0$
  else
    if $k = 2$ **then**
      // add x_2 *to existing cloud 1*
      $s_2^1 \leftarrow 2$
      $\mu_2^1 \leftarrow ((s_2^1 - 1)/s_2^1) * \mu_1^1 + (1/s_2^1) * x_2$
      $[\sigma^2]_2^1 \leftarrow ((s_2^1 - 1)/s_2^1) * [\sigma^2]_1^1 +$
        $(1/(s_2^1 - 1) * \|x_2 - \mu_2^1\|$
    else
      if $k >= 3$ **then**
        for all *existing clouds* i **do**
          calculate $[s']_k^i$ *by equation 9*
          calculate $[\mu']_k^i$ *by equation 10*
          calculate $[[\sigma^2]']_k^i$ *by equation 11*
          calculate ξ_k^i *by equation 7*
          calculate ζ_k^i *by equation 8*
          if $\zeta_k^i <= (m^2 + 1)/2 * m$ **then**
            // add x_k *to cloud* i
            $s_k^i \leftarrow [s']_k^i$
            $\mu_k^i \leftarrow [\mu']_k^i$
            $[\sigma^2]_k^i \leftarrow [[\sigma^2]']_k^i$
          else
            $s_k^i \leftarrow s_{k-1}^i$
            $\mu_k^i \leftarrow \mu_{k-1}^i$
            $[\sigma^2]_k^i \leftarrow [\sigma^2]_{k-1}^i$
          end if
        end for
        if $x_k \notin c_i, \forall i$ **then**
          //create cloud l
          $n_k^l \leftarrow 1$
          $\mu_k^l \leftarrow x_k$
          $[\sigma^2]_k^l \leftarrow 0$
        end if
        for all *pairs of clouds* i *and* j ***do***
          if $s_k^i \cap s_k^j > s_k^i - s_k^i \cap s_k^j$ ***or***
            $s_k^i \cap s_k^j > s_k^j - s_k^i \cap s_k^j$ ***then***
            //merge clouds i *and* j *in cluster* l
            calculate s_k^l *by equation 15*
            calculate μ_k^l *by equation 16*
            calculate $[\sigma^2]_k^l$ *by equation 17*
          end if
        end for
      end if
    end if
  end if
end while

Fig. 7. Proposed TEDA-Cloud algorithm.

the incremental/evolving online learning process. The creation of a new cloud increments $\overline{N}_k$ by 1, while merging two similar/close data clouds decrements $\overline{N}_k$ by 1.

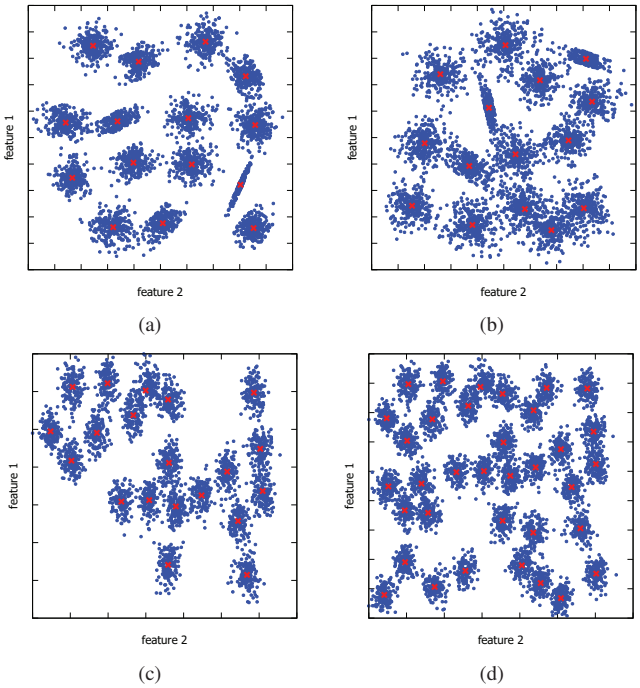

Fig. 8. Data sets used for validation and the mean of each data cloud returned by TEDA-Cloud: (a) S1, (b) S2, (c) A1 and (d) A2.

It should be noted that the variation of $\overline{N}$ between close samples is limited in 1 cloud, which shows that the merging operation is based on appropriate criteria, where neither creation nor merging of clouds is performed in an unordered manner.

V. Conclusion

In this paper we presented a new algorithm, called TEDA-Cloud, for clustering/grouping of online data streams. TEDA-Cloud is based on the statistical analysis of the input data samples, based on the eccentricity of each sample to all existing data and existing data groups. The eccentricity determines if a particular data sample belongs to a specific group. Instead of the traditional concept of clusters, TEDA-Cloud is based on the concept of data clouds, which are data structures with no pre-specified shapes and boundaries. It is also based on a non-exclusive clustering model, where each sample can belong to more than one cluster simultaneously, with different membership degrees.

TEDA-Cloud is very suitable for problems that require real-time data analytics, since it is a recursive algorithm with very low computational cost. Moreover, it is fully autonomous and does not require any user-defined parameter or previous knowledge about the data, such as number of clusters, radius, size and distribution. TEDA-Cloud does not require a previous training, since it is based on the idea of incremental/evolving learning and is not restricted by size, balance or shape of the clusters.

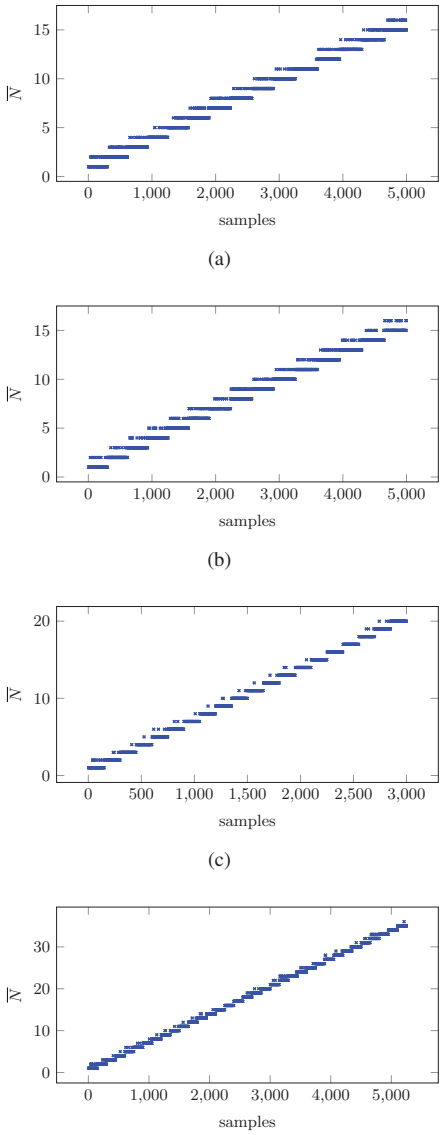

Fig. 9. Creation and merging of data clouds over time: (a) S1, (b) S2, (c) A1 and (d) A2.

TEDA-Cloud is a self-evolving algorithm, since it is able to autonomously create and merge clusters over time. Both operations are complementary in the sense that they might correct the estimated model of the system when new data samples are available.

The results obtained with TEDA-Cloud from four very well known clustering data sets are very satisfactory. All presented clusters were promptly detected and successfully located. As future work, TEDA-Cloud will be applied to real-world and real-time problems and a new unsupervised fuzzy classifier based on TEDA-Cloud framework will be proposed.

REFERENCES

[1] V. Dehariya, S. Shrivastava, and R. Jain, "Clustering of image data set using k-means and fuzzy k-means algorithms," in *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, Nov 2010, pp. 386–391.

[2] X. Peng, C. Zhou, D. Hepburn, M. Judd, and W. Siew, "Application of k-means method to pattern recognition in on-line cable partial discharge monitoring," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 20, no. 3, pp. 754–761, June 2013.

[3] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-means algorithms for very large data." *IEEE T. Fuzzy Systems*, vol. 20, no. 6, pp. 1130–1146, 2012.

[4] L. M. O. Mesa, L. F. N. Vasquez, and L. L. Kleine, "Identification and analysis of gene clusters in biological data," in *2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops, BIBMW 2012, Philadelphia, USA, October 4-7, 2012*, 2012, pp. 551–557.

[5] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[6] E. Fix and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination: Consistency properties," *US Air Force School of Aviation Medicine*, vol. Technical Report 4, no. 3, pp. 477+, Jan. 1951.

[7] I. Soraluze, C. Rodriguez, F. Boto, and A. Perez, "Multidimensional multistage k-nn classifiers for handwritten digit recognition," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, 2002, pp. 19–23.

[8] A. Junoh and M. Mansor, "Home security system based on fuzzy k-nn classifier," in *Instrumentation Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on*, vol. 2, Aug 2012, pp. 361–363.

[9] Q. Deng and G. Mei, "Combining self-organizing map and k-means clustering for detecting fraudulent financial statements," in *Granular Computing, 2009, GRC '09. IEEE International Conference on*, Aug 2009, pp. 126–131.

[10] D. Wu, Q. Yang, F. Tian, and D. X. Zhang, "Fault diagnosis based on k-means clustering and pnn," in *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*, Nov 2010, pp. 173–176.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. AAAI Press, 1996, pp. 226–231.

[12] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 13:1–13:31, Jul. 2013.

[13] R. D. Baruah, P. P. Angelov, and D. Baruah, "Dynamically evolving fuzzy classifier for real-time classification of data streams," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 383–389.

[14] A. Bouchachia, "Evolving clustering:an asset for evolving systems," *IEEE SMC Newsletter*, no. 36, 2011.

[15] M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 6, pp. 859–874, June 2011.

[16] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier," *Neurocomputing*, vol. 150, Part A, pp. 289 – 303, 2015.

[17] B. Costa, P. Angelov, and L. A. Guedes, "A new unsupervised approach to fault detection and identification," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 1557–1564.

[18] P. Angelov, "Anomaly detection based on eccentricity analysis," in *IEEE Symposium on Evolving and Autonomous Learning Systems, EALS 2014, Orlando, FL, USA, December 9-12, 2014*, 2014, pp. 1–8.

[19] B. S. J. SCosta, C. G. Bezerra, L. A. Guedes, and P. P. Angelov, "Online fault detection based on typicality and eccentricity data analytics," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, July 2015, pp. 1–6.

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

[21] A. Bernieri, B. G., and L. C., "On-line fault detection and diagnosis obtained by implementing neural algorithms on a digital signal processor," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, p. 894899, 1996.

[22] P. Angelov, *Autonomous Learning Systems: From Data to Knowledge in Real Time*. John Willey and Sons, 2012.
[23] 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, 2012.
[24] "Clustering datasets - joensuu," https://cs.joensuu.fi/sipu/datasets/, accessed: 2016-01-25.
[25] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recogn.*, vol. 39, no. 5, pp. 761–775, May 2006.
[26] I. Krkkinen and P. Frnti, "Dynamic Local Search Algorithm for the Clustering Problem," Department of Computer Science, University of Joensuu, Tech. Rep. A-2002-6, 2002.