Merging Clusters in Summary Structures for Data Stream Mining based on Fuzzy Similarity Measures

Leonardo Schick[a], **Priscilla de Abreu Lopes**[b] **and Heloisa A. Camargo**[c]
[a]Department of Computer Science,UFSCar, São Carlos, SP, Brazil, leonardo.schick@dc.ufscar.br
[b]Itera, São Carlos, SP, Brazil, alopes.priscilla@gmail.com
[c]Department of Computer Science, UFSCar, São Carlos, SP, Brazil, heloisa@dc.ufscar.br

Abstract

Fuzzy Clustering is one of the mining techniques that have been used to extract information from Data Streams. The *d-FuzzStream* algorithm is a fuzzy version of the *Online-Offline Framework*, which consists of two steps: an online step, where a summary structure formed by fuzzy micro-clusters is built and an offline step, where the micro-clusters are clustered in batch mode. The quality of the data summary depends on the criteria used to decide whether an example starts a new micro-cluster or is absorbed by the existing ones; and whether two micro-clusters became similar enough to be merged. In *d-FuzzStream* algorithm such decisions are based on concepts of fuzzy dispersion and a distance-based fuzzy clusters similarity. In this paper we investigate the behavior of different fuzzy similarity measures on the decision of merging two fuzzy micro-clusters during the online step. Experiments were run using five synthetic data sets and four fuzzy similarity measures. The results obtained are analyzed and discussed through informative and purity measures.

Keywords: Data stream clustering, Data summary, Fuzzy similarity measures, Fuzzy clustering.

1 Introduction

The availability of data that are generated in form of streams is continuously increasing in several domains and stimulating research on methods dedicated to extract useful information from this source of data. Fuzzy Clustering is one of the mining techniques that have been used for mining Data Streams. In this context, methods must take into account particular features of the stream, such as the continuous arrival of large volumes of data and the possibility of change in data distribution. In our previous research we have developed an algorithm called *d-FuzzStream*[18], a fuzzy version of the *Online-Offline Framework*, which consists of two steps: an online step, where data instances are seen one by one as they arrive in the stream and a summary structure formed by Fuzzy Micro-Clusters (FMiC) is built to store statistical information on the data; an offline step, where the FMiCs are clustered in a user-required basis.

The quality of the data summary depends on the operations that are performed when each example arrives. The algorithm must decide whether a new incoming data is to be absorbed by the existing structure or carries enough novelty to start a new FMiC; and whether two FMiCs became similar enough to be merged. In *d-FuzzStream* algorithm such decisions are based on concepts of fuzzy dispersion and a distance-based fuzzy clusters similarity measure.

Ever since the advent of fuzzy sets, different measures to evaluate the similarity between fuzzy sets have appeared, under different sets of properties. These measures are defined based on operations on fuzzy sets [17] [7], distance measures between fuzzy sets or implication operators [21]. Applications of this measures include image processing [22], fuzzy reasoning [20] and pattern recognition [5]. A review of fuzzy similarity measures and applications to classification and recognition problems have been reported in [3].

In this paper we investigate the behaviour of four different fuzzy similarity measures when used to decide whether or not two FMiCs are to be merged during the maintenance of the summary structure in the online step. The main point to be considered when applying fuzzy similarity measures for the purpose exposed here is that, since the arriving examples themselves are not stored and the summary structure stores only some selected statistics, the similarity measure must be defined in such a way that it can be calculated incrementally, as each example arrives. Experiments were run using five synthetic data sets. The results obtained are used to analyze the impact of each measure in the quality of the summary structure based on the number of FMiCs creations, removals and merges as well as the purity measure.

This paper is organized as follows. In section 2 related work found in the literature is presented. In section 3, the *d-FuzzStream* algorithm is briefly reviewed. In section 4 the fuzzy similarity measures used in this study are presented and the design of experiments is described. Results are presented and discussed in section 5 and conclusions are addressed in section 6.

2 Related Work

Most fuzzy alternatives to Data Stream clustering are based on the *Single Pass Fuzzy C-Means* (SPFCM) [12]. This algorithm divides the data set into chunks and clusters each chunk in sequence using the *Weighted Fuzzy C-Means* algorithm (WFCM) [4]. The *weighted FCM - Adaptive Cluster* [15] and *Online Fuzzy C-Means* [11] are examples of algorithms based on this approach. A survey on fuzzy methods for data streams clustering can be found in [1].

The *Online-Offline Framework* (OOF) approach was originally proposed with the *CluStream* algorithm in [2]. OFF divides the learning process into two steps: online step (or data abstraction) and offline step (or clustering). While the online step continuously summarizes the data stream with the help of a summary structure, the offline step is initiated explicitly by the user, generating the data partition by clustering the summary structure.

The summary structure is composed of a set of *Micro-Clusters* (MiCs), which are cluster feature vectors composed by statistics that can be used to calculate the radius and the centroid of clusters, while also storing a timestamp to represent their time relevance. When the offline step is applied, the set of MiCs is converted into a set of weighted examples (one for each MiC), and is then clustered by a variant of the *k-means* algorithm, resulting in *Macro-Clusters* (MaC). Many algorithms are based on the OOF, employing different summary structures and different clustering techniques for both online and offline steps [6], [10], [8], [13].

FuzzStream [14] is, to the best of our knowledge, the first fuzzy clustering algorithm proposed as a fuzzy extension of the OOF, introducing concepts of the fuzzy set theory to all steps of the framework. This new proposal, called *Fuzzy Online-Offline Framework* (FOOF), is depicted in Figure 1. In *FuzzStream*, a set of *Fuzzy Micro-Clusters* (FMiCs) is maintained during the online step and clustered in the offline step using the WFCM algorithm.

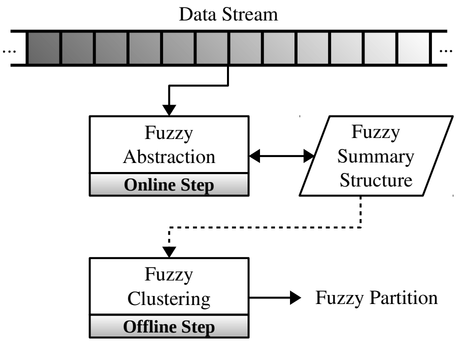

Figure 1: *Fuzzy Online-Offline Framework* [14].

A cluster feature has the properties of being incremental and additive. The incremental property allows a cluster feature to summarize an example by updating its statistics, while the additive property means that two cluster features can be merged by simply adding their statistics [19].

For every new Data Stream example, *FuzzStream* either assigns the example to the existing FMiCs or creates a new FMiC. In this work, we call the example assignment to the existing set of FMiCs as example absorption. The summary structure has a maximum size, and in case the structure reaches this size, the oldest FMiCs are deleted to give place to new ones. To minimize even further the size of this structure, after processing each new example, the algorithm tries to identify overlapping FMiCs (which may represent redundant information) and merge them.

When execution of the offline step is triggered, the set of FMiCs is converted into a set of weighted examples and clustered using WFCM.

Despite its robustness, the example absorption and FMiC merging rates in *FuzzStream* tend to be very low, which hinders the data summarization to retain much information about past examples. *d-FuzzStream* was developed as an improved version of *FuzzStream* aiming at avoiding the aforementioned problem.

3 Dispersion-Based Fuzzy Data Stream Clustering

The *d-FuzzStream* uses the fuzzy dispersion and fuzzy similarity measures to identify outliers and overlapping FMiCs. The FMiC structure is defined as the vector $(\overline{CF}, SSD, M, N, t)$, whose components are defined in Table 1.

$\overline{CF}$	linear sum of examples weighted by their membership to the FMiC
SSD	quadratic sum of distances between the examples and the FMiC prototype weighted by examples membership to the FMiC
N	number of examples assigned to the FMiC
t	timestamp of last example assigned to the FMiC
M	sum of memberships of the examples assigned to the FMiC

Table 1: FMiC Structure

The following sections detail the concepts of fuzzy dispersion and similarity-driven merging (3.1), the FMiC maintenance (3.2) and the data partition generation process (3.3).

3.1 Fuzzy Dispersion and Similarity-Driven Merging Criterion

The fuzzy dispersion of a cluster is a measure based on the *Root-Mean-Square Deviation* (RMSD) and can be used to represent the radius of a fuzzy cluster [24].

Let x_j be an example in the stream $(x_1, ..., x_n)$, $\{C_1, ..., C_k\}$ a set of k FMiCs and N_i the number of examples assigned to FMiC C_i. The fuzzy dispersion for cluster C_i ($disp_i$) is calculated as shown in (1) where $\mu(x_j)$ is the membership value of example x_j to cluster C_i, $m \in [0, \infty)$ is a fuzziness parameter, c_i is the prototype of C_i and $\|x_j - c_i\|$ represents the distance between example x_j and prototype c_i.

$$disp_i = \sqrt{\frac{\sum_{x_j \in C_i} \mu(x_j)^m \left\|x_j - c_i\right\|^2}{N_i}} \tag{1}$$

Since the statistics SSD and N are already stored in the FMiC, they can be used to calculate $disp_i$ as shown in (2).

$$disp_i = \sqrt{\frac{SSD_i}{N_i}} \tag{2}$$

Using the concept of fuzzy dispersion it is possible to evaluate whether an example x_j should be absorbed by the summary structure or initiate a new FMiC.

Additionally, the fuzzy dispersion is employed to calculate a *Fuzzy Cluster Similarity* matrix R [24], which can be used to represent the similarity between pairs of FMiCs. Each cell of matrix $R = \{R_{ij}, (i, j) = 1, 2, ..., k\}$ reflects the ratio of the sum of the fuzzy dispersion of two FMiCs to the distance between their prototypes as detailed in (3). Note that the principle behind the similarity measure R_{ij} is very similar to the concept of the clustering validation index *Xie-Beni*[23], with a slightly different form of calculation.

$$R_{ij} = \frac{disp_i + disp_j}{\|c_i - c_j\|} \tag{3}$$

The *Similarity-Driven Merging Criterion* [24] uses the R matrix and a threshold τ to identify overlapping clusters. If the R_{ij} value is greater than τ, the two FMiCs can be merged. If not, the two FMiCs are considered to be completely separated and not similar enough to be merged.

The threshold τ can assume values in the interval $[0, +\infty[$, where $\tau < 1$ considers non-overlapping FMiCs and $\tau >= 1$ considers only overlapping FMiCs. The greater the value of τ, the more overlapped the FMiCs must be for a merge to occur. On the other hand, the lower the value of τ, the less similar the FMiCs have to be for a merge to occur.

The two concepts defined in this section are used in the online step of the *d-FuzzStream* algorithm for identification of outliers and possible merges of overlapping FMiCs.

3.2 Online Step: FMiC Maintenance

The maintenance for a set of FMiC requires as entries the Data Stream, the fuzzification parameter for the FCM algorithm(m), the minimum and maximum number of FMiCs allowed in the structure (minMiC, maxMiC) and the merge threshold (τ).

The summary structure is initially empty. The first examples in the Data Stream are used to create FMiCs until the structure reaches the minimum number of FMiCs. If the summary structure already has the minimum number of FMiCs, the algorithm proceeds to evaluate whether the example is an outlier or not: the Euclidean distances between the example and all FMiCs prototypes, as well as the radius for each FMiC, are calculated.

When an example falls into a FMiC radius, the example is not considered an outlier. In such case, the structure must be updated to absorb the example. The memberships between the example and all FMiCs are calculated, like in the traditional FCM algorithm, and all FMiCs are updated. The timestamp for the FMiCs for which the example falls into their radius are also updated. If the example is an outlier, a new FMiC has to be created and added to the summary structure. In this case, if the structure is full, the oldest FMiC is replaced by the new one. This strategy provides a drift detection capability to the summary structure.

Finally, the merge step is initiated. The fuzzy similarity matrix R between all FMiCs is calculated and pairs of FMiCs with the highest fuzzy similarity among the ones with fuzzy similarity greater than the parameter τ are merged.

3.3 Offline Step: Weighted Fuzzy C-Means Clustering

The offline step is executed when required by the user. The set of FMiCs is turned into a set of weighted examples to be clustered in batch mode. For each FMiC in the summary structure, a prototype is obtained dividing $\overline{CF}$ by M and its weight is M itself. The weighted prototypes are then clustered using the WFCM algorithm [4] to generate the fuzzy partition. The initial clusters' prototypes are the examples with greatest weights.

Since the objective of this paper is to analyze the behavior of fuzzy similarity measures in the online step, the offline step will not be executed in the experiments.

4 Fuzzy Measures for Fuzzy Micro-clusters Merge

The objective of this study is to experiment different similarity measures between fuzzy clusters, besides the one used in the original implementation of *d-FuzzStream* described in Section 3, to decide whether they are to be merged or not. This strategy requires the definition of a threshold above which the merging is done. In the following subsections, the fuzzy similarity measures used are defined. From now on, the similarity measure used in the original version of *d-FuzzStream* will be denoted as S_1, that is, assuming that A and B are two FMiCs, $S_1(A,B) = R_{AB}$ where R_{AB} is defined in (3).

4.1 Fuzzy Similarity Measures

Two Fuzzy Similarity Measures (FSM) based on operations on fuzzy sets and two based on distance measures were selected. In all equations it is assumed that $\mu_A(x_i)$ and $\mu_B(x_i)$ denote the membership degree of element x_i in fuzzy sets A and B, respectively.

4.1.1 Measures Based on Operations on Fuzzy Sets

These measures are based on intersection and union operations and cardinality of fuzzy sets. S_2 was proposed in [17] and S_3 was proposed in [7].

$$S_2(A,B) = \frac{\sum_{i=1}^{n} min(\mu_A(x_i), \mu_B(x_i))}{\sum_{i=1}^{n} max(\mu_A(x_i), \mu_B(x_i))} \qquad (4)$$

$$S_3(A,B) = max_i(min(\mu_A(x_i), \mu_B(x_i))) \qquad (5)$$

4.1.2 Measures Based on Difference and Sums of Fuzzy Values

In the equations that follow, $|x|$ denotes the absolute value of x. Measure S_4 was proposed in [16] and measure S_5 was defined in [17].

$$S_4(A,B) = 1 - max_i(|\mu_A(x_i) - \mu_B(x_i)|) \qquad (6)$$

$$S_5(A,B) = 1 - \frac{\sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)|}{\sum_{i=1}^{n} |\mu_A(x_i) + \mu_B(x_i)|} \qquad (7)$$

4.2 Design of the Approach and Experiments

The data sets used in the experiments are described in Table 2. These are synthetic data sets with two attributes, generated using the *stream* and *streamMOA* packages in R and simulated as data streams. As long as the data sets have been created specifically to evaluate the method proposed, they have a clear predefined clustering structure. Additional information on these data is available at [9].

Identifier	# Instances	# Clusters	Noise?	Stationary?
BG_10k	10,000	4	No	Yes
Bench1_11k	11,000	2	Yes	No
Bench2_20k	20,000	2	No	No
RBF3_40k	40,000	7	Yes	No
RBF4_40k	40,000	7	Yes	No

Table 2: Data sets features

The online step of the algorithm was run using the parameters values $MinFMiC = 5$, $MaxFMiC = 100$ and $m = 2$ for all datasets. These values have been defined empirically, based on previous experiments.

In *d-FuzzStream* algorithm, the decision to merge two FMiCs is based on the parameter τ. When $R_{ij} > \tau$, FMiCs i and j are merged. In this study, two important changes are made. First, each one of the selected FSM are used to calculate the similarity matrix R. Then, a threshold σ is defined to decide when the FMiCs are to be merged. While τ varies in the range $[0, +\infty[$, the FSM varies in $[0,1]$. We ran three sets of experiments (Ex_1, Ex_2, Ex_3) varying the τ and σ values. The values for each experiment are shown in Table 3.

The evaluation measures were calculated at five different time moments, defined after the arrival of certain

Experiment	τ	σ
Ex_1	1	0.9
Ex_2	0.9	0.8
Ex_3	0.8	0.7

Table 3: Parameters value for experiments.

amount of examples, which we called windows. Each data set have its own window size, shown in Table 4. We refer to each evaluation moment as an observation.

Dataset	Window Size
BG_10k	2000
Bench1_11k	2200
Bench2_20k	4000
RBF3_40k	8000
RBF4_40k	8000

Table 4: Window Size for each data set

5 Results and Analysis

The results obtained were analysed by means of informative measures such as numbers of creations, eliminations, absorptions and merges of FMiCs, as well as the purity measure. It is important to recall that only the results of the online step are been evaluated. All three experiments demonstrated very similar tendencies in the results. Tables 5, 6 and 7 contain the mean purity value of all 5 observations for each data set in Ex_1, Ex_2 and Ex_3, respectively, with best values in bold. The values show that FSM S_3 obtained better results than the original measure S_1 and both, S_1 and S_3 generated higher values than the other three measures.

The numbers of creations, removals, absorptions and merges for each data set are depicted in Table 8. Since these values follow a very similar tendency through all three experiments, only the values for Ex_1 are shown. The values demonstrate that S_1 and S_3 lead to a much lower number of merges and absorptions and a larger number of creations and removals than the other measures, what makes the structure as a whole to change faster. This may explain the results concerning purity. With a low number of merges and absorptions, the structure tend to have a larger number of FMiCs and consequently a better purity value. This behavior of FSMs can be justified by the fact that S_3, as well as S_1 have a very simple form of calculation, involving the maximum degree of membership in the *min* intersection of FMiCs. This causes a low number of pairs of FMiCs to reach the similarity threshold to be merged. On the other side, the other three FSM S_2, S_4 and S_5 take into account the membership of all examples in the FMiCs, tending to increase the number of merges.

	S_1	S_2	S_3	S_4	S_5
BG	0.9943	0.7873	**0.9946**	0.7331	0.7878
Bench1	**0.9977**	0.9713	0.9975	0.9067	0.9720
Bench2	0.9839	0.9005	**0.9924**	0.8494	0.8913
RBF3	0.9929	0.7843	**0.9962**	0.7050	0.7972
RBF4	0.9829	0.7926	**0.9938**	0.6997	0.7549

Table 5: Purity - Experiment 1

	S_1	S_2	S_3	S_4	S_5
BG	0.9927	0.7619	**0.9947**	0.7352	0.7685
Bench1	0.9991	0.9408	**0.9993**	0.8930	0.9431
Bench2	0.9833	0.8639	**0.9940**	0.8265	0.8738
RBF3	0.9951	0.7517	**0.9962**	0.6876	0.7617
RBF4	0.9791	0.7549	**0.9940**	0.6686	0.7617

Table 6: Purity - Experiment 2

	S_1	S_2	S_3	S_4	S_5
BG	0.9929	0.7456	**0.9946**	0.7240	0.7535
Bench1	0.9964	0.9288	**0.9993**	0.8861	0.9280
Bench2	0.9760	0.8630	**0.9927**	0.8382	0.8768
RBF3	0.9910	0.7280	**0.9964**	0.6735	0.7424
RBF4	0.9789	0.7362	**0.9933**	0.6513	0.7360

Table 7: Purity - Experiment 3

The analysis can be confirmed by the Figures presented here. In Figures 2-5 and 7-10, which are plotted in the space of data attributes, the FMiCs are represented by circles of different colors, according to their real clusters and the light blue circles represent the weight of each FMiC. Comparing Figures 2 and 4, which show the FMiCs obtained by S_1 and S_4 in Observation 1 for data set BENCH1_11, one can see that the FMiCs generated by S_1 are more concentrated and similar in shape to the real data, which contains at this point two well separated clusters. On the other hand, FMiCs generated by S_4 have an elongated shape, because the old data remain in the structure for a longer time, due to the larger number of merges and absorptions and lower numbers of creations and removals. Figures 3 and 5, show the FMiCs in Observation 4, when the groups are very close to each other. S_1 still maintains a regular structure with a good density while S_4 gets confused, causing a decrease in the purity value (Figure 6).

		S_1	S_2	S_3	S_4	S_5
BG	**Creations**	8687	8193	8615	7828	8162
	Removals	6160	860	7811	248	907
	Absorptions	1313	1807	1385	2172	1838
	Merges	2427	7233	704	7483	7156
Bench1	**Creations**	8333	7901	8189	7598	8020
	Removals	3589	2005	6534	691	2125
	Absorptions	2667	3099	2811	3402	2980
	Merges	4676	5796	1555	6815	5800
Bench2	**Creations**	17334	16391	17230	15602	16419
	Removals	12030	2008	15797	521	2246
	Absorptions	2666	3609	2770	4398	3581
	Merges	5204	14291	1334	14988	14078
RBF3	**Creations**	33311	31900	33012	31370	31851
	Removals	20392	4577	28616	1687	4987
	Absorptions	6689	8100	6988	8630	8149
	Merges	12819	27224	4296	29587	26767
RBF4	**Creations**	32924	32100	32863	31162	31931
	Removals	19487	4744	28455	1598	5175
	Absorptions	7076	7900	7137	8838	8069
	Merges	13337	27263	4308	29467	26659

Table 8: Informative measures - Experiment 1

Figures 7 and 9 show the FMiCs generated by S_2

and S_3, respectively, in Observation 2 for data set RBF4_40k. This data set contains groups that change very fast, and some groups can appear and disappear along time. In Observation 2, there are four groups. In a visual analysis, it is possible to note that the structure generated by S_3 is able to identify the four groups, while the one generated by S_2 presents a higher mixture (overlapping) of FMiCs. As shown in Figure 11, the purity value for S_2 is good, even though lower than the one for S_3. Figures 8 and 10 illustrate the summary structure at Observation 3 for the same data set. At this moment, a new group starts to appear. While S_3 generated a structure that reflects the real structure of data with five groups, S_2 generated a structure with a visible larger overlapping among FMiCs. This situation provokes a large decrease in the purity value for S_2 (Figure 11), while the value for S_3 remains high.

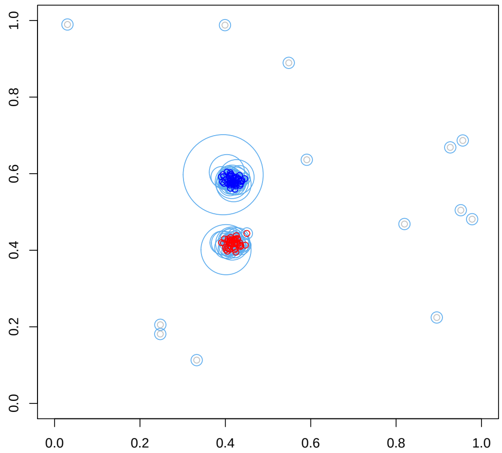

Figure 2: BENCH1_11k, Measure S_1, Observation 1

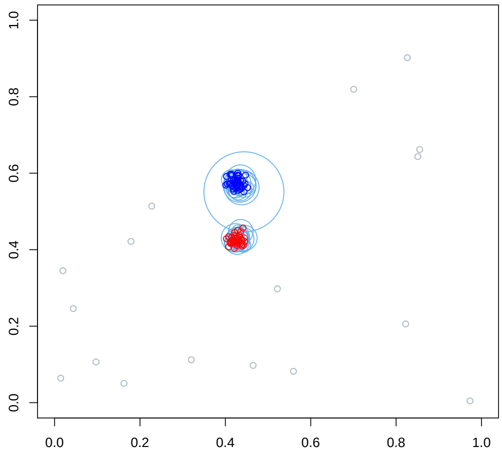

Figure 3: BENCH1_11k, Measure S_1, Observation 4

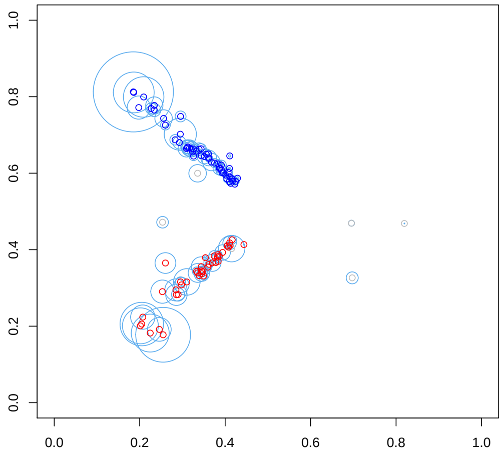

Figure 4: BENCH1_11k, Measure S_4, Observation 1

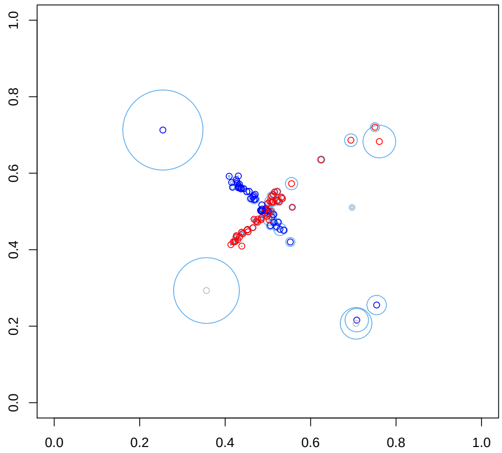

Figure 5: BENCH1_11k, Measure S_4, Observation 4

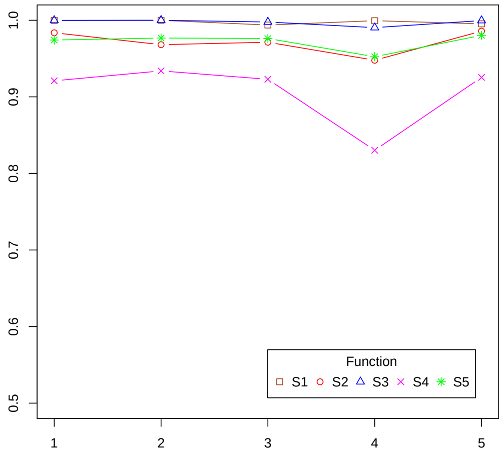

Figure 6: BENCH1_11k - Purity for each Observation

6 Conclusions

The work presented here was designed to explore the use of different fuzzy similarity measures in the merg-

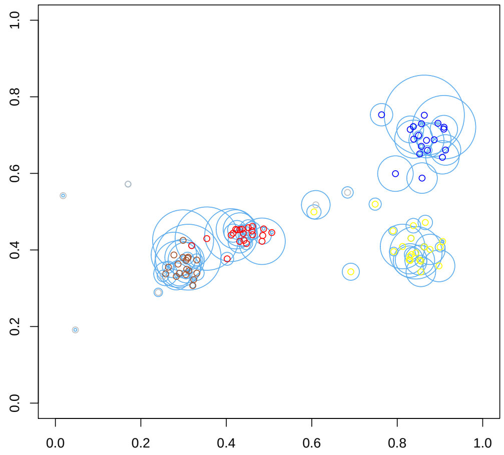

Figure 7: RBF4_40k,Measure S_2, Observation 2

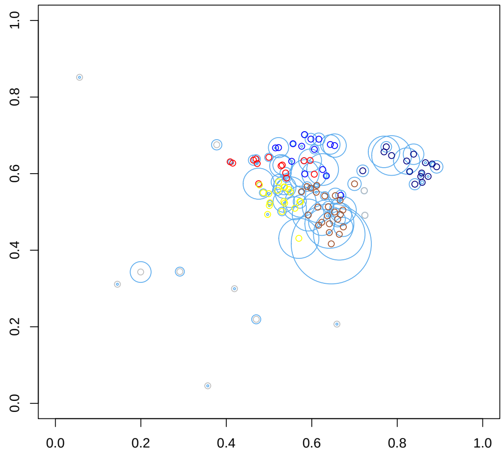

Figure 8: RBF4_40k,Measure S_2, Observation 3

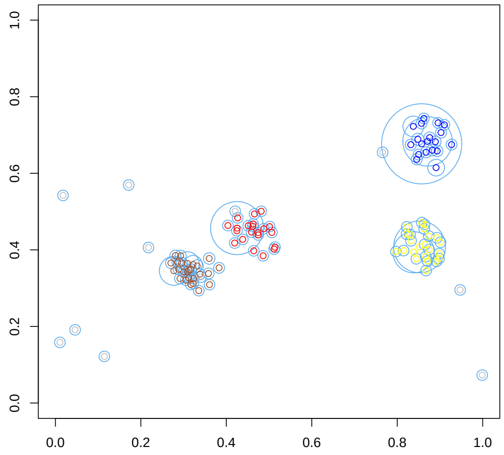

Figure 9: RBF4_40k,Measure S_3, Observation 2

ing operation of the online step of *d-FuzzStream* algorithm. The fuzzy measures were selected based on the

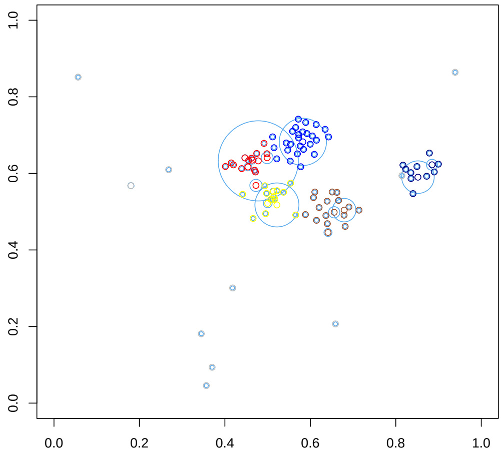

Figure 10: RBF4_40k,Measure S_3, Observation 3

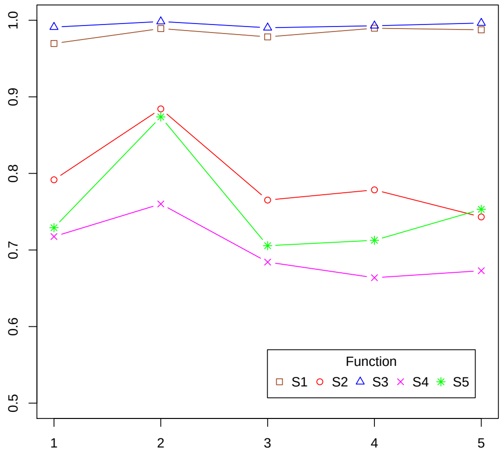

Figure 11: RBF4_40k - Purity for each Observation

incremental and additive properties, due to the form of calculation involved. This way, the fuzzy similarity matrix between every pair of FMiCs can be easily updated when two of them are merged. The results obtained evidenced that FSM S_1 and S_3 obtained similar behavior in all evaluations with a larger number of creations and removals and a lower number of mergers and absorptions when compared to the other three FSM, S_2, S_4 and S_5. These last three measures, on the other hand, have demonstrated to be more sensitive to the fuzziness of the set of data. In our future work we plan to evaluate the performance of the method on real data sets as well as to investigate, in a deeper way, the causes for the different results obtained, including different internal and external fuzzy clustering measures.

Acknowledgement

The authors are grateful to the Brazilian agencies CAPES, CNPq and FAPESP for the support.

References

[1] A. Abdullatif, F. Masulli, S. Rovetta, Clustering of nonstationary data streams: A survey of fuzzy partitional methods, WIREs Data Mining and Knowl. Discov. (8) (2018) 1–18.

[2] C. C. Aggarwal, J. Han, J. Wang, P. S. Yu, A framework for clustering evolving data streams, in: Proceedings of the 29th international conference on Very large data bases-Volume 29, VLDB Endowment, 2003, pp. 81–92.

[3] L. Baccour, A. M. Alimi, R. I. John, Some notes on fuzzy similarity measures and application to classification of shapes, recognition of arabic sentences and mosaic, AIENG International Journal of Comp. Sci. 41 (2014) 1–10.

[4] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer US, 1981.

[5] R. Buse, Z. Liu, J. Bezdek, Word recognition using fuzzy logic, IEEE Trans. on Fuzzy Systems 10 (2002) 65–76.

[6] F. Cao, M. Ester, W. Qian, A. Zhou, Density-Based Clustering over an Evolving Data Stream with Noise, in: Proceedings of the 6th SIAM International Conference on Data Mining, 2006, pp. 328–339.

[7] S. M. Chen, M. S. Yeh, P. Y. Hsiao, A comparison of similarity measures of fuzzy values, Fuzzy Sets and Systems 72 (1995) 79–89.

[8] Y. Chen, L. Tu, Density-based clustering for real-time stream data, in: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Jose, Califórnia, EUA, 2007, pp. 133–142.

[9] Computational Intelligence Group (CIG), Data stream repository, Department of Computing, Universidade Federal de São Carlos - UFSCar, 2017. [Online]. Available: http://github.com/CIG-UFSCar/DS_Datasets

[10] M. Ester, H.-P. Kriegel, J. Sander, 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, AAAI Press, 1996, pp. 226–231.

[11] P. Hore, L. Hall, D. Goldgof, W. Cheng, Online fuzzy c means, in: NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society, IEEE, 2008, pp. 1–5.

[12] P. Hore, L. O. Hall, D. B. Goldgof, Single Pass Fuzzy C Means, in: 2007 IEEE International Fuzzy Systems Conference, IEEE, 2007, pp. 1–7.

[13] P. Kranen, I. Assent, C. Baldauf, T. Seidl, The ClusTree: Indexing Micro-clusters for Anytime Stream Mining, Knowl. Inf. Syst. 29 (2) (2011) 249–272.

[14] P. A. Lopes, H. A. Camargo, Fuzzstream: Fuzzy data stream clustering based on the online-offline framework, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[15] S. Mostafavi, A. Amiri, Extending fuzzy c-means to clustering data streams, in: 20th Iranian Conference on Electrical Engineering (ICEE2012), IEEE, 2012, pp. 726–729.

[16] C. P. Pappis, Value approximate of fuzzy systems variables, Fuzzy Sets and Systems 39 (1991) 111–115.

[17] C. P. Pappis, I. Karacapilidis, A comparative assessment of measures of similarity of fuzzy values, Fuzzy Sets and Systems 56 (1993) 171–174.

[18] L. Schick, P. Lopes, H. Camargo, *d-FuzzStream*: A dispersion-based fuzzy data stream clustering, in: Proc. Intern. Conference on Fuzzy Systems (FuzzIEEE'18), Vol. 1, Rio de Janeiro, Brazil, 2018, pp. 135–142.

[19] J. A. Silva, E. R. Faria, R. Barros, E. R. Hruschka, A. P. L. Carvalho, J. Gama, Data stream clustering, ACM Computing Surveys 46 (1) (2013) 1–31.

[20] D.-G. Wang, Y.-P. Meng, H.-X. Li, A fuzzy similarity inference method for fuzzy reasoning, Computers and Mathematics with Applications 56 (2008) 2445–2454.

[21] X. Z. Wang, B. D. Baets, E. Kerre, A comparative study of similarity measures, Fuzzy Sets and Systems 73 (1995) 259–268.

[22] D. V. Weken, M. Nachtegael, E. Kerre, Using similarity measures and homogeneity for the comparison of images, Image Vision Computing 22 (2004) 695–702.

[23] X. L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pat.An.Mach.Int. 13 (1991) 841–847.

[24] X. Xiong, K. L. Chan, K. L. Tan, Similarity-driven cluster merging method for unsupervised fuzzy clustering, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2004, pp. 611–618.