PL 4

Algoritmos Probabilísticos

SECÇÃO PARA AVALIAÇÃO [1]

Considere uma aplicação, a desenvolver em Matlab, tendo por objetivo ajudar na gestão de contactos pessoais. A aplicação deve suportar um conjunto de utilizadores identificados por um ID e um conjunto de contactos também identificados por um ID (ID definido por um inteiro positivo).

Dados de entrada:

Considere o ficheiro `contactos.txt` disponível para download em `https://bit.ly/3o3pSwD`. As colunas deste ficheiro contêm a seguinte informação: **coluna 1**, ID de um utilizador; **coluna 2**, ID de um contacto do utilizador da coluna 1; **colunas 3 a 5**, ano, mês e dia do último contacto.

A informação sobre cada um dos utilizadores encontra-se num segundo ficheiro, `utiliz.txt`, disponível em `https://bit.ly/3raMFsI`, com o seguinte conteúdo:

```
8;Loureiro;Valentino;44;FR;Surf;Fotografia;Jogos;Leitura;Política;Andebol
9;Branco;Liliana;44;NL;Parapente;Música;Gastronomia;Fórmula 1; ...
10;Rocha Freitas;Martina;39;NL;Música;Filmes;Política
```

em que os dados de cada coluna estão separados por ";". A linha número n contém a informação do utilizador com o ID n usado no ficheiro `contactos.txt`. A primeira coluna contém o número, a segunda o apelido, a terceira o nome (próprio), a quarta a idade e a quinta o código do país de que é natural. As restantes colunas contém um número variável de interesses do utilizador, como, por exemplo, "Parapente".

NOTA: executando no Matlab: `dic= readcell('utiliz.txt', 'Delimiter',';');` é criado o cell array `dic` em que a célula `dic{i,j}` contém a informação da linha i e da coluna j do ficheiro.

Descrição da aplicação a desenvolver:

A aplicação deve começar por pedir o ID do utilizador que se torna o utilizador actual [2]:

```
Insert Valid User ID :
```

certificando-se que o número introduzido é um ID válido. De seguida, a aplicação deve permitir o utilizador seleccionar uma de 5 opções:

```
1 - Your Contacts
2 - Interests from most similar user
3 - Search Name
4 - Find most similar contacts based in list of interests
5 - Exit
Select choice:
```

Opção 1: A aplicação lista os contactos do utilizador atual. Cada linha deve mostrar o ID e o nome completo.

Opção 2: A aplicação lista os interesses do utilizador mais "similar" ao utilizador actual e, após isso, lista os não partilhados pelo utilizador atual, como sugestões de novos interesses. A aplicação deve começar por

[1] A execução desta secção será objeto de avaliação. Assim, deverá fazer um relatório em PDF com todos os códigos Matlab desenvolvidos devidamente explicados e as opções de desenvolvimento devidamente justificadas. O relatório deverá começar por identificar o ano letivo, a disciplina, a turma prática e os elementos do grupo (nome e No. Mec.) que realizou o trabalho. **Deverá submeter um ficheiro comprimido com o relatório e todos os ficheiros necessários à execução da aplicação desenvolvida. Tenha em atenção os prazos estipulados**

[2] Para introdução de dados pelo teclado, investigue a utilidade da função Matlab `input`

determinar qual de todos os outros utilizadores é mais similar ao utilizador actual **em termos da lista de nacionalidades dos seus contactos**.

***Opção 3*[3] :**

A aplicação deve primeiro pedir a inserção de uma string: `Write a string:`
e depois apresentar os nomes completos de utilizadores mais similares à string introduzida (um por linha). Para além do nome, cada linha deve apresentar a estimativa da distância de Jaccard entre a string introduzida e cada nome completo.

A lista apresentada deve ter no máximo 4 nomes, ordenados por ordem crescente de distância de Jaccard e deve apresentar apenas nomes cuja distância seja menor ou igual a 0.7. Se não houver nenhum nome nestas condições, a aplicação deve indicar que não encontrou nenhum nome.

Antes da lista de nomes, com base na utilização de um filtro de Bloom, deve aparecer uma mensagem a indicar se a string introduzida pode corresponder exatamente a um dos nomes dos utilizadores.

Opção 4: A aplicação começa por listar os contactos do utilizador atual (ID e nome) e pede para escolher um desses contactos. Em seguida a aplicação deve apresentar os 4 utilizadores mais similares ao contacto escolhido com base nos interesses dele. A determinação destes utilizadores tem de obrigatoriamente ser baseada na utilização de MinHash.

Opção 5: A aplicação termina.

Notas sobre a implementação das funcionalidades da aplicação a desenvolver:

A estimativa da similaridade entre conjuntos (i.e., entre lista de contactos de 2 utilizadores, na Opção 2, entre 2 vectores de caracteres, na Opção 3, e entre conjuntos de interesses de cada utilizador, na Opção 4) tem de ser obrigatoriamente implementada por um método *MinHash*.

Na Opção 2, pode adaptar a implementação que efectuou na secção 4.3 deste guião (PL04). O número adequado de funções de dispersão k pode ser escolhido de acordo com as conclusões que retirou nessa altura.

Na Opção 3, deve desenvolver um método *MinHash* adequado à similaridade entre vectores de caracteres escolhendo de forma fundamentada tanto o tamanho dos *shingles* como o número adequado de funções de dispersão k (sugere-se que experimente tamanhos de *shingle* entre 3 e 6 caracteres). A verificação da existência de um nome terá de ser implementada usando um filtro de Bloom com parâmetros adequados ao problema (e devidamente fundamentados).

Na Opção 4, desenvolva um método *MinHash* para a similaridade entre conjuntos de vetores de caracteres.

Requisitos para a implementação em Matlab

É obrigatório desenvolver 2 scripts Matlab (para além das funções necessárias). O primeiro corre uma única vez para ler os dois ficheiros de entrada e guardar em ficheiro todas as estruturas de dados associadas aos utilizadores e aos filmes, incluindo a matriz com os vectores *MinHash* de cada utilizador (de suporte à Opção 2), a matriz com os vectores *MinHash* de cada nome (de suporte à Opção 3) e a matriz com os vectores *MinHash* associados ao conjunto de interesses de cada utilizador (de suporte à Opção 4).

O segundo script começa por ler do disco todas as estruturas previamente guardadas pelo primeiro script e depois implementa todas as interacções com o utilizador descritas anteriormente.

Avaliação do trabalho:

1. Opção 1 a funcionar corretamente **(máximo 3 valores)**
2. Opção 2 a funcionar corretamente **(máximo 4 valores)**
3. Opção 3 a funcionar corretamente **(máximo 5 valores)**
4. Opção 4 a funcionar corretamente **(máximo 5 valores)**
5. Fundamentação/avaliação das opções tomadas na implementação dos métodos probabilísticos (exemplos: número de funções de dispersão, tamanho de *shingles*, dimensionamento do filtro de Bloom) **(máximo 2 valores)**
6. Qualidade do relatório **(máximo 1)**

[3] Esta opção é independente do ID do utilizador actual.