Agrupamento Fuzzy para Fluxo Contínuo de Dados – Um Estudo de Algoritmos Baseados em Blocos

R. K. Asbahr[1], P. A. Lopes[2], H. A. Camargo[1]

[1] Universidade Federal de São Carlos
rodolfo.kasbahr@gmail.com
heloisa@dc.ufscar.br
[2] Itera
alopes.priscilla@gmail.com

Abstract. Data Stream Mining (DSM) has become an important topic due to the increasing availability of large collections of data. These data sets are characterized by having potentially infinite size, which prevents them from being stored in their entirety, and can generate examples with changeable statistical distribution according to time. These characteristics impose the need to create and use appropriate algorithms. Clustering algorithms are appropriate for DSMs where the labeling of the examples is costly and time consuming. Fuzzy clustering algorithms present an additional benefit in these contexts by allowing decision surfaces to be defined flexibly. The objective of this work was to implement and analyze the behavior of chunk based fuzzy clustering algorithms for DSM. The experiments, using two synthetic datasets and one real data set, allow us to extract analyzes regarding trends in the behavior of the algorithms according to their abilities to treat two critical problems for this type of algorithm: change in the distribution of the data and definition of the number of groups.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: data stream mining, fuzzy clustering, concept drift, machine learning

1. INTRODUÇÃO

Com a constante redução de custos das tecnologias observada atualmente, as fontes de dados crescem em volume, velocidade e tornam-se contínuas ao longo do tempo. Essa realidade despertou o interesse da comunidade científica para a necessidade de extrair conhecimento dessas fontes de dados, consolidando o tema conhecido como mineração em Fluxo Contínuo de Dados (FCD). Devido ao grande volume e a grande velocidade com que são gerados, tais conjuntos de dados não podem ser armazenados em sua totalidade. Além disso, frequentemente, os dados gerados por uma mesma fonte apresentam variações na sua distribuição [Gama 2010]. Essas características, próprias do contexto de FCD, impõem a necessidade de criação e uso de algoritmos de extração de conhecimento capazes de tratar tanto a impossibilidade de armazenamento do conjunto completo quanto a possibilidade de mudanças nas tendências apresentadas por esses dados.

O Aprendizado de Máquina (AM) é a subárea da Inteligência Artificial que se refere à investigação de métodos computacionais capazes de adquirir conhecimento de forma automática [Mitchell 1997]. A maioria dos algoritmos mais tradicionais de AM, no entanto, considera que o conjunto total de dados está disponível e pode ser acessado a qualquer momento. Para fazer a extração de conhecimento útil em ambientes dinâmicos, métodos de AM devem ser adaptados para considerar novos dados de forma contínua. Dentro da área de FCD é comum verificar a falta de informação de classe, seja por conta da natureza do domínio ou pela dificuldade em rotular exemplos. Nesse caso, são necessárias abordagens utilizadas no aprendizado não supervisionado, entre as quais as mais investigadas são as de agrupamento de dados [Silva et al. 2013]. Algoritmos de agrupamento

para FCD são, em geral, propostos como extensões dos algoritmos de agrupamento conhecidos.

O objetivo deste trabalho é implementar e avaliar algoritmos de agrupamento fuzzy para FCD baseados na abordagem de blocos de dados (chunks) para gerar análises demonstrativas do comportamento dos algoritmos. Os algoritmos estudados foram escolhidos com foco em dois desafios relacionados a esse contexto: a resposta a mudanças que podem ocorrer ao longo do FCD e a definição do número de grupos do agrupamento. Um algoritmo de agrupamento para FCD deve, não apenas sumarizar os dados vistos até o momento, mas detectar mudanças que ocorram na distribuição dos dados ao longo do tempo. Já a questão da definição prévia do número de grupos é inerente aos algoritmos de agrupamento particionais e, devido à sua influência nos resultados obtidos, deve ser tratada durante o processo. Assim, foram selecionados para as análises comparativas: um algoritmo que não possui mecanismo de captação de mudanças [Hore and Hall 2007b]; um algoritmo que altera o número de blocos considerados nos agrupamento dos blocos seguintes [Hore and Hall 2007a]; um algoritmo que possui um fator de decaimento, o qual define a taxa de esquecimento dos dados antigos [Jaworski et al. 2012]; um algoritmo que apresenta uma proposta simples de definição dinâmica do número de grupos [Mostafavi 2012]. Os algoritmos foram implementados em R e os experimentos foram executados com dois conjuntos de dados sintéticos e um conjunto de dados reais.

Este artigo está organizado da seguinte forma. Na seção 2 são apresentados alguns trabalhos representativos para contextualizar a proposta apresentada aqui. Na seção 3, os algoritmos selecionados para estudo são apresentados resumidamente. Os experimentos e análises são descritos na seção 4 e as conclusões e trabalhos futuros são abordados na seção 5.

2. TRABALHOS RELACIONADOS

Desde a última década, surgem cada vez mais métodos diferentes que aplicam processo de aprendizagem em FCD [Gama 2012]. Neste trabalho são abordados, especificamente, os algoritmos de agrupamento. Métodos e algoritmos de agrupamento [Jain and Murty 1999] são ferramentas de análise de dados eficazes e, sendo assim, constituem um importante ramo da mineração de dados [Witten et al. 2017]. Os algoritmos de agrupamento fuzzy são aqueles que permitem que um objeto pertença a mais de um grupo com graus diferentes. Essa possibilidade de modelar as fronteiras entre grupos de forma gradual e imprecisa, ao invés de rígida, oferece meios mais adequados para tratar diversos problemas reais [Bezdek 1981].

Algoritmos de agrupamento para FCD são, em geral, propostos como extensões dos algoritmos de agrupamento conhecidos. A maior parte dos trabalhos encontrados atualmente está fundamentada em algoritmos particionais, especificamente o algoritmo K-Means[Macqueen 1967]. Entre esses, é possível identificar duas categorias gerais de trabalhos: abordagens baseadas em framework on-line/off-line e baseadas em blocos de dados (chunks). As abordagens baseadas no framework on-line/off-line (FOO) possuem duas fases: sumarização, ou fase on-line, e agrupamento, ou fase off-line. O FOO foi inicialmente proposto em [Aggarwal et al. 2003], com o algoritmo CluStream. Entre os vários algoritmos propostos posteriormente, que adotam a abordagem FOO, destaca-se o algoritmo ClusTree[Kranen et al. 2011]. Nas abordagens baseadas em blocos de dados, os dados do fluxo são separados em blocos à medida que são gerados e um algoritmo de agrupamento é aplicado separadamente a cada bloco. Os centros de clusters obtidos em um bloco são utilizados nos blocos seguintes como uma forma de manter o histórico dos dados mais antigos.

Encontra-se também, na literatura, propostas para explorar a flexibilidade do agrupamento fuzzy de dados em FCD, sendo a maioria delas, variantes do algoritmo Fuzzy C-Means (FCM)[Bezdek 1981]. Em [Hore and Hall 2007a], foi proposto um algoritmo chamado Stream FCM (SFCM), variante do FCM para FCD que agrupa os blocos de dados e, a cada agrupamento, mantém os centros de grupo ponderados, descarta os dados do bloco e une os centros ao próximo bloco para serem agrupados em conjunto, como uma forma de manter a história dos blocos anteriores. Essa proposta modifica uma versão anterior do algoritmo chamado Single Pass FCM (SPFCM), dos mesmos autores [Hore and Hall 2007b], a qual foi projetada visando permitir a escalabilidade do algoritmo FCM para conjuntos de dados grandes. Ambos os trabalhos utilizam o agrupamento fuzzy ponderado (Weighted FCM – wFCM) proposto em [Hore and Hall 2007b]. Em [Hore et al. 2008] foi proposto ainda

outro algoritmo chamado Online FCM (OFCM) que também pode ser usado para agrupar FCD. O trabalho apresentado em [Li et al. 2016] propõe duas versões de um algoritmo para FCD que utiliza um FCM ponderado de uma forma diferente daquela do wFCM, com base em densidade. Em [Jaworski et al. 2012] os autores estendem o trabalho proposto em [Hore and Hall 2007a] com a inclusão de um fator de decaimento no cálculo dos pesos dos dados, que reflete a velocidade de esquecimento dos pesos de dados antigos reduzindo sua influência no resultado do agrupamento de dados mais recentes. O mesmo mecanismo é aplicado também como uma variante do algoritmo fuzzy possibilístico (Possibilistic C-Means - PCM)[Krishnapuram 1993]. No trabalho descrito em [Mostafavi 2012], é apresentada uma estratégia simples, com base no FCM, que determina o número de grupos dinamicamente. Na próxima Seção serão descritos os algoritmos selecionados para o estudo apresentado neste artigo.

3. ALGORITMOS

Nesta Seção serão apresentados os algoritmos analisados, destacando suas principais características. Inicialmente será apresentado o algoritmo Weighted Fuzzy C Means (WFCM)[Hore and Hall 2007b], que é a base de todos os algoritmos estudados.

3.1 Weighted Fuzzy C Means

O algoritmo WFCM [Hore and Hall 2007b] foi proposto como uma variante do Fuzzy C Means(FCM), para situações em que o conjunto de dados não pode ser armazenado em memória. A principal diferença com relação ao FCM é que os centroides de grupo possuem pesos que representam, de forma sumarizada, os dados que pertencem a um grupo. O algoritmo é aplicado em blocos de dados e, após cada execução, os centroides têm seus pesos calculados e são adicionados ao próximo bloco para serem agrupados na prÃşxima iterção. Inicialmente, todos os dados têm peso igual a 1. A cada bloco recebido, o WFCM gera uma matriz de pertinência com valores aleatórios e, com os dados do bloco e a matriz gerada, define os centroides iniciais usando a equação 1 Nessa equação, c_i são os centroides do algoritmo, k sendo o número de centroides, e_j são os dados do bloco, w_j são os pesos dos dados do bloco, u_{ij} são os elementos da matriz de pertinência, m é a constante de fuzzificação e n é o número de dados contidos no bloco.

$$c_i = \frac{\sum_{j=1}^{n} w_j u_{ij}^m e_j}{\sum_{j=1}^{n} w_j u_{ij}^m}, i = 1, ..., k. \tag{1}$$

Após esse passo inicial, o algoritmo atualiza a matriz de pertinência, usando a equação 2 e, em seguida, atualiza os centroides usando a equação 1. Esses dois passos se repetem até que a condição de parada seja satisfeita.

$$u_{ij} = \left[\sum_{l=1}^{k} \left(\frac{||e_j - c_i||}{||e_j - c_l||} \right) \frac{2}{m-1} \right]^{-1} \tag{2}$$

O peso dos centros é calculado de acordo com a equação 3.

$$w_i = \sum_{j=1}^{n} u_{ij} w_j, i = 1, ..., k. \tag{3}$$

3.2 Online Fuzzy C Means

O algoritmo Online Fuzzy C Means (OFCM) [Hore et al. 2008] foi originalmente proposto para agrupar conjuntos de dados grandes, porém com tamanho definido. Este algoritmo separa os dados do conjunto de dados original em vários bloco e agrupa cada deles um separadamente, usando o Weighted Fuzzy C Means. Dessa forma, cada agrupamento gera um determinado número de centroides com seus respectivos pesos. O

próximo passo do algoritmo é considerar todos os centroides de todos os grupos como dados ponderados e agrupá-los novamente, obtendo assim o agrupamento final.

Neste trabalho, foi desenvolvida uma adaptação do algoritmo para fluxo contínuo de dados. Como no contexto de fluxo contínuo de dados os dados não podem ser armazenados na sua totalidade, a adaptação consiste em armazenar os dados que chemgam em um bloco até que fique cheio e agrupar os dados desse bloco, apenas. Esse processo se repete para um número de blocos e, de tempos em tempos, os centroides resultantes desses agrupamentos são agrupados novamente.

3.3 Single Pass Fuzzy C Means

O algoritmo Single Pass Fuzzy C Means (SPFCM), proposto em [Hore and Hall 2007b] agrupa blocos de dados um a um. Cada bloco é agrupado usando o WFCM e, em seguida, os centroides gerados são adicionados ao bloco seguinte e utilizados como centroides iniciais no próximo agrupamento. Os pesos dos centroides do agrupamento anterior não são considerados no cálculo dos pesos dos centroides do agrupamento atual.

Neste trabalho foi utilizada a versão estendida do SPFCM, apresentada em [Hore and Hall 2007a], que consiste em adicionar o conceito de tamanho de histórico variável ao algoritmo, ou seja, o número de blocos precedentes considerados em cada novo agrupamento, podem ser considerados centroides ainda mais antigos. Ao definir o número de blocos anteriores que vão contribuir com seus centroides para o próximo agrupamento, define-se também uma maior ou menor influência dos dados mais antigos nesse agrupamento.

3.4 Weighted Fuzzy C Means com desvio de conceito

Como os dados do fluxo são infinitos, eles apresentam evolução com o passar do tempo, mas essas mudanças podem ser bem sutis em um período pequeno de tempo [Jaworski et al. 2012].

O algoritmo WFCM por si só não é capaz de detectar e tratar a evolução dos dados, podendo considerar que a diferença no padrão dos dados seja um outlier (informação atípica, que não corresponde ao padrão das demais informações). O algoritmo WFCM considera que os pesos de todos os dados provenientes do fluxo tem sempre peso 1 como definido na equação 4, sendo w_j^p o peso de um dado j do bloco p e n_p o número de exemplos do bloco p.

$$w_j^p = 1, p \geq 1, j = 1, \ldots, n_p \tag{4}$$

Dessa forma, é possível observar que todos os dados do fluxo têm mesma importância dentro do agrupamento, mesmo os dados mais antigos que já não necessariamente representam o comportamento padrão do fluxo. Para que o WFCM consiga tratar evolução dos dados, foi proposto em [Jaworski et al. 2012] uma forma de ponderar os dados usando um fator de decaimento que aumenta o peso dos dados mais recentes, diminuindo, assim a influência dos dados antigos no agrupamento atual.

A solução proposta foi de atribuir aos dados um peso maior com base no peso dos dados passados, como mostra a equação 5. É possível observar que essa solução aumenta o peso dos dados progressivamente conforme eles são captados pelo algoritmo. Dessa forma os dados mais antigos vão perdendo peso por ter peso menor que o dos dados mais novos.

$$w_{j+1}^p = w_j^p 2^{\lambda}, w_1^p = 1, p \geq 1, j = 1, \ldots, n_p - 1 \tag{5}$$

Na equação 5, $\lambda > 0$ é o fator de decaimento, cujo valor reflete a velocidade de esquecimento da influência dos dados antigos nos resultados do agrupamento. Note que se $\lambda = 0$, a equação 5 se torna equivalente à equação 4. O WFCM-DC integra os centroides do agrupamento anterior ao agrupamento presente, para que os dados antigos tenham influência no cálculo dos novos centroides.

3.5 Weighted Fuzzy C Means - Adaptive Cluster number

A definição prévia do número de grupos a serem descobertos é uma questão crucial para os algoritmos de agrupamento. No FCD, em particular, essa questão se torna ainda mais crítica, uma vez que, com a possível mudança na distribuição dos dados, é possível que também mude o número de grupos. Com o objetivo e adaptar o número de grupos às variações do fluxo de dados, foi proposta em [Mostafavi 2012] uma abordagem para definir dinamicamente qual o melhor número de grupos a cada agrupamento chamada Weighted Fuzzy C Means - Adaptive Cluster number (WFCM-AC). A estratégia utilizada consiste em agrupar cada bloco, usando WFCM, com k, k-1 e k+1 grupos e selecionar aquele agrupamento que apresenta o melhor resultado com base na medida Xie-Beni.

4. ANÁLISE DE RESULTADOS

Os conjuntos de dados utilizados com suas características são apresentados na Tabela 1.

Nome	Instâncias	Atributos	Classes	Estacionário
BarsGaussAN0_10000	10000	3	3	Sim
Benchmark2_10000	10000	3	3	Não
KDD-Cup`99 (10%)	494021	41	5	Não

Tabela 1. Conjuntos usados para o agrupamento e informações sobre eles.

Os conjuntos BarsGaussAN0_10000 e Benchmark2_10000 são conjuntos sintéticos retirados do repositório do Computational Inteligence Group(CIG) [Group 2017]. Como descrito na tabela, os conjuntos KDD-Cup'99 e Benchmark2_10000 não são estacionários, ou seja, os dados deles evoluem com o tempo. Os atributos nominais do KDD-Cup'99 foram desconsiderados para o agrupamento. Todos os algoritmos foram executados com o número de grupos igual a 3. Para o algoritmo SPFCM, foi definido o tamanho de histórico (número de blocos que contribuem com centroides para o agrupamento atual) igual a 5.

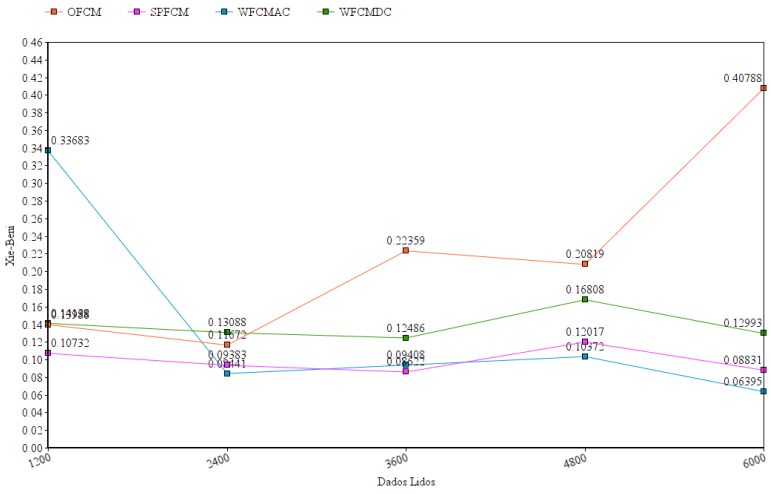

Fig. 1. Medidas Xie-Beni para todos os algoritmos com o conjunto BarsGaussAN0_10000.

A avaliação e comparação dos algoritmos será feita através da medida Xie-Beni e do tempo de execução dos algoritmos. A medida Xie-Beni [Xie 1991] consiste em calcular o quociente entre o erro quadrático médio

de todos os pontos em relação a um centro e a distância média entre os centroides dos clusters. Para ter uma medida ótima, é necessário que o erro seja pequeno e que a distância entre os centros seja grande, o que significa que os pontos que têm maior pertinência ao grupo estão próximos entre si e distantes de pontos com menor pertinência. Os valores ótimos são os mais próximos de zero. SerÃą usada a medida implementada pela biblioteca fclust, uma biblioteca da linguagem R que que contém o cálculo da medida Xie-Beni

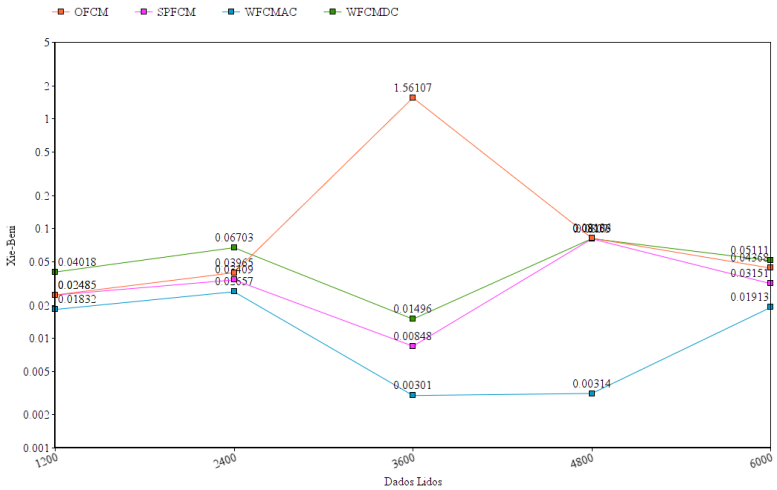

Fig. 2. Medidas Xie-Beni para todos os algoritmos com o conjunto KDD-Cup'99.

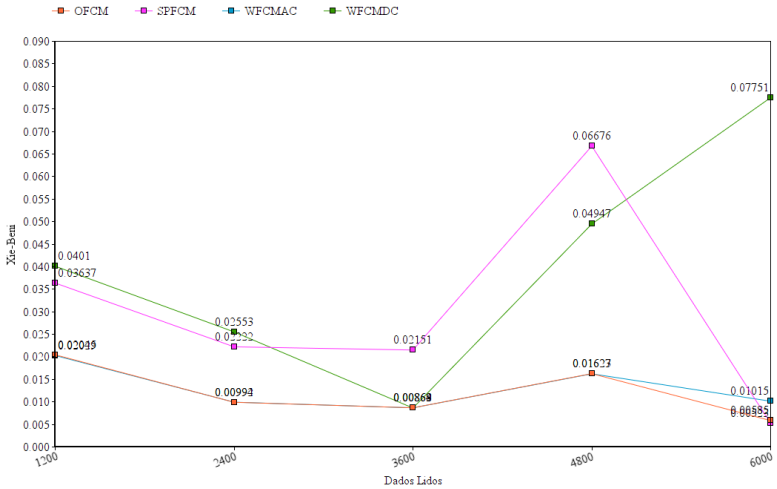

Fig. 3. Medidas Xie-Beni para todos os algoritmos com o conjunto Benchmark2_10000.

As figuras 1, 2 e 3 mostram os valores da medida Xie-Beni para cada um dos conjuntos de dados, calculados a cada bloco de 1200 dados, até o total de 6000 dados. Analisando as figuras 1, 2 e 3, nota-se que o algoritmo WFCM-AC mostrou um comportamento estável, obtendo as menores (melhores) medidas Xie-Beni para os três conjuntos de dados. Sendo esse um algoritmo que ajusta dinamicamente o número de grupos, tal resultado confirma a importância da escolha do número para agrupar FCD.

O OFCM teve medidas baixas(boas) para apenas o conjunto Benchmark2_10000, que é um conjunto que há pouco ruído e é focado apenas na evolução dos dados. Apesar de apresentar desempenho satisfório, a medida não mostra se o algoritmo consegue acompanhar a evolução dos dados, tendo em vista que o algoritmo analisa o comportamento dos centroides.

O algoritmo SPFCM apresentou bons resultados com relação à medida Xie-Beni para os conjuntos de dados BarsGaussAN0_10000 e KDD-Cup'99, comparáveis aos obtidos pelo WFCM-AC. Entretanto seu desempenho decaiu no conjunto de dados Benchmark2_10000, o que pode ser explicado pelo fato desse conjunto apresentar mudanças na distribuição dos dados, situação para a qual o SPFCM não provê tratamento especial.

O algoritmo WFCM-DC teve comportamento similar ao do SPFCM nos dois primeiro conjuntos de dados (figuras 1 e 2). No conjunto Benchmark2_10000, apresentou melhoras sucessivas à medida que os dados foram tratados, apresentando melhores índices que o SPFCM, exceto na última avaliação (6000 dados). Essa evolução evidencia que WFCM-DC foi capaz de captar as alterações na distribuição dos dados, o que não ocorreu com o SPFCM.

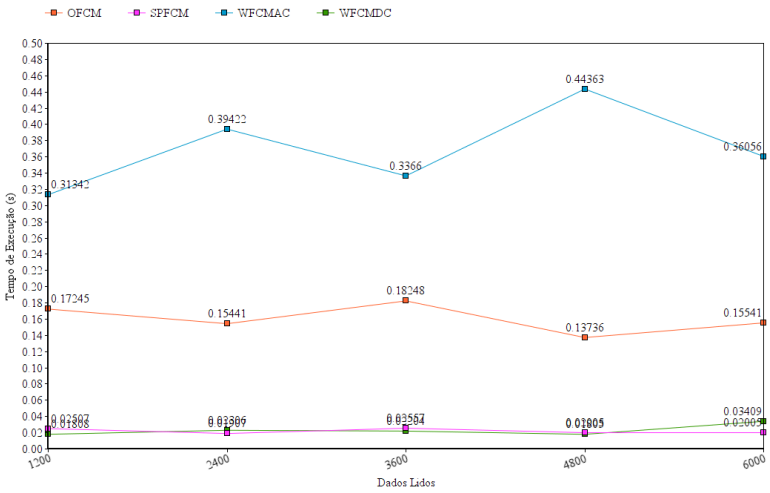

Fig. 4. Tempo de execução para todos os algoritmos com o conjunto Benchmark2_10000.

Com relação a tempo de execução, analisando a figura 4, nota-se que os algoritmos SPFCM e WFCM-DC foram os mais rápidos com uma diferença significativa em relação aos outros dois algoritmos. Esse resultado era esperado, já que SPFCM e WFCM-DC fazem um agrupamento a cada bloco de dados enquanto os algoritmos OFCM e WFCM-AC precisam fazer mais de um agrupamento por iteração. O tempo de execução dos algoritmos OFCM e WFCM-AC é de alguns segundos, ao passo que os algoritmos SPFCM e WFCM-DC requerem apenas alguns milésimos de segundos para terminar sua execução.

Enfim, é possível observar que cada algoritmo apresenta vantagens e desvantagens. Os algoritmos OFCM e WFCM-AC são capazes de obter precisões de agrupamento maiores que os algoritmos SPFCM e WFCM-DC,

que, por sua vez, são mais rápidos.

5. CONCLUSÃO

Neste artigo, foram apresentados e discutidos os resultados da implementação e execução de algoritmos de agrupamento fuzzy baseados em blocos para FCD. Os algoritmos selecionados possuem diferentes características relacionadas à identificação de mudanças na distribuição dos dados e definição no número de grupos. Os experimentos, executados com três conjuntos de dados, permitiram extrair conclusões sobre o comportamento dos algoritmos utilizados. Espera-se, como contribuição deste trabalho, que as implementações dos algoritmos e as análises realizadas possam ser utilizadas como embasamento para pesquisas futuras do nosso grupo de pesquisa e da comunidade de inteligência computacional. Em continuidade ao estudo apresentado neste artigo, a próxima etapa prevista será a utilização de outras medidas de avaliação de agrupamento e a expansão de experimentos com outros conjuntos de dados, visando consolidar as conclusões sobre o comportamento dos algoritmos.

REFERENCES

AGGARWAL, C. C., HAN, J., AND WANG, J. & YU, P. S. A framework for clustering evolving data streams. *In Proceedings of the 29th International Conference on Very Large Data bases.* vol. 29, pp. 81âĂŞ92, 2003.

BEZDEK, J. C. Pattern recognition with fuzzy objective function algorithms. https://doi.org/10.1007/978-1-4757-0450-1, 1981.

GAMA, J. Knowledge discovery from data streams, 2010. Chapman and Hall.

GAMA, J. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence.* 1 (1): 45âĂŞ55, 2012. https://doi.org/10.1007/s13748-011-0002-6.

GROUP, C. I. Data stream repository. http://github.com/CIG-UFSCar/DS Datas, 2017.

HORE, P. AND HALL, L. O. & GOLDGOF, D. B. A fuzzy c means variant for clustering evolving data streams. *In 2007 IEEE International Conference on Systems, Man and Cybernetics*, 2007a. https://doi.org/10.1109/ICSMC.2007.4413710.

HORE, P. AND HALL, L. O. & GOLDGOF, D. B. Single pass fuzzy c means. *In 2007 IEEE International Fuzzy Systems Conference*, 2007b. https://doi.org/10.1109/FUZZY.2007.4295372.

HORE, P., HALL, L. O., AND GOLDGOF, D. B. . C. W. Online fuzzy c means. *n NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society*, 2008. https://doi.org/10.1109/NAFIPS.2008.4531233.

JAIN, A. K. AND MURTY, M. N. & FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys* 31 (3): 264–323, 1999. https://doi.org/10.1145/331499.331504.

JAWORSKI, M., DUDA, P., AND PIETRUCZUK, L. On fuzzy clustering of data streams with concept drift. *Artificial Intelligence and Soft Computing* vol. 2, pp. 82–91, 2012.

KRANEN, P., ASSENT, I., AND BALDAUF, C. & SEIDL, T. The clustree: Indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems* 29 (2): 249–272, 2011. https://doi.org/10.1007/s10115-010-0342-8.

KRISHNAPURAM, R. & KELLER, J. M. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems* 1 (2): 98–110, 1993. https://doi.org/10.1109/91.227387.

LI, Y., YANG, G., HE, H., AND JIAO, L. & SHANG, R. A study of large-scale data clustering based on fuzzy clustering. *Soft Computing* 20 (8): 3231–3242, 2016. https://doi.org/10.1007/s00500-015- 1698-1.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* vol. 1, pp. 281–297, 1967. https://doi.org/citeulike-article-id:6083430.

MITCHELL, T. Machine learning. McGraw-Hill Education, 1997.

MOSTAFAVI, S. & AMIRI, A. Extending fuzzy c-means to clustering data streams. *20th Iranian Conference on Electrical Engineering*, 2012. https://doi.org/10.1109/IranianCEE.2012.6292449.

SILVA, J. A., FARIA, E. R., BARROS, R. C., HRUSCHKA, E. R., AND CARVALHO, A. C. P. L. F. D. . G. J. Data stream clustering - a survey. *ACM Computing Surveys* 46 (1): 1–31, 2013. https://doi.org/10.1145/2522968.2522981.

WITTEN, I. H., FRANK, E., AND HALL, M. A. & PAL, C. Data mining: Practical machine learning tools and techniques. Morgan Kaufmann Series in Data Management Systems., 2017.

XIE, X. L. & BENI, G. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991. https://doi.org/10.1109/34.85677.