# Um novo método de inicialização para o algoritmo fuzzy c-means

Heloína Alves Arnaldo
*Dep. de Matemática Aplicada e Informática, DIMAp*
*Universidade Federal do Rio Grande do Norte, UFRN*
*59072-970, Natal, RN, Brasil*
*E-mail: heloar.alves@gmail.com*

Benjamín R. C. Bedregal
*Dep. de Matemática Aplicada e Informática, DIMAp*
*Universidade Federal do Rio Grande do Norte, UFRN*
*59072-970, Natal,Natal, RN, Brasil*
*E-mail: bedregal@dimap.ufrn.br*

***Abstract*—Agrupamento de dados é uma técnica aplicada a diversas áreas como mineração de dados, processamento de imagens e em problemas de reconhecimento de padrões. Um dos algoritmos de agrupamento mais populares é o Fuzzy C-Means (FCM). O desempenho do FCM depende da seleção dos centroides iniciais de agrupamento ou o valor de pertinência inicial. Se bons centroides iniciais que estão próximos dos centroides finais forem encontrados, o algoritmo convergirá com menor quantidade de iterações e o tempo de processamento pode ser reduzido. Neste trabalho é proposto um novo método de inicialização para o FCM a fim de acelerar o tempo de convergência necessário para particionar um conjunto de dados em grupos sem afetar a qualidade da partição obtida. Nossos resultados experimentais mostraram que o FCM inicializado com nosso método, obteve melhor desempenho em relação ao algoritmo original, cuja inicialização é aleatória. O novo método de inicialização permite reduzir o número de iterações e consequentemente acelerar o algoritmo. Os nossos experimentos também mostraram uma melhoria na qualidade do agrupamento.**

***Keywords*-agrupamento, fuzzy c-means, centroides iniciais.**

## I. Introdução

Agrupamento de dados desempenha uma tarefa muito importante em muitos campos da engenharia, tais como reconhecimento de padrões, modelagem de sistemas, processamento de imagem, comunicação, mineração de dados, e assim por diante. Métodos de agrupamento dividem um conjunto de $N$ objetos de dados em $c$ grupos, em relação a uma medida de similaridade apropriada, de modo que os membros de um mesmo grupo são mais semelhantes entre si do que para os membros de outros grupos [1].

Os algoritmos de agrupamentos podem ser classificados por meio de diferentes aspectos. Umas das classificações mais frequentes é proposta em [2] e utilizada em [3], onde os algoritmos são classificados de acordo com o método para definir os grupos. Neste caso, são divididos em hierárquicos e particionais. Algoritmos hierárquicos geram, a partir de uma matriz de proximidade, uma sequencia de partições aninhadas, enquanto os particionais procuram obter uma única partição dos dados de entrada, movendo os objetos de um grupo para outro e otimizando o critério de agrupamento.

O agrupamento pode ser realizado de acordo com a abordagem clássica (crisp), na qual cada objeto de dado pertence a um único grupo, ou de acordo com abordagens alternativas, como a *fuzzy*, onde um objeto pode pertencer a mais de um grupo, com diferentes graus de pertinência [4].

Em aplicações reais não é muito frequente existir fronteira crisp entre os grupos que descrevem os dados de modo que, o agrupamento *fuzzy* é mais adequado para capturar a incerteza da situação. Os graus de pertinência entre [0,1] representam melhor a natureza do problema prático em vez da atribuição crisp 0 ou 1.

Desde a teoria de conjuntos *fuzzy* [5] ,o agrupamento *fuzzy* tem sido amplamente estudado e aplicado em várias áreas, como processamento de imagens, recuperação de informação, mineração de dados e outras [6]. Na literatura o algoritmo de agrupamento Fuzzy C-Means (FCM), proposto em [7] e estendido em [8] é o algoritmo de agrupamento *fuzzy* mais utilizado e discutido. De acordo com a referência [9] o desempenho do FCM e variantes depende fortemente dos parâmetros iniciais, pois há deficiências neste método. Primeiro, a função objetivo do agrupamento *fuzzy* é uma função não convexa, que tem muitos extremos locais. É fácil o algoritmo mergulhar nestes extremos locais e consequentemente não obter a melhor partição *fuzzy*. Segundo, o desempenho de tempo não pode ser satisfeito para grandes conjuntos de dados e com alta dimensão, o que limita a aplicação do algoritmo. Devido à natureza iterativa do algoritmo, o processo de agrupamento de dados pode ser demorado quando se tem muitos objetos e grande número de vetores de atributos envolvidos nos cálculos.

Existem dois modos de inicialização para o FCM: inicializando aleatoriamente a matriz de pertinência e inicializando aleatoriamente os centroides iniciais de agrupamento. Um bom centroide inicial, pode chegar a uma melhor solução global, com poucas iteraçoes do algoritmo. No entanto, é difícil escolher um bom conjunto de centroides iniciais aleatoriamente. Este método atraiu atenções de muitos pesquisadores [9], [10], [11]. Em [12] é proposto o algoritmo *Multistage Random Sampling Fuzzy C-Means*. Este método baseia-se no pressuposto de que um pequeno subconjunto de dados pode ser utilizado para aproximar os centroides do conjunto de dados completo. Sob esta premissa o algoritmo FCM é usado para computar os centroides de um subconjunto de tamanho apropriado do conjunto de dados original. Depois de obter estes centroides, o subconjunto de

dados é mesclado com outro pequeno subconjunto adicional, selecionado aleatoriamente dos dados não processados para formar um grande subconjunto, que é processado pelo FCM. Os centroides anteriormente calculados são utilizados para a inicialização da matriz de pertinência *fuzzy* deste conjunto recém-formado. Este procedimento é repetido até que o tamanho do conjunto usado nos cálculos seja grande o suficiente para aproximar os centroides reais do conjunto de dados completo. Os centroides resultantes são então utilizados para a inicialização da matriz de pertinência *fuzzy* utilizada pelo FCM quando ele é aplicado ao conjunto de dados original.

Em [13] é proposto o algoritmo *Partition Simplification Fuzzy C-Means*. A ideia do método é simplificar o conjunto de dados e encontrar um conjunto candidato de centroides iniciais mais próximo possível dos centroides reais. Ele é dividido em duas fases. Na Fase I, o conjunto de dados é dividido em algumas células de blocos pequenos, utilizando o método *k-d tree* [14]. O conjunto de dados original é reduzido em um conjunto de dados simplificado com blocos de unidades, conforme descrito em [15]. Todos os dados de um bloco de unidade são substituídos pelo centroide destes dados. Em seguida, o grande número de dados no conjunto original é drasticamente reduzido a um pequeno número de centroides de blocos de unidades, ou seja, o conjunto de dados simplificado. Na etapa seguinte os centroides reais deste conjunto de dados simplificado são encontrados pelo algoritmo FCM. A Fase II é o processo padrão do FCM com os centroides inicializados pelos centroides finais da Fase I.

Em [16] os centroides são estimados com base no conceito de função montanha. O método baseia-se na formação de grades no espaço $R^s$ e a construção de uma função montanha a partir dos dados. Em seguida, é feita uma destruição das montanhas para obter os centroides do conjunto. A interseção das linhas de grades corresponde aos pontos candidatos a centroides. Em [17] a inicializacão dos centroides é feita usando o algoritmo *subtractive clustering* [17].

Neste artigo nós propomos um método para obter os centroides iniciais no algoritmo FCM utilizando a distância entre cada ponto do conjunto de dados e o ponto médio do conjunto. Nosso método seleciona melhores centroides iniciais nas bases de dados utilizadas nos experimentos em relação ao FCM com inicialização aleatória. Os experimentos mostram que nosso método pode acelerar a velocidade de processamento e reduzir a quantidade de iterações do algoritmo. Ao mesmo tempo, este método melhora a qualidade do agrupamento.

A estrutura do restante deste artigo é a seguinte. A seção II apresenta brevemente o algoritmo FCM. Na seção III apresentamos as medidas quantitativas usadas na avaliação da qualidade do agrupamento produzido pelos algoritmos aqui discutidos. A seção IV apresenta o nosso método de inicialização. Experimentos comparativos são descritos na seção V e os resultados apresentados na seção VI. E finalmente na seção VII apresentamos a conclusão.

## II. Fuzzy C-Means

O algoritmo para agrupamento de dados *fuzzy* foi proposto por [7] e estendido por [8]. Para um determinado conjunto de dados $X = \{x_1, x_2, ..., x_n\} \subset R^s$ o FCM é um processo iterativo que divide $X$ em $c$ grupos. O resultado do agrupamento é expresso pelos graus de pertinência na matriz $\mu$, onde $\mu_{ij}$ é o grau de pertinência do objeto $x_i$ ao j-ésimo grupo. O algoritmo FCM tenta encontrar uma partição *fuzzy* que representa a estrutura dos dados, minimizando a função objetivo definida como:

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c} \mu_{ij}^m d(x_i; c_j)^2 \tag{1}$$

com as seguintes restrições:

$$\begin{gathered} \sum_{j=1}^{c} \mu_{ij} = 1, \forall i \in \{1, ..., n\}; \\ 0 < \sum_{i=1}^{n} \mu_{ij} < n, \forall j \in \{1, ..., c\} \end{gathered} \tag{2}$$

onde:

- $n$ é o número de dados;
- $c$ é o número de grupos considerados no algoritmo;
- $m > 1$ é o parâmetro de ponderação que controla o quão *fuzzy* é a partição (valor de *fuzziness*). Usualmente, $m$ está no intervalo [1.25;2];
- $x_i \in R^s (i = 1, ..., n)$ é um vetor de dados. Onde cada posição no vetor representa um atributo do dado;
- $c_j \in R^s (j = 1, ..., c)$ é o centroide do j-ésimo grupo;
- $d(x_i; c_j)$ é a distância entre $x_i$ e $c_j$;

O algoritmo FCM consiste em atualizar os graus de pertinência e os valores dos centroides, utilizando (3) e (4). Em (3) o objetivo é atribuir um grau de pertinência ao dado $x_i$ em relação ao grupo *j* que seja proporcional ao seu grau de pertinência em relação aos demais grupos.

Os parâmetros de entrada para o algoritmo são os $n$ dados, o número de grupos $c$ e o valor do parâmetro de fuzzificação $m$. Os principais passos do FCM estão descritos abaixo:

Passo 1: Inicialize a matriz de pertinência $\mu$ com números aleatórios contínuos no intervalo [0,1];

Passo 2: Calcule o centroide do grupo $j$ usando (3)

$$c_j = \frac{\sum_{i=1}^{n} \mu_{ij}^m x_i}{\sum_{i=1}^{n} \mu_{ij}^m} \tag{3}$$

Passo 3: Calcule um valor inicial para $J$ usando (1);

Passo 4: Calcule a matriz de pertinência *fuzzy* $\mu$ da seguinte forma:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d(x_i;c_j)}{d(x_i;c_k)}\right)^{\frac{2}{m-1}}} \tag{4}$$

Passo 5: Se $d(J_U; J_A) \leq \varepsilon$ Pare. Se não retorne ao passo 2.

O critério de convergência é o limiar $\varepsilon > 0$. Outro critério de parada possível é quando um número de iterações préfixado for executado.

A matriz de pertinência $\mu$ requerida pelo algoritmo pode ser substituída pelos centroides iniciais de agrupamento, valores aleatórios dentro do intervalo de dados. Para isso, o passo 2 deve ser substituído pelo passo 4 em cada iteração.

## III. Validação de Agrupamento

Para medir os resultados de um algoritmo de agrupamento, alguns índices de validade foram propostos no domínio de mineração de dados [2]. Estes índices são utilizados para medir a qualidade do resultado de agrupamento levando em conta distâncias intra e intergrupos em busca de grupos compactos e bem separados [18], comparando-a com outros resultados obtidos por outros algoritmos de agrupamento, ou o mesmo algoritmo mas variando parâmetros diferentes. Tais índices podem ser de três tipos [3]:

- Internos: medem a qualidade de um agrupamento a partir de informações do próprio conjunto de dados (matriz de objetos ou matriz de similaridade). Geralmente, um critério interno analisa se as posições dos objetos em um agrupamento obtido corresponde à matriz de similaridade;
- Externos: avaliam um agrupamento de acordo com uma informação externa, geralmente uma intuição do pesquisador sobre a estrutura presente nos dados ou um agrupamento construído por um especialista de domínio;
- Relativos: comparam diversos agrupamentos para decidir qual deles é o mais adequado aos dados.

Neste trabalho, utilizamos dois índices de validade internos para avaliar os resultados de agrupamento do algoritmo FCM original, cuja inicialização dos centroides é aleatória e o FCM inicializado com nosso método proposto. O índice Davies-Bouldin (DB) [19] e o índice Silhueta Fuzzy (SF) [20]. Apresentamos agora as definições formais destes dois índices de validação de agrupamentos.

Considere um objeto $j \in \{1, ..., n\}$ pertencente a um grupo $p \in 1, ..., c$. A silhueta deste objeto pode ser calculada como segue:

$$s_j = \frac{b_{pj} - a_{pj}}{max\{a_{pj}, b_{pj}\}} \tag{5}$$

onde $a_{pj}$ é a distância média do objeto $j$ a todos os objetos de seu grupo $p$. Para calcular $b_{pj}$, considere a distância média $d_{qj}$ do objeto $j$ a todos os objetos de outro grupo $q$. Pode-se definir $b_{pj}$ como o menor valor $d_{qj}$ calculado para $q \in 1, ..., c$, $q \neq p$, que representa a dissimilaridade do objeto $j$ ao grupo vizinho mais próximo. O denominador desta equação é usado apenas como um termo de normalização. Claramente, o maior valor de $s_j$ é a melhor atribuição do objeto $j$ ao grupo $p$.

A Silhueta Fuzzy baseia-se na similaridade entre os objetos de um grupo e na dissimilaridade destes objetos em relação ao grupo vizinho mais próximo para avaliar quantitativamente partições de dados. É formalmente definida como:

$$SF = \frac{\sum_{j=1}^{N}(u_{pj} - u_{qj})^{\alpha} s_j}{\sum_{j=1}^{N}(u_{pj} - u_{qj})^{\alpha}} \tag{6}$$

onde $s_j$ é a silhueta do objeto $j$ obtida usando (5), $u_{ij}$ e $u_{qj}$ são o primeiro e o segundo maiores elementos da j-ésima coluna da matriz de pertinência, respectivamente, e por fim, $\alpha \geq 0$ é um coeficiente de ponderação, cujo valor normalmente é 1.

A $SF$ assume valores entre -1 e 1. Para este índice quanto maior o valor de $SF$, melhor é a partição do agrupamento.

Dada uma partição *fuzzy* $C = \{C_1, ..., C_k\}$ resultante do agrupamento do conjunto $X = \{x_1, x_2, ..., x_n\}$ em $k$ grupos, o índice Davies-Bouldin é definido da seguinte forma:

$$DB(C) = \frac{1}{k}\sum_{i=1}^{k}\max_{i \neq j}\left\{\frac{S_i + S_j}{M_{ij}}\right\} \tag{7}$$

onde $S_i$ é a distância intragrupo e $M_{ij}$ é a distância intergrupo. Elas são definidas por (8) e (9):

$$S_i = \sqrt[q]{\frac{1}{T_i}\sum_{j=1}^{T_i}|X_j - A_i|^q} \tag{8}$$

$$M_{ij} = \sqrt[q]{\sum_{k=1}^{n}|a_{k,i} - a_{k,j}|^p} \tag{9}$$

onde $T_i$ é o tamanho do grupo $i$, $A_i$ é o centroide de $i$ e $a_{ki}$ é o k-ésimo elemento do vetor s-dimensional $A_i$. Normalmente, o valor de $q$ é 2, o que torna $S_i$ a distância Euclidiana entre os centroides dos grupos e os objetos de dados. Quando $p = 2$, $M_{ij}$ é a distância Euclidiana entre os centroides do grupo $i$ e $j$.

O índice Davies-Bouldin minimiza a distância intragrupo e maximiza a distância intergrupo. Portanto, para um determinado conjunto de dados quanto maior a semelhança dentro do grupo e maior a separação dos grupos, menor será o valor do índice DB. Um bom método de agrupamento deve fazer o valor do índice DB o mais baixo possível.

## IV. Um novo método de inicialização

O algoritmo FCM é inicializado com valores aleatórios. Diferentes centroides iniciais resultam em resultados diferentes de agrupamento. Portanto, este método não produz um agrupamento único, sendo necessárias várias execuções do mesmo para obter uma configuração inicial com resultados mais representativos do agrupamento.

Neste trabalho, propomos uma nova abordagem para encontrar os centroides iniciais com reduzida complexidade de tempo. Estes centroides são utilizados para inicializar a matriz de pertinência, eliminando o processo aleatório de inicialização. Nosso método consiste em primeiro lugar no cálculo da distância entre cada objeto de dado e o ponto médio do conjunto de dados. Em seguida, os objetos originais são ordenados de acordo com tais distâncias e particionados em c conjuntos iguais, onde c representa o número de grupos no agrupamento. Em cada conjunto os pontos médios são tomados como centroides iniciais. Estes centroides conduzem a um agrupamento único.

Nosso método de inicializações dos centroides é descrito nos seguintes passos:

- Passo 1 : Calcule a média $\overline{x}$ do conjunto de dados $X = \{x_1, x_2, ..., x_n\}$;
- Passo 2 : Para cada objeto de dado, calcule a distância entre $x_i$ e o ponto médio $\overline{x}$;
- Passo 3 : Ordene os objetos de acordo com as distâncias;
- Passo 4 : Particione os objetos ordenados em $c$ conjuntos iguais, $X = \{X_1, ..., X_c\}$, no número de elementos;
- Passo 5 : Em cada conjunto $X_i$, tome o ponto médio como o centroide inicial.
- Passo 6 : Inicialize a matriz de pertinência $\mu$ usando a equação (4);
- Passo 7 : Execute o passo 2 ao passo 5 do algoritmo FCM.

## V. Experimentos

A fim de comparar o nosso método proposto e o FCM de inicializações aleatórias, tanto do ponto de vista de eficiência (tempo de execução e quantidade de iterações) como qualidade do agrupamento (valores dos índices DB e SF), utilizamos para o problema de agrupamento os conjuntos de dados Iris[1] [21] e E.coli[2] [22]. Estes conjuntos de dados são descritos a seguir.

Os algoritmos deste trabalho foram desenvolvidos em Python (versão 2.7.3) usando a biblioteca numérica Numpy[3] (versão1. 7.1). Os experimentos foram executados em um notebook com processador Intel Core i7-3612QM, 4 CPU's de 2.10GHz, com 8Gb de memória principal, usando o sistema operacional Linux (Kernel 3.8.3-030803-generic, Cinnamon 1.8.8, Mint 13).

### A. Conjunto de Dados Iris

O conjunto de dados Iris [21] contém 150 instâncias, com quatro atributos numéricos cada (comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala). Estes dados estão classificados em três classes de 50 instâncias, onde cada classe se refere a um tipo da planta Iris (Iris Setosa, Iris Versicolour e Iris Virginica).

### B. Conjunto de Dados E.coli

O conjunto de dados E.coli[23] contém 336 instancias descritas por sete atributos numéricos. Estes atributos correspondem a sequencias de aminoácidos em proteínas de bactérias gram-negativas [24] e de células eucarióticas [25]. As 336 instancias estão classificadas em oito classes, listadas na Tabela I. Cada uma das classes se refere ao local de localização de uma proteína numa célula.

Table I
CLASSIFICAÇÃO DA BASE E.COLI

| Classes | Instancias |
|---|---|
| citoplasma | 143 |
| membrana interna sem sequencia de sinal | 77 |
| periplasma | 52 |
| lipoproteína de membrana externa | 5 |
| membrana exterior | 20 |
| lipoproteína de membrana interna | 2 |
| membrana interna de sequencia de sinal inquebrável | 35 |
| membrana interna de sequencia de sinal quebrável | 2 |

### C. Configuração de Parâmetros

Os experimentos foram executados variando os parâmetros descritos na tabela II. O número de grupos $c$ foi definido como sendo o valor da quantidade de classes em cada conjunto de dados. Assim, para os conjuntos de dados Iris e E.coli temos $c = 3$ e $c = 8$, respectivamente. O algoritmo FCM original foi executado com 100 inicializações aleatórias diferentes, até convergir em cada execução. A média do resultado das 100 execuções foi utilizada na comparação com o FCM inicializado com nosso método. Este foi executado com apenas uma inicialização, pois nossa proposta fornece um agrupamento único.

Table II
PARÂMETROS DE INICIALIZAÇÃO

| Parâmetro | Variável | Valor |
|---|---|---|
| valor de fuzziness | m | 1.5; 1.75; 2.0 |
| critério de parada | $\varepsilon$ | 0.0001 |

## VI. Resultados Experimentais

Como já mencionado, avaliamos os algoritmos de agrupamento em termos de dois índices internos, DB e SF. Eles medem a compactação dos agrupamentos gerados. Comparamos também o tempo de execução (em segundos) e o número de iterações para o algoritmo convergir.

[1]base de dados do repositório Machine Learning Repository (UCI). Disponível em http://archive.ics.uci.edu/ml/datasets/Iris.

[2]http://archive.ics.uci.edu/ml/datasets/Ecoli

[3]pacote para computação científica em Python. Disponível em http://www.numpy.org/

### A. Iris

As Tabelas III e IV apresentam os resultados do índice DB e SF respectivamente, nos experimentos com a base Iris, onde $\varepsilon = 0,0001$. Para o algoritmo FCM estes valores representam a média de 100 execuções. Os melhores resultados estão destacados.

Table III
VALORES DO ÍNDICE DAVIES-BOULDIN PARA A IRIS

| c | m | DB | |
|---|---|---|---|
| | | FCM proposto | FCM original |
| 3 | 1.5 | **0.630208** | 0.700325 |
| | 1.75 | 0.640762 | 0.710331 |
| | 2.0 | 0.651879 | 0.651885 |

Como os grupos precisam ser compactos e separados, quanto menor for o valor do índice DB, melhor a partição do agrupamento. Nota-se pela Tabela III que o menor valor do índice DB, é obtido pelo algoritmo FCM inicializado com nosso método. A diferença entre este valor e o resultado do FCM com inicialização aleatória é de 0.07017. Logo, nosso método trouxe melhoria para a qualidade do agrupamento. Nota-se ainda que o menor valor de DB é obtido quando o valor de fuzziness m = 1.5.

Table IV
VALORES DO ÍNDICE SILHOUETA FUZZY PARA A IRIS

| c | m | SF | |
|---|---|---|---|
| | | FCM proposto | FCM original |
| 3 | 1.5 | 0.597677 | 0.591932 |
| | 1.75 | 0.614557 | 0.606554 |
| | 2.0 | **0.628263** | 0.628234 |

O índice SF com maior valor indica um melhor agrupamento. Observando a Tabela IV, nota-se que o nosso método obteve um melhor resultado com $SF = 0.628263$. A diferença para o FCM original é de apenas $0.000029$. O maior valor de SF é alcançado quando o valor de fuzziness $m = 2.0$.

A Tabela V mostra a quantidade de iterações e o tempo total em segundos que os algoritmos levaram para convergir. Para o FCM com inicialização aleatória, esses valores são a média de 100 execuções.

Table V
TEMPO DE EXECUÇÃO E NÚMERO DE ITERAÇÕES PARA A BASE IRIS

| c | m | Tempo | | Iterações | |
|---|---|---|---|---|---|
| | | FCM proposto | FCM | FCM proposto | FCM |
| 3 | 1.5 | 0.32 | 0.34 | 12 | 13 |
| | 1.75 | **0.31** | 0.41 | **12** | 14 |
| | 2.0 | 0.33 | 0.44 | 12 | 16 |

Observe que o algoritmo FCM inicializado com nosso método convergiu com o número de iterações e tempo de execução menor em todas as configurações de parâmetros. O nosso algoritmo convergiu mais rápido com o valor de fuzziness $m = 1.75$, totalizando um tempo de execução de $0.31$ segundos em 12 iterações. Já o FCM inicializado aleatóriamente totalizou $0.41$ segundos de processamento em 14 iterações.

### B. E.coli

As Tabelas VI e VII apresentam os resultados dos índices DB e SF respectivamente, nos experimentos com a base E.coli, onde $\varepsilon = 0.0001$.

Table VI
VALORES DO ÍNDICE DAVIES-BOULDIN PARA A E.COLI

| c | m | DB | |
|---|---|---|---|
| | | FCM proposto | FCM original |
| 3 | 1.5 | **1.869467** | 1.979089 |
| | 1.75 | 2.068824 | 2.321475 |
| | 2.0 | 2.684128 | 3.015531 |

Analisando a Tabela VI notamos que o menor valor de DB é obtido quando o valor de fuzziness $m = 1.5$. Enquanto nosso método obteve o valor $DB = 1.869467$, o FCM tradicional teve como resultado o valor $DB = 1.979089$. A diferença entre estes resultados é de $0,109622$. Logo, nosso método trouxe melhoria para a qualidade do agrupamento também na base E.coli.

Table VII
VALORES DO ÍNDICE SILHOUETA FUZZY PARA A E.COLI

| c | m | SF | |
|---|---|---|---|
| | | FCM proposto | FCM original |
| 3 | 1.5 | 0.307569 | 0.295582 |
| | 1.75 | 0.319813 | 0.302884 |
| | 2.0 | **0.320246** | 0.296463 |

Observando a Tabela VII, nota-se que o maior valor de SF é alcançado quando o valor de fuzziness $m = 2.0$. Nosso método obteve o valor $SF = 0.320246$, contra $0.296463$ do FCM com inicialização aleatória. Portanto houve ganho na qualidade do agrupamento pelo nosso método.

A Tabela VIII mostra a quantidade de iterações e o tempo total em segundos que os algoritmos levaram para convergir.

Table VIII
TEMPO DE EXECUÇÃO E NÚMERO DE ITERAÇÕES PARA A BASE E.COLI

| k | m | Tempo | | Iterações | |
|---|---|---|---|---|---|
| | | FCM proposto | FCM | FCM proposto | FCM |
| 3 | 1.5 | **2.23** | 3.65 | **52** | 91 |
| | 1.75 | 2.36 | 4.79 | **52** | 111 |
| | 2.0 | 2.30 | 3.37 | 57 | 86 |

Analisando os resultados listados na Tabela VIII podemos ver que o desempenho do nosso método foi superior em todos os casos. Usando nossa inicialização o FCM convergiu em menor tempo e com um número menor de iterações. O melhor resultado foi obtido com valor de fuzziness $m = 1.5$, onde o FCM proposto convergiu em $2.23$ segundos e

52 iterações, enquanto o FCM original executou em 3.65 segundos e 91 iterações.

Com os resultados apresentados, percebe-se que, apesar de não serem tão significantes, as diferenças mostram que o método proposto é mais eficiente e eficaz do que o método de escolha aleatória de centroides.

## VII. Conclusão

A principal contribuição deste trabalho foi propor uma nova abordagem para obter os centroides iniciais no algoritmo de agrupamento Fuzzy C-Means, reduzindo o número de iterações necessárias para o algoritmo convergir. A nossa proposta fornece uma aceleração no tempo de processamento e melhoria na qualidade do agrupamento perante a aplicação do FCM original, onde os centroides iniciais são escolhidos aleatoriamente.

A análise de agrupamento usando o Fuzzy C-Means com inicialização aleatória não é um processo realizado em apenas uma execução. É necessária uma série de tentativas e repetições para uma dada configuração de parâmetros de entrada. Uma vez que na nossa abordagem de inicialização o algoritmo gera sempre os mesmos centroides iniciais, não é necessário a execução reiterada do FCM (modificado) para se fazer uma análise estatística do seu desempenho. Nosso método consiste no particionamento em subgrupos do conjunto de dados ordenado, segundo a distância de cada dado ao ponto médio do conjunto. Em cada subconjunto o ponto médio é tomado como centroide.

Os experimentos realizados com os conjuntos de dados Iris e E.coli mostram que o agrupamento utilizando o nosso método, em termos de valores dos índices de validação DB e SF, tem qualidade superior ao agrupamento com o FCM original. Em nossos resultados experimentais também conseguimos acelerar o tempo de execução do algoritmo. Portanto, para estas bases de dados nossa proposta de inicialização seleciona melhores centroides iniciais para o algoritmo FCM.

## References


[1] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[3] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[4] S. Nascimento, B. Mirkin, and F. Moura-Pires, "A fuzzy clustering model of data and fuzzy c-means," in *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 302–307.

[5] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[6] F. d. A. de Carvalho, "Fuzzy c-means clustering methods for symbolic interval data," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 423–437, 2007.

[7] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[8] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[9] X. Gao and W. Xie, "Research process of fuzzy clustering theory development and its application," *Science Bulletin*, vol. 44, no. 21, pp. 2241–2251, 1999.

[10] D.-W. Kim, K. H. Lee, and D. Lee, "A novel initialization scheme for the fuzzy c-means algorithm for color clustering," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 227–237, 2004.

[11] J. Pei, J. Fang, and W. Xie, "An initialization method of cluster centers," *Journal of Electronics and Science*, vol. 21, no. 11, pp. 320–325, 1999.

[12] T. W. Cheng, D. B. Goldgof, and L. O. Hall, "Fast fuzzy clustering," *Fuzzy sets and systems*, vol. 93, no. 1, pp. 49–56, 1998.

[13] M.-C. Hung and D.-L. Yang, "An efficient fuzzy c-means clustering algorithm," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 225–232.

[14] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[15] D.-L. Yang, J.-H. Chang, M. Hung, and J. Liu, "An efficient k-means-based clustering algorithm," in *Proceedings of 1st Asia-Pacific Conference on Intelligent Agent Technology*, 1999, pp. 269–273.

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

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

[18] F. Boutin and M. Hascoet, "Cluster validity indices for graph partitioning," in *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*. IEEE, 2004, pp. 376–381.

[19] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 224–227, 1979.

[20] R. J. Campello and E. R. Hruschka, "A fuzzy extension of the silhouette width criterion for cluster analysis," *Fuzzy Sets and Systems*, vol. 157, no. 21, pp. 2858–2875, 2006.

[21] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[22] B. Vandeginste, "Parvus: An extendable package of programs for data exploration, classification and correlation, m. forina, r. leardi, c. armanino and s. lanteri, elsevier, amsterdam, 1988, price: Us 645 isbn 0-444-43012-1," *Journal of Chemometrics*, vol. 4, no. 2, pp. 191–193, 1990.

[23] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins." in *Ismb*, vol. 4, 1996, pp. 109–115.

[24] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Bioinformatics*, vol. 11, no. 2, pp. 95–110, 1991.

[25] ——, "A knowledge base for predicting protein localization sites in eukaryotic cells," *Genomics*, vol. 14, no. 4, pp. 897–911, 1992.