RELATÓRIO - MS777

BIBLIOTECA DATA CLUSTERING EM C++

por Igor Felipe Carboni Battazza (RA 096866)
ORIENTADOR: Prof. Francisco Magalhães Gomes

Análise de Cluster - Definições

Definição 1. *Análise de Cluster é um método para criar grupo de objetos (ou clusters), de modo que pode-se observar semelhanças entre estes e classificá-los de acordo com critério(s) pré determinado(s).*

Definição 2. *Um conjunto de dados (data set) de n objetos, cada um descrito por d atributos é denotado por* $D = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}\}$, *onde* $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{\mathrm{id}})^T$ *é um vetor denotando o i-ésimo objeto e* x_{ij} *é um escalar denotando o j-ésimo componente ou atributo de* x_i. *O número de atributos d é chamado de dimensão do data set.*

Definição 3. *Distâncias são usadas na análise de clursters para descrever quantitativamente o quanto similar dois pontos são similares ou quanto dois clusters são similares. Quanto menor a distância, maior a similaridade entre os dois objetos. Reciprocamente, quanto maior a distância, menor a suas semelhanças.*

Definição 4. *O conceito de cluster é usado sem uma definição formal. Usualmente, um cluster contém objetos que seguem o(s) seguinte(s) critério(s):*

- *Possuem a(s) mesma(s) característica(s) ou característica(s) próxima(s);*
- *Possuem pequena distâncias mútuas ou dissimilaridades;*
- *Possuem "contatos" ou "relações" com pelo menos um outro objeto no grupo; ou*
- *Distinguem-se do resto dos objetos no grupo de dados.*

Usualmente, um cluster pode ser representado por um centro geométrico ou ponto.

Definição 5. *Tipos de dados:*

- *Contínuo. Ex.: O intervalo* $[0, 1] \subset \mathbb{R}$.
- *Discreto:*
  - *Nominal.*
  - *Binário Simétrico e Binário Assimétrico.*

Definição 6. *(TABELA DE CATEGORIA) Seja* $D = \{x_1, x_2, ..., x_n\}$ *um data set categorizado com n ocorrências, cada uma é descrita por d atributos* $v_1, v_2, ..., v_d$. *Seja* $\mathrm{DOM}(v_j)$ *o domínio do atributo* v_j.

Definição 7. *(TABELA DE SIMBOLOS) Para um data set catalogado D, suponha que* $\mathrm{DOM}(v_j) = [A_{j1}, A_{j2}, ..., A_{jn_j}]$ *para* $j = 1, 2, ..., d$. *Chame* A_{jl} $(1 \leqslant l \leqslant n_j)$*de um estado de atributo categorizado* v_j,*e* n_j *o número de estados de* v_j *no data set dado D. Então uma tabela de simbolos T, do data set é definida por* $T_s = (s_1, s_2, ..., s_{il})$ *onde* s_j $(1 \leqslant j \leqslant d)$ *é um vetor definido por* $s_j = (A_{j1}, A_{j2}, ..., A_{jn_j})^T$.

Definição 8. *(TABELA DE FREQUÊNCIA) Seja C um cluster. Então a Tabela de Frequência* $T_f(C)$ *do cluster C é definida por* $T_f(C) = (f_1(C), f_2(C), ..., f_d(C))$, *ond* $f_j(C)$ *é um vetor definido por* $f_j(C) = (f_{j1}(C), f_{j2}(C), ..., f_{jn_j}(C))^T$. *Onde* $f_{jr}(C)$ $(1 \leqslant j \leqslant d, 1 \leqslant r \leqslant n_j)$ *é o número de pontos no cluster C que tem o valor* A_{jr} *na j-ésima dimensão, i.e.,* $f_{jr}(C) = |\{x \in C : x_j = A_{jr}\}|$, *onde* x_j *é o valor do j-ésimo componente de x.*

Algoritmo das Médias *k*

Seja D um data set com n instancias, e seja $C_1, C_2, ..., C_k$ k clusters dijuntos de D. Então a função erro é definida por $E = \sum_{i=1}^{k} \sum_{x \in C} d(x, \mu(C_i))$, onde $\mu(C_i)$ é o centroide do cluster C_i, $d(x, \mu(C_i))$ a distancia entre x e $\mu(C_i)$.

Pseudo Algorítmo

ENTRADA: Data set $\boldsymbol{D}$. Número de Clusters $\boldsymbol{k}$, Dimensões $\boldsymbol{d}$:

[$\boldsymbol{C_i}$ é o $\boldsymbol{i}$-ésimo cluster]

{1. Fase de Inicialização}

1. $(\boldsymbol{C_1}, \boldsymbol{C_2}, ..., \boldsymbol{C_k}) =$ Partição inicial de $\boldsymbol{D}$.

{2. Fase de Iteração}

2. REPETIR

3. $\boldsymbol{d_{ij}} =$ distancia entre caso $\boldsymbol{i}$ e cluster $\boldsymbol{j}$;

4. $\boldsymbol{n_i} = \arg\min_{1 \leqslant j \leqslant k} \boldsymbol{d_{ij}}$;

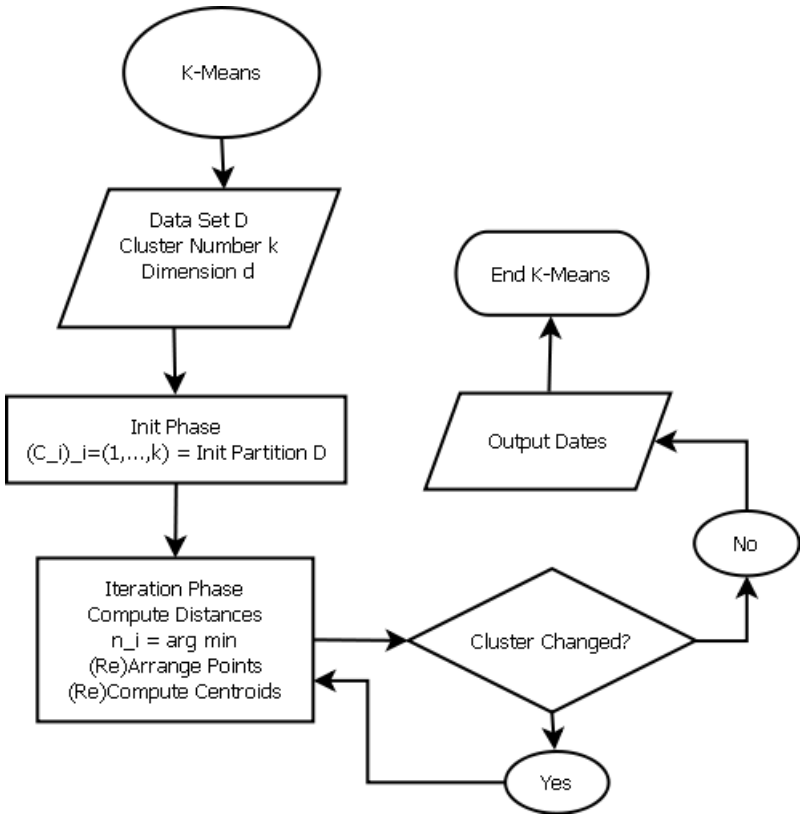
5. Assimilar caso $\boldsymbol{i}$ com cluster $\boldsymbol{n_i}$;

6. Recalcular a média do cluster com qualquer cluster mudado anteriormente;

7. ATÉ não houver mudanças de membros de cluster;

8. Saida de resultados.

Fluxograma



Algorítmo das Médias k Fuzzy

Seja D um data set com n objetos, cada um descrito por d atributos e seja c um inteiro entre um e n. Então uma c-partição é definida por uma $c \times n$ matriz $U = (u_{li})$ que satisfaz $u_{li} \in [0, 1]$, $1 \leqslant k \leqslant c$, $1 \leqslant i \leqslant n$, $\sum_{l=1}^{c} u_{li} = 1$, $1 \leqslant i \leqslant n$, $\sum_{i=1}^{n} u_{li} > 0$, $1 \leqslant l \leqslant c$, onde u_{li} denota o grau do membro do objeto i no l-ésimo cluster.

Para cada c-partição fuzzy, existe uma correspondencia c-partição hard. Seja u_{li} ($l = 1, 2, ..., c$, $i = 1, 2, ..., n$) um membro de qualquer c-partição fuzzy. Então a correspondência c-partição hard de u_{il} pode ser definida como ω_{li} dada por $w_{li} = \begin{cases} 1, \arg \max\limits_{l \leqslant j \leqslant c} u_{ji}, \\ 0, \text{caso contrario} \end{cases}$.

Dado um data set $D = \{x_1, x_2, ..., x_n\}$, o algoritmo de médias k fuzzy é baseado em minimizar a função obetiva $J_q(U, V) = \sum\limits_{j=1}^{n} \sum\limits_{i=1}^{k} u_{ij}^q d^2(x_j, V_i)$ com respeito a U (uma k-partição fuzzy do data set) e de V (um conjunto de k prototipos), onde q é um número real maior que 1, V_i é o centroide do cluster i, u_{ij} é o grau de membro do objeto x_j pertencentes ao cluster i, $d^2(.\,,.)$ é o quadrado da distância euclidiana, e k é o número de clusters. O parametro q controla o "fuziness" dos clusters resultantes.

Pseudo Algorítmo

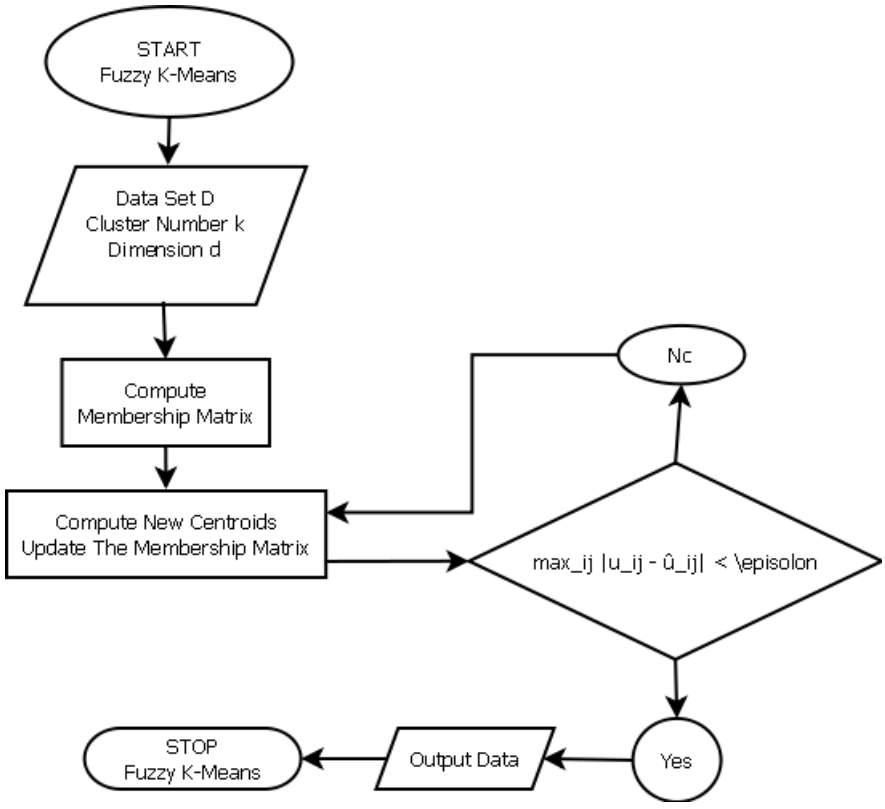
1. Calcule os centroides iniciais $V_i (i = 1, 2, ..., k)$;

2. Compute a matriz membro dada por $u_{ij} = \frac{[d^2(x_j, V_i)]^{-\frac{1}{q-1}}}{\sum\limits_{l=1}^{k} [d^2(x_j, V_l)]^{-\frac{1}{q-1}}}$, $l = 1, 2, ..., k$, $j = 1, 2, ..., n$;

3. Compute os novos centroides $\hat{V}_i$ $(i = 1, 2, ..., k)$ dado por $\hat{V}_i = \frac{\sum\limits_{j=1}^{n} u_{ij}^q x_j}{\sum\limits_{j=1}^{n} u_{ij}^q}$. e atualize a matriz membro (u_{ij}) para $(\hat{u}_{ij})$ de acordo com a equação do passo (2).

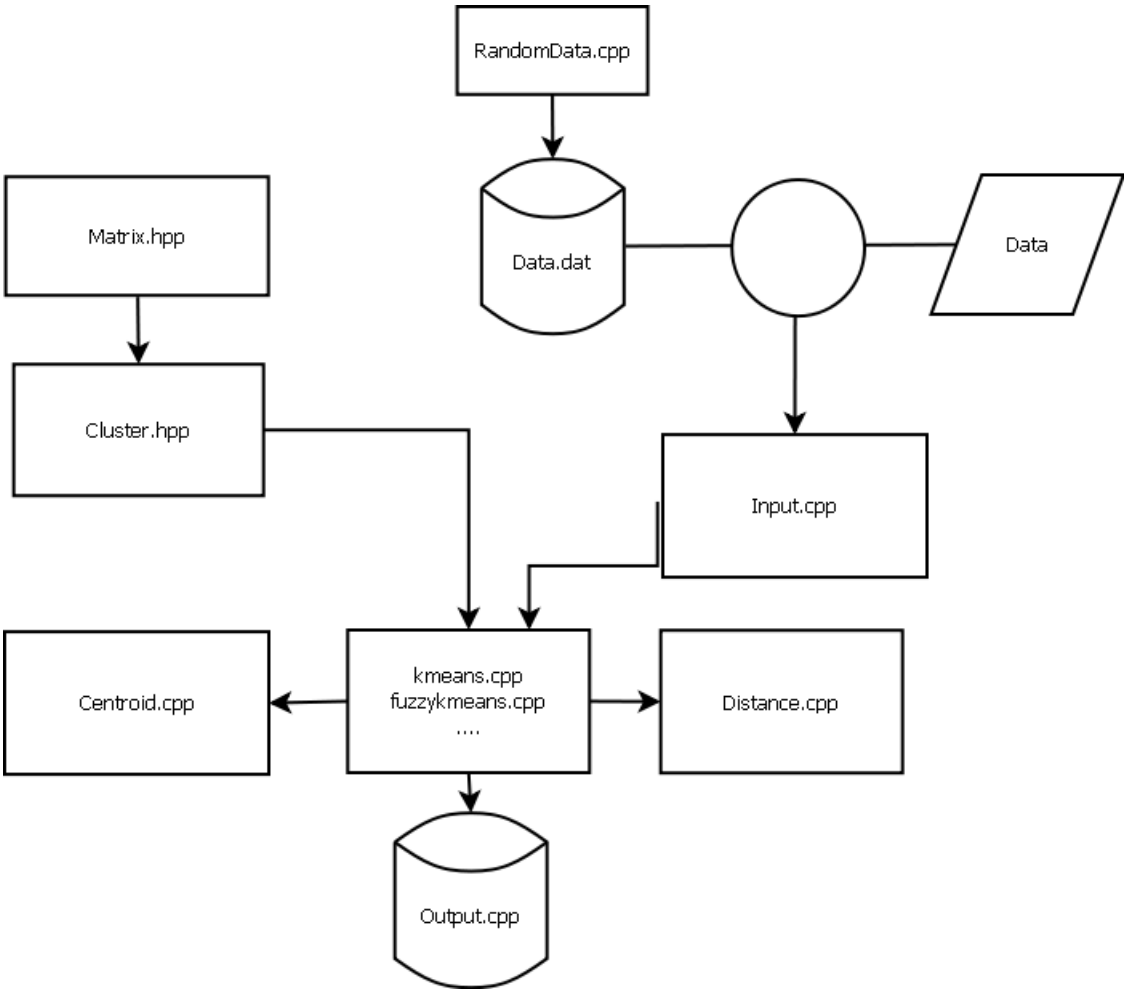
4. Se $\max_{ij} |u_{ij} - \hat{u}_{ij}| < \varepsilon$, então pare. Caso contrário, vá para o passo (3), onde ε é a terminação do criterio entre 0 e 1.

Fluxograma



Biblioteca Data Clustering

Biblioteca construida na linguagem C++ para dar suporte para algorítmos de análise de cluster.



A biblioteca se baseia em quatro partes independentes:

(a) Análise de Cluster

Inicialmente foi construída uma estrutura de classe baseada no Basic Linear Algebra Subprogram (BLAS) Nível 1 permitindo operações do tipo $A = \alpha \cdot B + \beta \cdot C$, sendo A, B, C matrizes e α, β um escalar. Posteriormente foram implementadas as operações de multiplicação de matrizes.

As operações de atribuição ($=$), adição ($+$) e multiplicação ($*$) foram implementadas utilizando sopreposição de operadores disponível na linguagem C++.

Foi introduzido também como função membro para transposta da matriz.

Todas as implementações acima estão contidas em uma camada primitiva "Matrix.hpp". Uma segunda camada acima desta, chamada "Cluster.hpp" faz um encapsulamento para o vocabulário utilizado em Data Clustering.

(b) Distâncias e Centroides

Há uma biblioteca específica, chamada Distance.cpp, para cálculo de distâncias onde foi implementado inicialmente uma função para calcular distâncias através da norma euclidiana generalizada pela norma de Minkowski definida por $d^p(P,Q) = \sum_{i=1}^{n} |x_i - y_i|^p$, onde $P = (x_1, x_2, ..., x_n)$ e $Q = (y_1, y_2, ..., y_n)$. Outra função importante é o cálculo do centroide, dado por $C = \frac{x_1 + x_2 + ... + x_n}{n}$, $x_i \in \mathbb{R}^n$.

(c) Entrada e Saída

Há funções específicas para o tratamento de entrada de dados de um arquivo .dat contendo as coordenadas iniciais dos pontos em seus devidos clusters. Foi implementado também funções para criação de dados de entrada randomicos para facilitar o teste dos algoritmos.

Funções de tratamento de saída de dados permitem de uma forma bastante elementar trazer toda a saída para plotagem em programas como GnuPlot e MatLab simplesmente imprimindo-os em um arquivo ".dat" com codificação ASCII em formato de linhas e colunas como ilustrado a seguir:

(d) Algorítmos de Clustering

Implementação dos algoritmos de Clustering. Implantados os algoritmos tradicionais de Médias k e a sua versão fuzzy. Futuramente será implementada outros algoritmos como por exemplo Médias c e QT (Quality Threshold), assim como variantes dos mesmos com otimizações.

Uma quinta parte independente já está planejada e será utilizada para realização de testes com imagens contendo as seguintes etapas:

(1) Fragmentação da imagem em muitos pedaços;

(2) Randonização da posição de cada pedaço;

(3) Reagrupamento destes pedaços procurando se aproximar da imagem original.

Bibliografia

[1] Guojun Gan; Chaoqun Ma; and Jianhong Wu. Data Clustering: Theory, Algorithms, and Applications. Editora SIAM, 2007, ISBN 089-871-623-3

[2] Paul Deitel, Harvey M. Deitel. C++ How to Program. 7a. Edição. Editora Prentice Hall, 2009, ISBN 013-611-726-0