DOCUMENTAÇÃO D ETALHADA DO ALGORITMO A SER IMPLEMENTADO.

O conjunto de dados utilizado foi extraído da tabela segmentatiton.test (http://archive.ics.uci.edu/ml/machine-learning-databases/image). Essa tabela apresenta 2100 objetos com 19 variáveis cada e essas variáveis podem ser divididas em 2 views:

- Shape view: as primeiras 9 variáveis
- RGB view: as 10 ultimas variáveis.

Esse conjunto de dados foram rotulados com 7 classes. Definindo o conjunto de objetos como $E = \{e_1, \dots, e_{2100}\}$ e seu conjunto de variáveis sendo definida como p então teremos para cada uma das views a seguinte quantidade de matrizes de dissimilaridade:

- **Shape view**: $D_j = [d_j(e_i, e_l)]$ onde $j = 1, \dots, p$, tal que $p = 9$ e $i, l = 1, \dots, n$, tal que $n = 2100$. No caso, ira existir 9 matrizes de dissimilaridade para essa view.
- **RGB view**: $D_j = [d_j(e_i, e_l)]$ onde $j = 1, \dots, p$, tal que $p = 10$ e $i, l = 1, \dots, n$, tal que $n = 2100$. No caso, ira existir 10 matrizes de dissimilaridade para essa view.

Existira no sistemas K clusters, no caso o conjunto de clusters fuzzy , também chamado de partição, é definido como $C = C_1, \dots, C_K$, sendo $K = 7$, teremos 7 partições no sistema. Cada uma dessas partições terá protótipos que apresentam os melhores representantes dessa partição, definidos pelo vetor $G = \{G_1, \dots, G_K\}$. O protótipo G_K apresenta um subconjunto de cardinalidade fixa igual a $i \leq q \leq n$. Dado que $q = 3$, teremos K protótipos com um subconjunto de 3 objetos pertencentes ao conjunto E. Temos a seguinte definição que é equivalente ao que foi dito anteriormente, onde $G_K \in E^q = \{A \subset E : |A| = q\}$. A função que será utilizada para definir os melhores parâmetros que irão representar as minhas partições é definida por, para $s = 1$:

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik})^m \sum_{j=1}^{p} \lambda_{kj} \sum_{e \in G_K} d_j(e_i, e)$$

Os parâmetros K, n, m, são conhecidos e apresentam o valor de 7, 2100 e 1,6 respectivamente. O valor de p varia para cada uma das views que será abordada separadamente, para Shape View teremos $p = 9$ e para RGB view teremos $p = 10$. Então teremos que definir os seguintes parâmetros da minha função:

- $\boldsymbol{u_{ik}}$: Representa a matriz de pertinência dos objetos em relação a cada uma dos cluster. No caso existira uma matriz de pertinência com 2100 linhas por 7 colunas, representando os objetos e as colunas representando as 7 partições.
- $\boldsymbol{\lambda_{kj}}$: Representa o vetor de pesos para cada um dos atributos dos objetos, para cada partição, no caso para Shape View teremos um vetor de pesos de tamanho 9 e para o RGB view teremos um vetor de pesos de tamanho 10.
- $\boldsymbol{G_K}$: Representa os protótipos de uma partição, os objetos que melhor representam uma determinada partição.

Inicialmente o algoritmo obtém os valores de G_K aleatoriamente, pegando um subconjunto aleatório pertencente ao conjunto E para os 7 protótipos de cada partição. O vetor de pesos $\boldsymbol{\lambda_{kj}}$ é iniciado com pesos iguais a 1 para cada partição. Já a matriz de pertinência $\boldsymbol{u_{ik}}$ utiliza a proposição 2.5 do artigo para se obter o grau de pertinência de cada um dos objetos a cada uma das partições do conjunto de dados.

$$u_{ik} = \left[\sum_{h=1}^{K}\left(\frac{D_{(\lambda_k,s)}(e_i, G_k)}{D_{(\lambda_h,s)}(e_i, G_h)}\right)^{1/(m-1)}\right]^{-1} = \left[\sum_{h=1}^{K}\left(\frac{\sum_{j=1}^{p}(\lambda_{kj})^s \sum_{e\in G_k} d_j(e_i, e)}{\sum_{j=1}^{p}(\lambda_{hj})^s \sum_{e\in G_h} d_j(e_i, e)}\right)^{1/(m-1)}\right]^{-1}$$

No caso, para um determinado objeto o seu grau de pertinência é definido como $\boldsymbol{u_{ik}}$, tal que o numerador calcula a distância do objeto em relação aos objetos do protótipo G_K para cada atributo e multiplica o resultado pelo peso de cada atributo, realizando o somatório de todos os resultado para cada atributo. Já o denominador o mesmo cálculo, entretanto, como se trata de G_h, o denominado irá calcular a distância para cada uma das partições.

Depois de realizar a primeira interação, recalculamos o valor dos protótipos G_k utilizando a proposição 2.3 do artigo.

$$G^* \leftarrow \emptyset$$
$$\textit{REPEAT}$$
$$\quad \textit{Find } e_l \in E, e_l \notin G^* \textit{ such that } l = \operatorname{argmin}_{1\le h\le n} \sum_{i=1}^{n}(u_{ik})^m \sum_{j=1}^{p}(\lambda_{kj})^s d(e_i, e_h)$$
$$\quad G^* \leftarrow G^* \cup \{e_l\}$$
$$\textit{UNTIL } |G^*| = q$$

Nesse cálculo será escolhido o objeto que apresentará menor dissimilaridade a uma partição K. No caso, teremos uma matriz de 2100 objetos que serão calculadas suas dissimilaridades em relação a cada umas das K partições do conjunto de dados. Pegando assim os 3 menores argumentos de cada uma das partições.

E finalmente será realizado o cálculo dos pesos de cada um dos atributos de cada uma das partições do conjunto de dados, utilizando a proposição 2.4 do artigo.

$$\lambda_{kj} = \frac{\{\prod_{h=1}^{p}[\sum_{i=1}^{n}(u_{ik})^m D_h(e_i, G_k)]\}^{1/p}}{[\sum_{i=1}^{n}(u_{ik})^m D_j(e_i, G_k)]} = \frac{\left\{\prod_{h=1}^{p}\left[\sum_{i=1}^{n}(u_{ik})^m \sum_{e\in G_k} d_h(e_i, e)\right]\right\}^{1/p}}{\left[\sum_{i=1}^{n}(u_{ik})^m \sum_{e\in G_k} d_j(e_i, e)\right]}$$

O numerador calcula a distância do objeto em relação aos objetos do protótipo G_K para cada atributo e multiplica o resultado pelo peso de cada atributo, realizando o somatório de todos os resultado para cada atributo. No caso o numerador percorrera todos os atributos dos objetos.

Já o denominador será fixo em um único atributo j, pois está sendo calculado o peso $\boldsymbol{\lambda_{kj}}$ de atributo j e partição k. Em seguida é realizado o cálculo da matriz de pertinência que já foi mostrado anteriormente. Em cada uma desses processos, são fixados dois parâmetros enquanto o outro parâmetro é recalculado.