07/12/2021

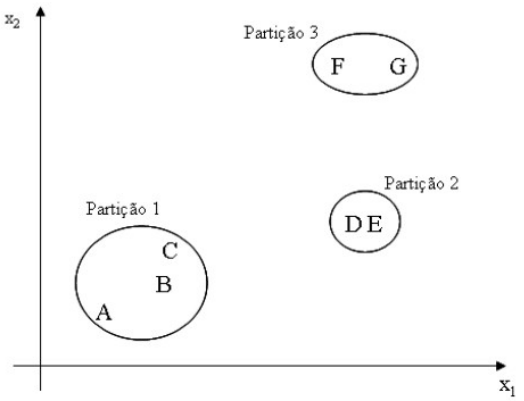
Primeiro Trabalho de Programação Funcional

Prof. Flávio Miguel Varejão

I. Descrição do Problema

Agrupamento de dados multidimensionais é um dos problemas mais comuns na área de aprendizado de máquina. Esse problema consiste em dividir um conjunto de pontos em um espaço multidimensional em um determinado número pré-especificado de grupos de modo que os pontos pertencentes a um mesmo grupo estão mais relacionados entre si e menos relacionados em relação aos pontos associados aos outros grupos.

A figura abaixo ilustra um exemplo de agrupamento no qual os sete pontos {A, B, C, D, E, F, G} foram agrupados em três grupos, indicando que os padrões {A, B, C} são mais similares entre si do que em relação aos demais, assim como os padrões {D, E} e {F, G}.



Formalmente, dado um conjunto de dados X com N pontos $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, sendo que cada ponto $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^t$ possui d coordenadas (dimensões), deseja-se encontrar K grupos $\{C_1, \ldots, C_K\}$, de tal forma que as seguintes condições sejam atendidas:

- $C_j \neq \emptyset,\ j = 1, \ldots, K$
- $\bigcup_{j=1}^{K} C_j = X$
- $C_i \cap C_j = \emptyset,\ i \neq j,\ i, j = 1, \ldots, K$

Uma forma de realizar agrupamento de dados multidimensionais envolve inicialmente a criação da árvore geradora mínima (minimal spanning tree) e em seguida cortar os ramos da árvore de forma a criar os K grupos desejados. O corte da árvore é sempre na maior aresta presente na floresta de árvores existentes.

O pseudo-código a seguir ilustra os passos para realização de agrupamento de dados usando uma árvore geradora mínima:

1. Escolher um ponto inicial para compor a árvore geradora mínima
2. Adicionar o ponto mais próximo a qualquer nó da árvores à árvore geradora mínima
3. Repetir o passo 2 até que todos os pontos tenham sido adicionados à árvore geradora mínima

4. Escolher a maior aresta da árvore geradora mínima para dividi-la em dois grupos
5. Repetir o passo 4 para a floresta de árvores formadas até que se tenham apenas K árvores (os K grupos).

Neste trabalho será usada a distância Euclidiana $|| x_i - x_j ||$ como métrica de distância. Ela é calculada pela expressão:

$$|| x_i - x_j || = \sqrt{((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{id} - x_{jd})^2}$$

Cada ponto do conjunto de dados a ser agrupado terá suas coordenadas (valor numérico em ponto flutuante) expressas em uma linha do arquivo csv de entrada. A linha em que os dados do ponto se encontra será o seu identificador único. Assim, o ponto na primeira linha será identificado pelo número 1, o ponto na segunda linha será identificado pelo número 2 e assim por diante.

Para uniformizar os resultados, o ponto inicial escolhido será o ponto 1, isto é, o primeiro ponto lido do arquivo.

II. Especificação do Sistema

Funcionalidades a serem implementadas:

1. Leitura do nome do arquivo de entrada, do nome do arquivo de saída e do número de grupos da entrada padrão.
2. Leitura da base de dados do arquivo csv de entrada.
3. Realização do agrupamento de dados.
4. Gravação dos identificadores dos pontos dos grupos no arquivo csv de saida. Cada linha do arquivo de saída corresponderá a um grupo.
5. Apresentação dos identificadores dos pontos dos grupos na saida padrão. Cada linha da saída corresponderá a um grupo.

Os exemplos seguintes são apenas ilustrativos dos formatos de entrada e saída e não existe correspondência entre os seus dados.

Exemplo de formato de arquivo de entrada:

```
7,5.4,6.32,9
17,32.3,5,9.99
33,54,5.6,65.8
77.7,33.4,98,7.56
8.9,5.8,6,9
```

Exemplo de formato de arquivo de saida (com K = 2):

```
1, 3, 5
2, 4
```

Exemplo de formato de interação do programa com o usuário:

```
Forneca o nome do arquivo de entrada: base.csv
Forneca o nome do arquivo de saida: saida.csv
Forneca o número de grupos (K): 2
Agrupamentos:
1, 3, 5
2, 4
```

III. Condições de Entrega

O trabalho deve ser feito individualmente e submetido pelo sistema da sala virtual até a data limite (31 de janeiro de 2022).

O trabalho deve ser submetido em um arquivo zip contendo todos os arquivos com código fonte em haskell. O arquivo zip deve possuir o nome Trab1_Nome_Sobrenome. Note que a data limite já leva em conta um dia adicional de tolerância para o caso de problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo. Em caso de recebimento do trabalho após a data limite, o trabalho não será avaliado e a nota será ZERO. Aluno que receber zero por este motivo e vier pedir para o professor considerar o trabalho não será nem respondido. Trabalhos em que se configure cópia receberão nota zero independente de quem fez ou quem copiou.

IV. Requisitos da implementação

a. Modularize seu código adequadamente.

b. Crie códigos claros e organizados. Utilize um estilo de programação consistente, Comente seu código.

c. Os arquivos do programa devem ser lidos e gerados na mesma pasta onde se encontram os arquivos fonte do seu programa.

V. Observação importante

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, freqüentando as aulas ou acompanhando as novidades na página da disciplina na sala virtual.