Kennedy R. LOPES[1], Ana C. C. ANDRADE[1], Bernardo F. A. LIMA[1], André L. MAITELLI[1]

Laboratory of Petroleum Automation - LAUT, Federal University of Rio Grande do Norte - UFRN (1)

doi:10.15199/48.2019.03.22

Development of automatic classifier for sensor measurements of an industrial process

Abstract. *Clustering algorithms are usually based on an initial estimate of cores, have performance dependent on the number of clusters and dimension of the data, and are performed offline. Thus, by categorizing a highly coupled sensor network as an industrial plant, it is necessary that all these characteristics are achieved. The article presents an improvement of the TEDA-Cloud, based on the Typicity and Eccentricity Data Analitics (TEDA). In this way,the proposed (TEDA-Cloud modified), method reduces the amount of stored data for merging cores and speeds up the classification of the presented data.*

Streszczenie. *Algorytmy klastrowania są zwykle oparte na wstępnym oszacowaniu rdzeni, mają wydajność zależną od liczby klastrów i wymiarów danych i są wykonywane w trybie offline. Zatem, poprzez kategoryzowanie wysoce sprzężonej sieci czujników jako instalacji przemysłowej, konieczne jest, aby wszystkie te cechy zostały osiągnięte. W artykule przedstawiono ulepszenie chmury TEDA opartej na analizie typowości i ekscentryczności danych (TEDA). W ten sposób proponowana (zmodyfikowana TEDA-Cloud) metoda zmniejsza ilość przechowywanych danych do łączenia rdzeni i przyspiesza klasyfikację prezentowanych danych.* ***(Automatyczna klasyfikacja w procesie przemysłowym w sieci czujników)***

Keywords: Clustering, eccentricity, typicality, sensor networks, stream data, industrial process
Słowa kluczowe: Klastrowanie, ekscentryczność, typowość, sieci czujników, dane strumieniowe, proces przemysłowy

Introduction

Clustering processes, in general, are dependent on the problem to be presented. If we know in advance the number of cores representing the data or the tolerance needed to consider the creation of a new core, then we will have a good ranking. However, these conditions are not always available. In an industrial process, for example, the need to identify patterns through clustering is hampered by the fact that known methods are performed offline [1].

The K-Means is an example of a clustering algorithm that needs in advance information on the number of core. In addition, the procedure is computationally costly because all data is labeled at each new location of the cores. This makes it unfeasible to use the algorithm in real-time processes.

Due to the limitations of existing clustering methods, a new clustering method has been proposed, known as TEDA-Cloud [2]. The method aims to solve all these limitations by applying the concepts of eccentricity and typicity (TEDA) that allows recursive methods of statistical calculations to be determined on the data [3].

Even with the improvement from TEDA-Cloud, clustering still needs to store which points are about the classification of each pattern and also the number of points that are under the effect of two cores. The TEDA-Cloud classifies the data using statistical concepts that indicate a probability of the point being associated with each cores. If the eccentricity is high enough to decide that the points presented do not fit any pattern, then a new core is formed. However, if the data represented by a core are very scattered, they can affect the abagence of another core, allowing them to be joined together. The determination of whether two cores should be joined is calculated by the number of points that are together in the two. This makes TEDA-CLoud a high computational cost.

The TEDA-Cloud modified brings the improvement of the way the fusion is defined. The method avoids the labeling of new data and also does not require that any additional data be stored in the memory beyond the location of the cores and their coverage on the data.

Description of the methodology

The clustering and classification of online form can be presented as Fig. 1. The data are presented to the algorithm that defines which cores represent the samples. If time samples are taken, the clusters may be used during the core determination procedure. If the data is already stored in memory, the validation step can occur when all patterns have already been formed.

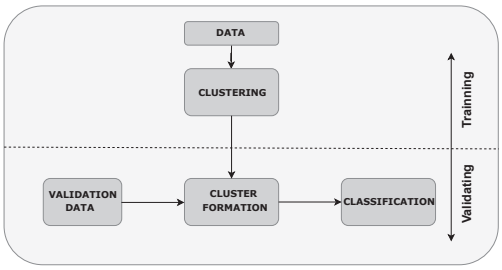

Fig. 1. Process of clustering and classification

k-means

One of the first algorithms based on the clustering method was called k-means. This method is widely known to have easy-to-understand techniques for implementation [4]. Based on the basic methodologies of groupings, it presents applicability in the most diverse areas as: medicine [5], economics [6], in market research [7], engineering [8], psychological [9], among other areas in search of solutions of its problems [10]. The k-means clustering method begins with k seed data points. These points are used as the centers of the clusters and any other remaining data sample added is assigned to the nearest cluster centres. Then, the means are calculated. Finally the samples are reassigned again, until the clusters do not change. To peform well, this method requires a prior knowledge about the number of clusters and a user input [11]. Therefore, it is not possible to use this method to create clusters on-line [1].

Teda

For a better understanding related to typicality and eccentricity, it is important to understand that these concepts are based on a spatial perspective, but specifically on the distance between the points of a sample. This distance can be Euclidian, Mahalonobis, Cosseno, Manhattan and so on [12, 13, 14]. Therefore, by knowing the distances between points, it is possible to calculate the centrality of a given point, which can be interpreted as how close or similar a data is to the other data in the sample. Calculating the centrality, it is possible to calculate the cumulative proximity, which is the inverse of the centrality squared, being represented by (1):

(1) $$q_N(x_i) = \sum_{i=1}^{N} d^2(x_i, x_j), N > 1$$

Eccentricity is the normalized cumulative proximity. Thus, the eccentricity, ξ_N, for a given point x_i in a data set is calculated as (2):

(2) $$\xi_N(x_i) = \frac{2q_N(x_i)}{\sum_{j=1}^{N} q_N(x_j)}$$

Substituting (1), which is the cumulative proximity equation, in (2), is obtained the following equation (3) :

(3) $$\xi_N(x_i) = 2\frac{\sum_{j=1}^{N} d^2(x_i, x_j)}{\sum_{j=1}^{N}\sum_{h=1}^{N} d^2(x_j, x_h)}$$

The coefficient 2 is added in the equation because each distance is counted twice and maintains the value of the eccentricity,x_i , between 0 and 1. The typicity of a data can be calculated as the complement of the eccentricity. Therefore, the typicity can be obtained by the following expression (4):

(4) $$\tau_N(x_i) = 1 - \xi_N(x_i),\ N > 2,\ i > 1$$

Both concepts can be normalized by (5) and (6):

(5) $$\zeta_N(x_i) = \frac{\xi_N(x_i)}{2}$$

(6) $$t_N(x_i) = \frac{\tau_N(x_i)}{N-2}$$

Since $\xi_N(x_i)$, can only be between zero and one, each value of $\zeta_N(x_i)$ must be between zero and 0.5.

The normalization of typicity and eccentricity, resembles the probability distribution function (PDF), since the sum of these values resulted in 1, but they are different because they do not require the premises that are obligatory for probability theory and both represent the standard spatial distribution and frequency of occurrence of a data sample [13].

The Figure 2 provides a graphic example of the concept of eccentricity for two-dimensional data, representing two parameters. In this example, the data B is a typical data, located near the other data and the data A should be considered eccentric, because it is spatially distant from the other data. Briefly, A should be considered more eccentric than B and B is more typical than A [15].

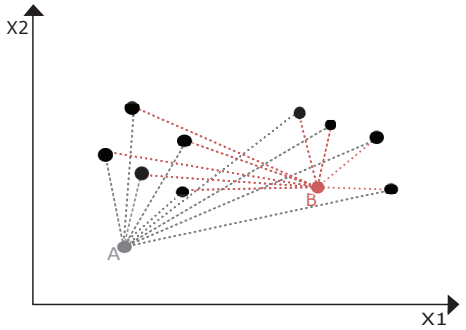

Fig. 2. Example of eccentric and typical data. Source [15].

All concepts presented are applied to data streams. For the Euclidean squared distance, the eccentricity can be calculated by updating the mean, μ, and the variance, σ^2 [13]. Then, after N samples, have (7) e (8):

(7) $$\mu_N = \frac{N-1}{N}\mu_{N-1} + \left(\frac{1}{N}\right) x_N,\ \mu_1 = x_1$$

(8) $$\sigma_N^2 = \frac{(N-1)}{N}\sigma_{N-1}^2 + \frac{1}{N-1}||\mu_N - x_N||^2,\ \sigma_1^2 = 0$$

With the disposition of the above values, the eccentricity is updated using (9):

(9) $$\xi_N(x_i) = \frac{1}{N} + (\mu_N - x_N)^T(\mu_N - x_N)/N\sigma_N^2$$

Then the typicity is updated using the Equation (5).

Teda-Cloud

TEDA-Cloud stores the relevance of each sample in its cores. Thus, the online classification attempt is inefficient, since a fixed amount of information is expected to be stored, regardless of the number of samples presented to the algorithm.

The general diagram of the TEDA-Cloud algorithm, presented in Fig. 3, shows three phases: stream data, clustering and cluster extraction. In which the main phases is the grouping process that performs the on-line calculation of eccentricity and typicity from the means and variations.

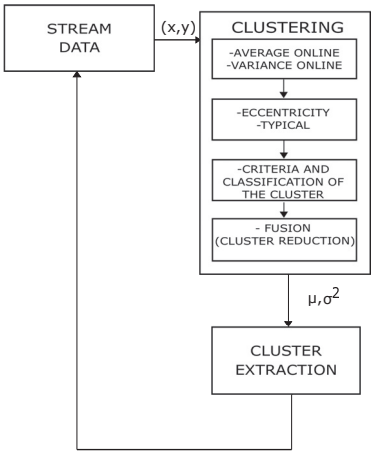

Fig. 3. General flowchart of the TEDA-Cloud method.

Only three characteristics are used for the creation of clouds, which are the number of samples belonging to the i^{th} cloud, the mean and the variance. These statistical characteristics will be used to calculate the eccentricity of a data cloud, and from the Chebyshev inequality, when a new data is present in the set, it will be checked whether that data belongs to a specific data cloud or more than one, in case of intersection between clouds. The Chebyshev inequality is described as follows [14, 16]. If X is a random variable with mean μ and variance $\sigma^2 < \infty$, then to $N > 1$, so the Chebyshev inequality is given by (10):

(10) $$P\{|X - \mu| > N\sigma\} < \frac{1}{N^2}$$

Or a similar inequality, according to (11):

(11) $$P\{|X-\mu| \geq N\sigma\} \leq \frac{1}{N^2}$$

However, to analyze this inequality from the TEDA's perspective, in terms of the normalized eccentricity, (12) is used:

(12) $$\zeta_{_N}(x_i) < \frac{(m^2+1)}{2M}$$

Where m represents the number of deviations from the mean chosen to represent the data, $\zeta_N(x_i)$ is the normalized eccentricity of the data cloud, and M is the number of samples in the data cloud.

Since a data is classified as not belonging to a data cloud, this data cloud remains unchanged, although if this data is classified as data from a cloud, that cloud will be updated recursively. First, there will be the update of the number of data in the cloud, which is done by (13):

(13) $$S^{N_i} := S^{N_i} + 1$$

Where S^{N_i} is the number of data that cloud i represents. Then the mean and variance will be updated by (14), (15):

(14) $$\mu^i_{_N} := \frac{S^{N_i}-1}{S^{N_i}}\mu^i_{_N} + \frac{1}{S^{N_i}}x_{_N}$$

(15) $$\sigma^2_{N_i} := \frac{S^{N_i}-1}{S^{N_i}}\sigma^2_{N_i}(x_i) + \frac{1}{S^{N_i}-1}||x_{_N} - \mu^i_{_N}||^2$$

After updating the statistical parameters, the eccentricity will be updated and then the normalized eccentricity (16):

(16) $$\xi_{_N}(x_i) = \frac{1}{S^{N_i}} + (\mu_{_N} - x_{_N})^T(\mu_{_N} - x_{_N})/((S^{N_i})(\sigma^2_N))$$

When a new data does not match with any of the data clouds, a new data cloud is created. In this case, the data cloud will have $S^{N_i} = 1$, $\mu^i_N = x_N$ and $\sigma^2_{N_i} = 0$. Where μ^i_N is the center of the cloud.

As mentioned earlier, there are some situations where clouds share some data, and an intersection occurs. These intersections can occur between two or more clouds. In some situations, these intersections are so relevant that clouds, which share a large amount of data, merge, creating a new cloud. Cloud merge occurs when the rules (17) or (18) are obeyed.

(17) $$S^{N_i \cap N_j} > S^{N_j - N_i}$$

(18) $$S^{N_i \cap N_j} > S^{N_i - N_j}$$

In which $S^{N_i \cap N_j}$ is the number of points at the intersection of clouds N_i e N_j.

$S^{N_i - N_j}$ the number of points that belongs to the data core N_i and not to the N_j.

This means that two clouds are merged when the number of samples belonging to the two clouds is greater than the number of samples belonging exclusively to one of them. When the clouds are merged, (19), (20) and (21) will update the new cloud properties.

(19) $$S^{N_l} = S^{N_i} + S^{N_j} - S^{N_i \cap N_j}$$

(20) $$\mu^l_N = \frac{(S^{N_i})(\mu^i_N) + (S^{N_j})(\mu^j_N)}{S^{N_i} + S^{N_i}}$$

(21) $$\sigma^2 = \frac{(S^{N_i})(\sigma^2_{N_i}) + (S^{N_j})(\sigma^2_{N_j})}{S^{N_i} + S^{N_j}}$$

Teda-cloud modified

The new method TEDA-Cloud modified proposes to accelerate the process of data-stream clustering. This is possible because there is a reduction of stored data and also because only the clouds that receive the data are selected to carry out the fusion. In addition, the Inequations (17), (18), (19), (20) and (21) have as difficulty the need to store, for each cloud, all the points to which they belong.

The values of S^{N_i}, S^{N_j} e $S^{N_i \cap N_j}$ through the formulas are only possible because the same information can be stored in several clouds. This prejudice the algorithm in computational time because there is redundancy of information. At the worst-case scenario, the same data can be in all the clouds already defined during the clustering process. Making the amount of data stored only in the process of joining the cores exceeds the amount of data already processed.

Storing the number of the classified samples for each cloud is possible. However, in addition to storing a mass of data, it is necessary that the sets in which the data belongs are defined for each sample. For example: Consider that in a given clustering step we have n clouds, each with $S^i = S^i_N$ samples classified to $1 < i < n$. Thus, we also have $S^{ij} = S^{N_i \cap N_j}$ data stored at the intersection pairs of clouds i and j, with $1 < i < j < n$. The Equation (22) presents the amount of data to be stored and analyzed on the necessity or not of the fusion between the cores.

(22) $$S_T = \sum_{i=1}^{n} S^i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} S^{ij}$$

The Table 1 shows the number of parameters required for classification using TEDA-Cloud. When there is only one core, only the amount of samples needs to be saved for clustering. With two or more, it is necessary to consider how many points are in each cloud pair. Another possibility would be to label each data with membership or not in relation to all cores and to count points for each intersection of two clusters. But this alternative requires more processing, in addition to having to store a quantity of data proportional to the size of the samples, it is necessary to make a recount of all the points that are within each range of cores for each new sample.

Table 1. Relation between the amount of clouds and stored samples of TEDA-Cloud

Number of Cores	Required parameters
1	1
2	3
3	6
4	10
...	...
n	$\frac{n(n+1)}{2}$

We then define the amount of total stored data S_T as a function of the number of cores by the (23):

$$S_T = \frac{n(n+1)}{2} \tag{23}$$

The TEDA-Cloud modified proposal is to reduce the amount of data stored in the grouping. Only their respective means and variances are stored. These values are updated dynamically and preserving them speeds up processing. Now, only $3n$ information needs to be stored per core.

However we do not have the information about the values of the intersections of the cores, $S_N^i \cap S_N^j$. These values must be approximated or a criterion based on mean and variance must be defined to predict the values of the number of points between each two cores.

The criterion used was based on the distance between the cores. The fusion will be performed when the centers are close enough to be considered that together they represent the same pattern. In this way, the choice of the union cores is only defined by the distances between centers.

To determine the union between two cores N_A and N_B, the distance between centers is used. Considering two-dimensional samples as indicated in the Fig. 4, the distances between the centers may follow the Euclidean norm.

$$d_{AB} = \sqrt{(\vec{\mu}_A - \vec{\mu}_B)^2} \tag{24}$$

The core range can be defined by the radius which is exactly the standard deviation calculated in the clustering process. If the range of one of the cores N_A or N_B, over their respective data, exceeds the distance d_{AB}, then the cores will be joined. The conditions of union are therefore:

$$d_{AB} < \mu_A + m\sigma_A \tag{25}$$

$$d_{AB} < \mu_B + m\sigma_B \tag{26}$$

Where m is the amount of standard deviations used to describe the significance of the categorized data and σ_A and σ_B the standard deviations of the clouds N_A and N_B respectively. These equations replace (17) and (18) for the implementation of TEDA-Cloud modified. In practice, cores will be united when the breadth of one cluster invades the center of another. This can be seen in the Figure 4.When the two-core coverage area has no intersection, then at the analyzed moment the clusters are interpreted as independent (Figure 4a). In 4b, the clusters have an intersection but not enough to decide that the two represent the same core. The others have an intersection large enough for one core to invade the center of the other cluster. The first being a partial intersection and the second a complete intersection. In these two situations, the cores must be united.

The new core generated should incorporate statistical characteristics of the cores that were fused. Therefore, the use of the updates by (19), (20) and (21) with the number of points $S_N^j \cap S_N^i$. Since this information is no longer saved, this expression must be approximated according to the number of points of each core, their variances and their means.

The larger cluster (of greater variance) incorporates the smaller cluster and it needs to find out the number of points associated to the joined cores. This quantity is approximated by (27).

$$S^{ij} = S^i + S^{i-j} \tag{27}$$

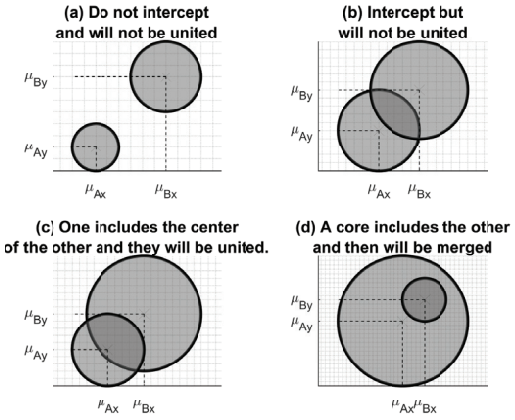

Fig. 4. Merge condition between cores

Since S^{i-j} is the number of points belonging to the smaller fused cluster and does not belong to the larger. This value is approximated to be proportional to the range difference of cores over the data

$$S^{ij} = \alpha(r_A - r_B) \tag{28}$$

With A being the cluster having variance greater than that of B and r_A and r_b their respective reaches in the same direction of a vector $\vec{\mu}_{AB}$. The α is chosen such that it is zero when the A core fully embeds the B and equals $\frac{\sigma_A}{2\sigma_B}$ when it does not incorporate. The Equations (27) and (28) allow an approximation of how many points the fused cores are to have.

In the Algorithm 1 presents the Fusion Algorithm of TEDA-Cloud modified. It is necessary that at least 2 samples of the sensors be available for classification, due to the requirements of Equations (4), (5) and (6).

Algorithm 1: Fusion Algorithm

Input : N_a - Clouds where d_x was added
m - Number of standard deviations used to represent a cluster
Output: N - New clouds merged

$S \leftarrow Pairs(N_a)$
$N \leftarrow \emptyset$
while $S \neq \emptyset$ **do**
  $(N_x, N_y) \leftarrow firstElement(S)$
  $\mu^x \leftarrow mean(N_x)$
  $\mu^y \leftarrow mean(N_x)$
  $[\sigma^2]^x \leftarrow variance(N_x)$
  $[\sigma^2]^y \leftarrow variance(N_y)$
  $dist \leftarrow |\mu^x - \mu^y|$
  $dist_x = m[\sigma^2]^x + \mu^x$
  $dist_y = m[\sigma^2]^y + \mu^y$
  remove N_x, N_y from N_a
  if $max(dist_x, dist_y) > dist$ **then**
    $N_w \leftarrow merged(N_x, N_y)$
    add N_w to N_a
    $S \leftarrow$ pairs(N_a);
  end
end

There are two procedures that allow the increase of cores variances: The first, which is a slower procedure, is the natural increase due to the dispersion of the data. At this time, all sensor measurements have near-average re-

sults and no new clustering is required. In the second moment, two cores join together in the fusion. From that moment, a new cloud of data will arise to replace the junction of the new clouds.

Tennessee Eastman Process

In order to understand better the performance of the process (TE), the Fig. 5 describes its operation. The process is composed by four gaseous reagents (A, C, D, E) and four main process, which are the reactor, condenser, vapor-liquid separator and a stripper. In the end of the process, there will be two products (G, H) and a by-product (F). The process' simulation is disposable on Simulink, with 20 disturbances, 41 measured variables and 12 manipulated variables [17, 18].

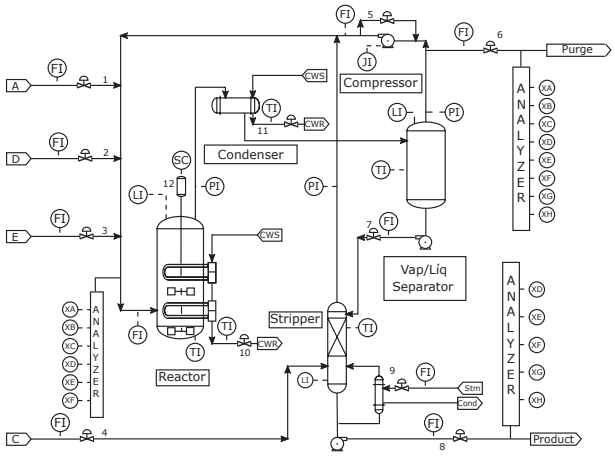

Fig. 5. Tennessee Eastman (TE) process diagram. Source [17]

Data Acquisition

The artificial data set was taken from a repository available in [19]. Five patterns are used: S-Set, A-Set, DIM-Set and Unbalace. The data were presented in order, obeying the sequence of time series and were compared on the amount of determined cores.

Four disturbances were chosen of the 20 available on simulation: Disturbances 6, 8, 13 and 14. The disturbances 6 indicates loss in feed of reagent A. The disturbance 8 represents a random variation of components A, B, and C at the Stripper inlet. Disturbance 13 is a slow reaction reaction kinetics. The disturbance 14 represents the sticking of the reactor cooling water valve. In addition to the availability of disturbances, the simulation also presents 41 sensors available for plant monitoring and control.

All data were simulated and processed in real time with the sampling rate of $Ts = 1ms$. For each new captured data, the TEDA-Cloud and TEDA-CLoud modified algorithms identify which cores it belongs to and decide its correct classification. If the sample takes more than Ts, then the new captured data enters in the stack of data to be processed.

K-means was compared only in terms of the number of cores formed, serving as a reference for the results of the online classification methods. Therefore, K-means was used with the stored samples.

Results and Discussion

Are two the important steps in determining the rate at which the time series is categorized. The process of creation and union of cores.

The creation of a new core is necessary when the recent data collected no longer represent a normality with those previously presented. In this situation, the need for a new grouping becomes important. In contrast, when cores increase in variance sufficiently to encompass another, a junction between cores must be realized.

The Figure 6 shows the evolution of the number of cores in the TE plant clustering process.

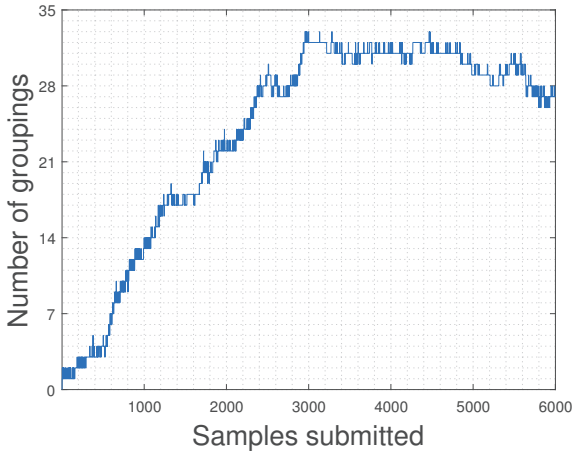

Fig. 6. Cluster formation for (TE) with disturbance 13 using $m = 1.5$.

K-means used previously stored data and the TEDA-Cloud and TEDA-Cloud modified methods will classify the samples in real time. The criteria for comparison among the methods were: number of cores determined and average occupation of the data stack to be processed. All methods determined the same amount of cores discovered by K-means. However, comparing online methods, TEDA-Cloud modified showed a lower average amount of stack occupation. The results are summarized in Table 2 for the artificial data set.

Table 2. Clustering through TEDA-Cloud and TEDA-Cloud modified

Set	N	Dim	K	Average Stack Occupation	
				TEDA-Cloud	TEDA-Cloud modified
S1	5000	2	15	0.0598	0.0507
S2	5000	2	15	0.0426	0.0421
A1	3000	2	20	2.3118	0.1890
A2	5250	2	35	5.299	0.1127
Dim512	1024	512	16	9.4672	0.4170
Dim1024	1024	1024	16	9.7913	0.4259
Unbalance	6500	2	8	0.6225	0.1748

Regarding the amount of patterns found in the simulation plant, we have the results presented in Table 3. At the end of the simulation with 3000 points of 41 sensor measurements, TEDA-Cloud and TEDA-Cloud modified achieved an amount close to K-means even executed online. TEDA-Cloud had a larger number of cores because its data stack was busier for real-time processing.

Table 3. Number of Cores using the $m = 1.25$.

Disturbance	**K**	**TEDA-Cloud**	**TEDA-Cloud modified**
Normal	14	14	14
Disturbance 6	12	20	12
Disturbance 8	26	30	26
Disturbance 13	65	85	70
Disturbance 14	120	208	138

Conclusions

This paper proposes a method of clustering for online data of sensor measurements in a simulated plant. The proposed method is the TEDA-Cloud modified, which is an improvement of the TEDA-Cloud with regard to the quality of clustering due to acceleration as the cores are formed and essentially reduced - merged. The results show that the same amount of cores are formed if the data were stored and executed by the K-means method using the artificial data. In addition, the method allows a smaller amount of information to be stored for online execution of clustering when compared to TEDA-Cloud.

***Authors**: M.Sc. Kennedy Reurison Lopes, e-mail:kenreurison@dca.ufrn.br; M.Sc. Ana Carla Costa Andrade, e-mail:acca.engpet@gmail.com; B.Sc. Bernardo Fonseca Andrade de Lima, e-mail:bernardo1411@hotmail.com; Ph.D. André Laurindo Maitelli, e-mail:maitelli@dca.ufrn.br; Laboratory of Petroleum Automation, Federal University of Rio Grande do Norte - UFRN, Natal-RN, Brazil.*

Funding:

This study was financed in part by the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Finance Code [001].

Acknowledgments

The authors thank the Coordination for the Improvement of Higher Education Personnel (CAPES), the financial support for the research, and the support of the Laboratory of Automation in Petroleum (LAUT/UFRN).

REFERENCES

[1] Angelov P., Gu Xiaowe., Gutierrez G., Iglesia J. A., Guedes L. A., Sanchis A.: Autonomous data density based clustering method, IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24-29 July, 2016.

[2] Bezerra C. G., Costa B. S. J., Guedes L. A., Angelov P. A.: A new evolving clustering algorithm for online data streams, IEEE Conference on Evolving and Adaptative Intelligent Systems (EAIS), pp. 162–168, Natal, 2016.

[3] Costa B. S. J., Bezerra C. G., Guedes L. A., Angelov P.: Online fault detection based on typicality and eccentricity data analytics, International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12-17 July, 2015.

[4] MacQueen J.: Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Calif., 1, pp. 281-297, 1967.

[5] Khanmohammadi S., Naiier Adibeig N., Samaneh Shanehbandy S.: An improved overlapping k-means clustering method for medical applications, Expert Systems With Applications, 67, pp. 12-18, Jan., 2017.

[6] Kamath R. S., Kamat R. K.: K-means clustering for analyzing productivity in light of R D spillover, International Journal of Information Technology, Modeling and Computing (IJITMC), 4, May, 2016.

[7] Punj G., Stewart D. W.: Cluster analysis in marketing research: review and suggestions for application, Journal of Marketing Research, 20, pp. 134-148, May, 1983.

[8] Alkilany A., Ahmed A., Said H., Bakar A. A.: Application of the k-means clustering algorithm to predict load shedding of the Southern Electrical Grid of Libya, Fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014), Luton, UK, 13-15 Aug., 2014.

[9] Zakharov K.: Application of k-means clustering in psychological studies, Tutorials in Quantitative Methods for Psychology, 12, pp. 87-100, 2016.

[10] Hartigan J. A.: Clustering algorithms, John Wiley Sons, Inc., 99th. ed. New York, NY, USA, 1975. ISBN ISBN: 047135645X

[11] Pietrzykowski M., Plucinski M.: Mini-model method based on k-means clustering, Przeglad Elektrotechniczny, pp. 73-76, Jan., 2017.

[12] Kangin D., Angelov P., Iglesias, J. A.: Autonomously evolving classifier TEDAClass, Information Sciences, 366, pp. 1-11, 2016.

[13] Angelov P.: Anomaly detection based on eccentricity analylis, IEEE Symposium on Evolving and Autonomous Learning Systems (EALS), Orlando, FL, USA, 9-12 Dec., 366, pp. 32-58, 2016.

[14] Bezerra C. G., Costa B. S. J., Guedes L. A., Angelov P.: An evolving approach to unsupervised and real-time fault detection in industrial processes, Expert systems with applications, 63, pp. 134-144, 2016.

[15] Angelov P.: Outside the box: An alternative data analitcs frame-work, Journal of automation, Mobile Robotics and Intelligent Systems, 8, pp. 53-59, 2014.

[16] Saw J. G., Yang M. C. K., Mo T. S. E. C.: Chebyshev inequality with estimated mean and variance, The American Statistician, 38, pp. 130-132, 1984.

[17] Downs J. J., Voguel E. F.: A plant-wide industrial process control problem, Computers Chemical Engineering, 17, pp. 142-149, June, 1993.

[18] Tyréus B. D., Voguel E. F.: Dominant variables for partial control. 2. Application to the Tennessee Eastman challenge process, Industrial & Engineering Chemistry Research, 38, pp. 1444-1455, 1999.

[19] Fränti P., Sieranoja S.: Clustering datasets, [web page] `http://cs.joensuufi/sipu/datasets/`. [Accessed on 10 July 2018.].