05/04/2019

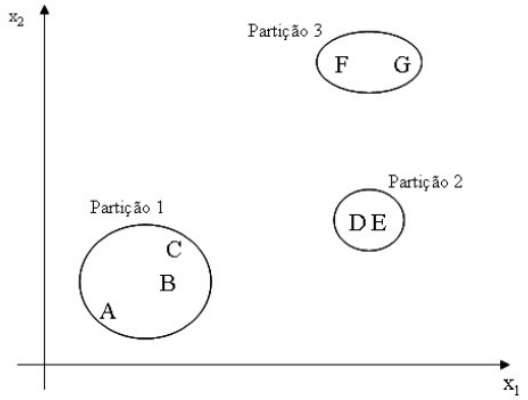
Primeiro Trabalho de LP

Prof. Flávio Miguel Varejão

I. Descrição do Problema

Agrupamento de dados multidimensionais é um dos problemas mais comuns na área de mineração de dados. 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:

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

Além de atender essas condições, a qualidade da divisão em grupos deve ser avaliada. Neste trabalho o critério de qualidade utilizado será a soma das distâncias euclidianas quadradas (SSE) entre os pontos pertencentes a cada um dos grupos:

$$SSE = \sum_{j=1}^{K} \sum_{\mathbf{x_i} \in C_j} \|\mathbf{x_i} - \mu_\mathbf{j}\|^2$$

onde $\| x_i - \mu_j \|$ é a distância Euclidiana entre o ponto x_i e o centróide μ_j.

O centróide $\mu_\mathbf{j} = [\mu_{j1}, \mu_{j2}, \ldots, \mu_{jd}]_t$ é o ponto representativo do grupo C_j e é calculado

como o centro de massa do grupo:

$$\mu_{\mathbf{j}} = \frac{1}{n_j} \sum_{\mathbf{x_i} \in C_j} \mathbf{x_i}$$

onde n_j é o total de pontos pertencentes ao grupo C_j.

A distância Euclideana $|| x_i - \mu_j ||$ é calculada pela expressão:

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

Neste trabalho será necessário implementar o algoritmo de agrupamento descrito a seguir:

Entradas:

Conjunto de Dados D

Distância limite τ

Saída:

SSE da divisão em grupos e os grupos formados

```
1  L ← D0;
2  F0 ← D0;
3  para cada d ∈ D \ {D0} faça
4  |  líder ← verdadeiro;
5  |  para cada l ∈ L faça
6  |  |  se ||l − d|| ≤ τ então
7  |  |  |  Fl ← Fl ∪ d;
8  |  |  |  líder ← falso;
9  |  |  |  break ;
10 |  se líder então
11 |  |  L ← L ∪ d;
12 |  |  Fd ← d;
13 retorne {L, F}
```

Nas primeiras duas linhas do algoritmo, o primeiro ponto D_o do conjunto de dados D é incluído nos conjunto de líderes L de grupos e no primeiro grupo F_0. Em seguida, na terceira linha, se percorre cada elemento d restante do conjunto de dados, se verificando nas linhas 5 e 6, se este elemento d está a uma distância inferior a τ de algum lider l. Em caso positivo, esse elemento d é inserido no grupo F_l correspondente a este líder l. Se o

elemento *d* não está próximo de nenhum líder, um novo grupo F_d é criado, tendo este elemento *d* como líder, o qual é inserido no conjunto de líderes. Ao final do algoritmo, se é retornado o conjunto de líderes *L* e o conjunto com todos os grupos *F*. A partir de *F*, se deve calcular a *SSE* do agrupamento.

II. Especificação do Sistema

Funcionalidades a serem implementadas:

*1.*Leitura da distância limite τ de um arquivo texto denominado "distancia.txt". Esse arquivo possui apenas uma linha contendo o valor de um ponto flutuante.
*2.*Leitura dos dados dos pontos de um arquivo texto denominado "entrada.txt". Cada linha corresponde a um ponto. As coordenadas de um ponto são colocadas sucessivamente em uma linha separadas por espaço
3. Executar o algoritmo de agrupamento
*4.*Gravação do valor da SSE do resultado do agrupamento em um arquivo denominado"result.txt". O valor da SSE deve ter obrigatoriamente 4 casas decimais.
5. Gravação dos pontos de cada grupo em um arquivo chamado “saida.txt”. Cada ponto é representado pelo número da linha no qual foi registrado no arquivo entrada.txt. Os números que representam os pontos de cada grupo devem ser colocados em uma ou mais linhas sequenciais separados por espaços. Os números de cada grupo devem ser apresentados em ordem crescente. A separação dos números de um grupo para outro será marcada por uma linha em branco

Formato dos Dados do Sistema:

```
distância limite (τ):    ponto flutuante não negativo
coordenadas dos pontos:  inteiro ou ponto flutuante
SSE:                     ponto flutuante
número das linhas:       inteiro não negativo
```

Exemplo de arquivo entrada.txt:

```
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 arquivo result.txt:

```
988.3217
```

Exemplo de arquivo saida.txt:

```
2 6

3 5

4
```

III. Requisitos da implementação

1.Modularize seu código adequadamente.

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

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

IV. Condições de Entrega

O trabalho deve ser feito individualmente e submetido por e-mail até as 23:59 horas da data limite especificada para o endereço fvarejao@gmail.com com o subject LP_TRABALHO_X_NomedoAluno_SobrenomedoAluno onde X é o número do trabalho (1, 2, 3 ou 4). 1 é o de Go, 2 é o de Lua, 3 é o de Haskell e 4 é o de Python. O e-mail deve conter também um arquivo .zip com o mesmo nome do subject do e-mail enviado. O arquivo principal (o que contém o main do trabalho) obrigatoriamente deve estar com o nome "main". 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 estará cometendo um ato de DESRESPEITO ao professor e estará sujeito a perda adicional de pontos na média.

V. Datas de Entrega

Go: 30/04/2018

Lua: 21/05/2018

Haskell: 11/06/2018

Python: 25/06/2018

A correção/revisão dos trabalhos será na aula de laboratório subsequente a data de entrega.

VI. Avaliação

Os trabalhos terão nota zero se:

A data de entrega for fora do prazo estabelecido;

O trabalho não compilar;

O trabalho não gerar o arquivo com o resultado e formato esperado;

For detectada a ocorrência de plágio.

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 Internet.