UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Trabalho Computacional 1: Fuzzy C-Means

Universidade Federal de Minas Gerais

Departamento de Engenharia Elétrica

ELE075 - Sistemas Nebulosos

Autor: Moises Mendes de Assis
(Matrícula: 2014015524)

Data: 22 de abril de 2018

1 Introdução

O algoritmo *Fuzzy C-Means* é um método de agrupamento (*clustering*) de dados baseado na teoria de conjuntos nebulosos (*fuzzy*), em que um dado pode pertencer a mais de um grupo. O grau de pertencimento a um grupo é definido com base na distância entre cada ponto e alguns pontos específicos chamados centroides, que é definido pelo parâmetro C. Este trabalho acompanha os códigos em *MATLAB* com a implementação do método. Serão descritas as decisões de implementação, a validação do algoritmo com a base de dados FCM e uma aplicação do método para segmentação de imagem.

2 Decisões de implementação

As principais decisões de implementação envolvem a escolha de alguns parâmetros e operações, descritas a seguir.

2.1 Parâmetro C (número de grupos)

O parâmetro C é o número de médias ou centroides calculadas pelo método, e também representa o número de grupos identificados, cada um associado com um centroide. Foram feitos testes com dois tipos de dados: um deles foi a base de dados FCM e o outro foram imagens.

No caso da base de dados FCM foi possível visualizar os dados graficamente, pois a base de dados é formada por 800 pontos com coordenadas x e y. Visualmente, foi possível identificar que se tratavam de 4 grupos de dados, portanto C = 4. Já no caso das imagens, o número de grupos é escolhido de acordo com uma inspeção visual, com base na quantidade de cores das diferentes regiões da imagem. Foram utilizados valores de C variando entre 5 e 8 grupos.

2.2 Inicialização da matriz de pertinência

Para inicialização da matriz, utilizou-se uma distribuição uniforme (função `randi` do MATLAB). Gerou-se uma matriz de números inteiros e dividiu-se cada linha pela soma daquela linha, de modo que a soma dos valores em cada linha fosse igual a 1.

2.3 Atualização da matriz de pertinência

A atualização da matriz de pertinência é calculada em função da distância entre um ponto x e os centroides. Para calcular a distância foi utilizada a norma do vetor $x - c_k$, onde c_k é o centroide do grupo ou *cluster* k. Essa norma é então elevada ao quadrado e utilizada no cálculo da pertinência da seguinte forma: quanto mais próximo um ponto estiver do centroide c_k, maior será o seu grau de pertencimento ao grupo k. Além disso, é importante garantir que a soma de cada linha continue igual a 1 após essa atualização.

2.4 Critério de parada

Foram definidos critérios de parada diferentes para cada um dos dois testes, um para o teste com a base de dados FCM e outro para a segmentação de imagens.

1. **Base de dados FCM.** O critério utilizado para essa base de dados foi verificar se os pontos não mudaram de grupo entre duas iterações consecutivas. Isso foi possível pois o número de pontos era relativamente pequeno ($n = 800$ pontos).

2. **Segmentação de imagens.** Nesse caso não foi possível aplicar o mesmo critério de parada utilizado no caso anterior, pois o número de pontos obtidos da conversão da imagem para RGB foi muito grande (entre 250 mil e 300 mil pontos). Portanto, o critério foi verificar se os centroides não mudavam muito entre duas iterações consecutivas. Para isso utilizou-se a ideia da norma infinita entre os centroides de uma iteração k e da iteração $k+1$.

$$\max_i(\max_j(c_k, c_{k+1})) < \epsilon$$

   Lembrando que c_k e c_{k+1} são matrizes, são necessários duas operações de máximo, uma que retorna um vetor e outra que retorna um valor. Além disso, o valor de ϵ foi definido empiricamente a partir de testes com as imagens como $\epsilon = 0.1$.

3 Validação com a base de dados FCM

A base de dados FCM é composta pelo conjunto de $n = 800$ pontos apresentados na Figura 1. É possível verificar visualmente a presença de 4 grupos distintos. Foram realizados três testes com essa base de dados:

1. Agrupamento em K = 2 grupos (Figura 2)
2. Agrupamento em K = 4 grupos (Figura 3)
3. Agrupamento em K = 8 grupos (Figura 4)

Para cada um dos testes acima, o algoritmo *Fuzzy C-means* foi comparado com o algoritmo *K-means*. Para isso, ambos foram executados N = 30 vezes para cada uma das três instâncias acima (K = {2, 4, 8}). Foi realizada uma comparação em termos de:

i. número médio de iterações até a convergência,

ii. número de vezes que o algoritmo encontra valores adequados para os centros dos *clusters*.

A verificação de valores adequados para os *clusters* foi feita através de inspeção visual do resultado do algoritmo, como nas Figuras 2-4, que apresentam os valores adequados para os centros dos *clusters*. Escolheu-se números para os quais

seria possível identificar facilmente grupos diferentes e igualmente divididos espacialmente. Para K = 2, dois grupos com duas nuvens de pontos são identificados, como mostrado na Figura 2. Para K = 4, cada nuvem de pontos deve ter um grupo associado a ela, como indicado na Figura 3. Finalmente, para K = 8, cada nuvem de pontos deve ter dois grupos associados a ela, como apontado na Figura 4.

Os resultados dos testes descritos acima são sumarizados nas Tabelas 1-3, que apresentam a comparação entre os dois algoritmos em termos de número de acertos na identificação dos *clusters* e número médio de iterações.

Os valores reportados nas Tabelas 1-3 mostram que, no geral, o algoritmo *Fuzzy C-means* demora mais iterações a convergir, porém encontra melhores resultados que o algoritmo *K-means*. Além disso, percebe-se que o *Fuzzy C-means* tem uma taxa de acerto maior que o *K-means*, especialmente à medida que o número de grupos a serem identificados aumenta.

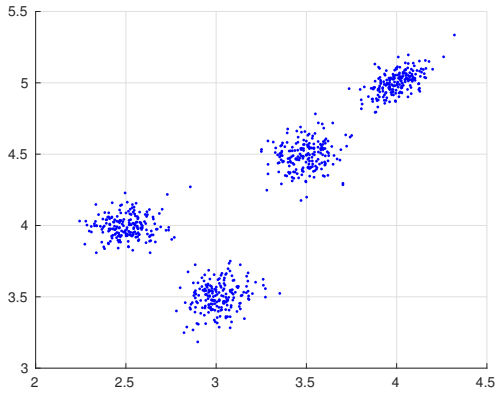

Figura 1: Base de dados FCM

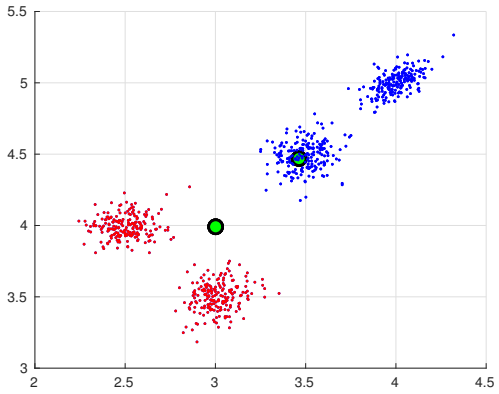

Figura 2: Resultado para K = 2

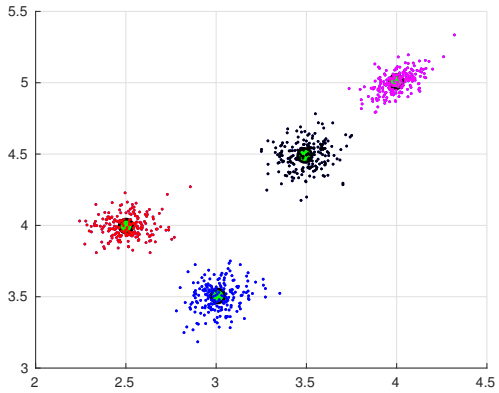

Figura 3: Resultado para K = 4

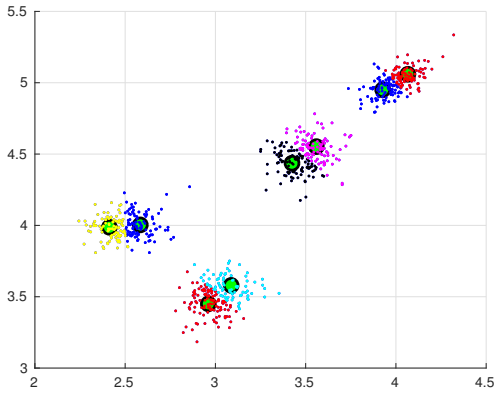

Figura 4: Resultado para K = 8

Tabela 1: Comparação para K = 2 grupos

N = 30 execuções	**K-means**	**C-means**
Erros	0 (0%)	0 (0%)
Acertos	30 (100%)	30 (100%)
Média de Iterações	4	4

Tabela 2: Comparação para K = 4 grupos

N = 30 execuções	**K-means**	**C-means**
Erros	16 (53%)	5 (17%)
Acertos	14 (47%)	25 (84%)
Média de Iterações	7	9

Tabela 3: Comparação para K = 8 grupos

N = 30 execuções	**K-means**	**C-means**
Erros	30 (100%)	24 (80%)
Acertos	0 (0%)	6 (20%)
Média de Iterações	13	24

4 Aplicação: segmentação de imagens por região

Após a validação do algoritmo *Fuzzy C-means* desenvolvido com a base de dados FCM, ele foi utilizado para segmentação de imagens RGB. Para isso, o conjunto de *pixels* da imagem no formato RGB foi transformado em uma matriz $X = [r, g, b]$. O algoritmo foi aplicado a essa matriz para identificação de K *clusters*. Ao final da execução, a matriz de partição U foi utilizada para colorir cada *cluster* da imagem com a tonalidade do *pixel* que corresponde ao centro da região, de forma que os *pixels* que apresentarem maior grau de compatibilidade (pertinência) a uma dada região foram coloridos com a tonalidade do *pixel* central daquela região.

Para todos as imagens testadas, o número inicial de *clusters* foi $K = 5$. Após uma primeira execução do algoritmo, a imagem resultante foi avaliada e, caso necessário, o número K foi incrementado. Para a maioria dos casos, o valor de K com resultado satisfatório foi $K = \{5, 6\}$. Houve dois casos em que foi utilizado $K = 8$ devido ao número de cores diferentes presentes na imagem, que foram os casos das Figuras 20 e 22.

Essa aplicação mostra a versatilidade e a eficiência do algoritmo *Fuzzy C-means*, pois é possível aplicá-lo a diferentes tipos de dados e, como foi apresentado na seção 3, esse algoritmo apresenta desempenho melhor que um algoritmo semelhante (*K-means*). Além disso, percebe-se que o número de grupos necessário para segmentar a imagem de forma satisfatória não é muito grande, o que diminui o esforço computacional para realização da aplicação.

Figura 5: Imagem original

Figura 6: Imagem resultante K = 5

Figura 7: Imagem original

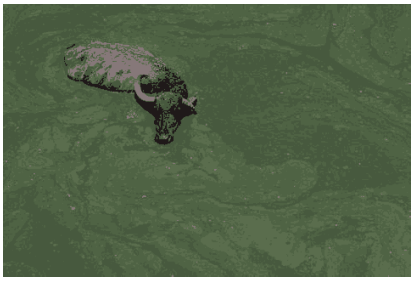

Figura 8: Imagem resultante K = 5

Figura 9: Imagem original

Figura 10: Imagem resultante K = 5

Figura 11: Imagem original

Figura 12: Imagem resultante K = 5

Figura 13: Imagem original

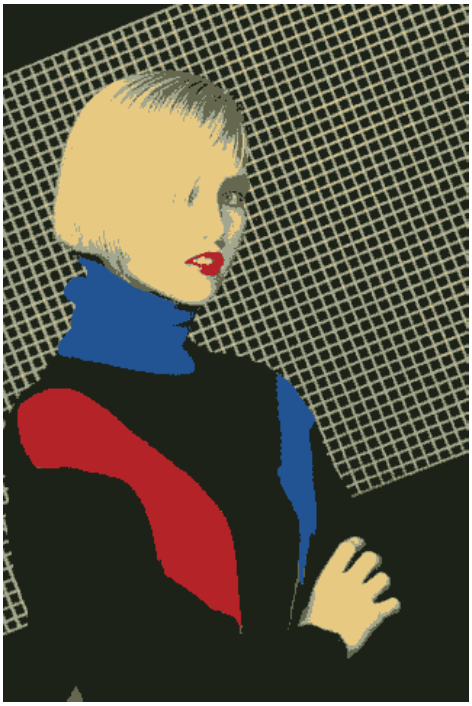

Figura 14: Imagem resultante K = 6

Figura 15: Imagem original

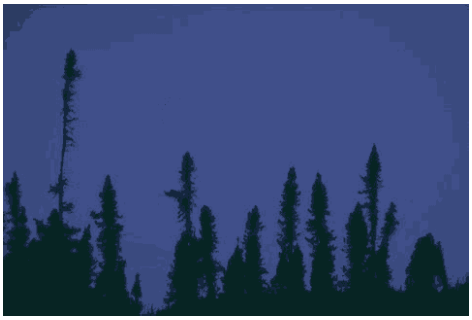

Figura 16: Imagem resultante K = 5

Figura 17: Imagem original

Figura 18: Imagem resultante K = 6

Figura 19: Imagem original

Figura 20: Imagem resultante K = 8

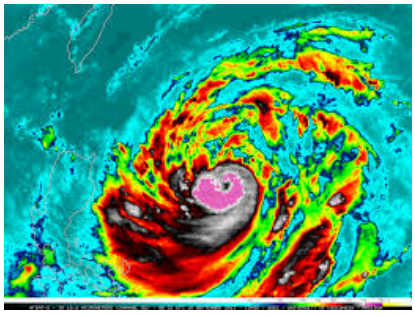

Figura 21: Imagem original

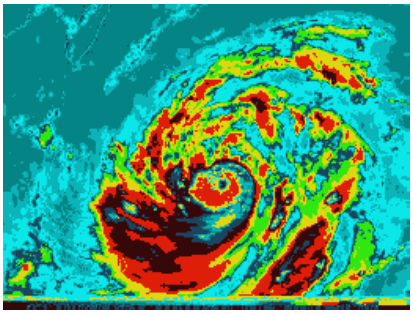

Figura 22: Imagem resultante K = 8

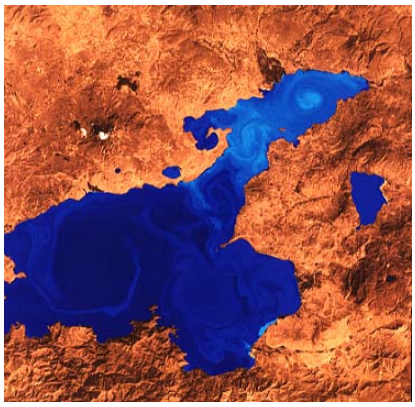

Figura 23: Imagem original

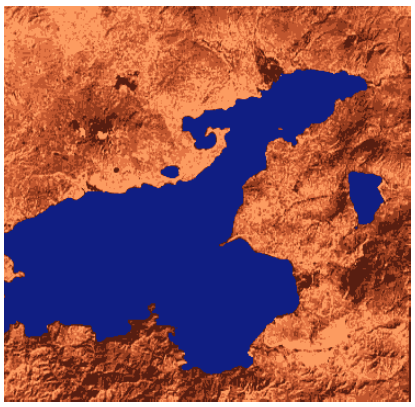

Figura 24: Imagem resultante K = 6

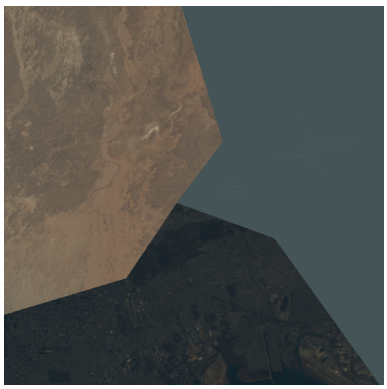

Figura 25: Imagem original

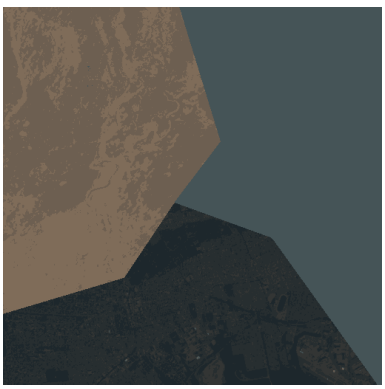

Figura 26: Imagem resultante K = 6