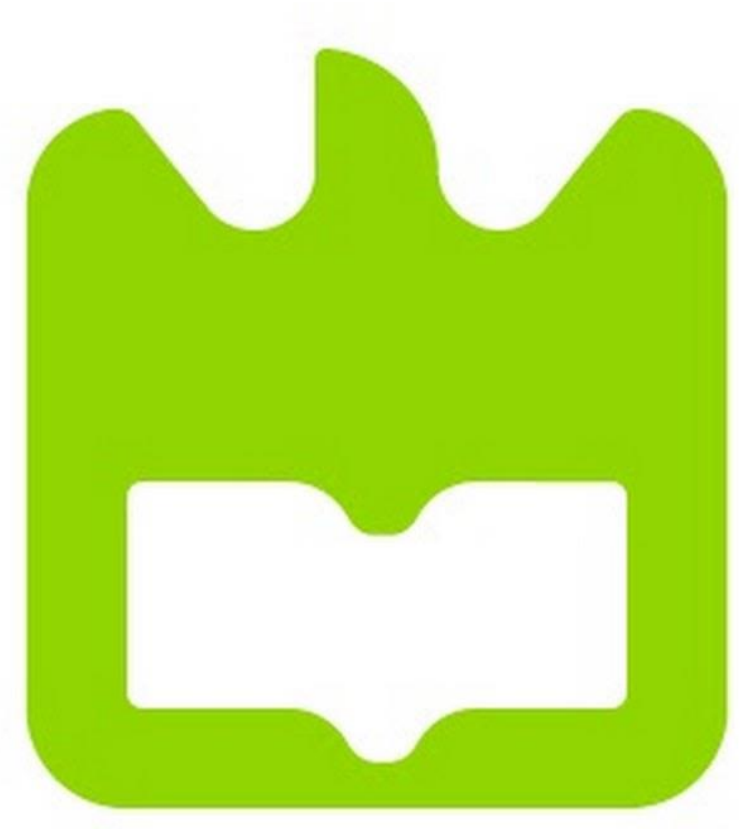
Universidade de Aveiro

MPEI 2020/21 Turma P1

3º guião para avaliação

Joaquim Andrade nº93432

Francisco Silva nº93400



Writing.m

Leitura dos dados

Carregámos ambos os ficheiros "u.data" e "u.item.txt". Retiramos então a primeira e segunda coluna do primeiro e uma tabela do segundo.

```
data= load('u.data');
us= data(1:end, 1:2); clear data;

filename = 'u_item.txt';
d = readtable(filename, 'Delimiter', '\t');
```

Retirar os dados necessários

Manipulámos os dados obtendo o Set que contém os Ids dos utilizadores juntamente com os ids dos filmes que os mesmos assistiram.

```
movies1= d(1:end,1);
movies1= table2cell(movies1);
movies1= movies1';

movies = unique(us(:,2));
id= unique(us(:,1));
Nid= length(id);
Set= cell(height(d),1);
nhash=1000;
for n=1:Nid
    ind= find(us(:,1)== id(n));
    Set{n} = [Set{n} us(ind, 2)];
end
[minh, hash,a, c]= filmesMinHash(movies1, 1000);
min= minHash(id, movies, Set, height(d));
```

Obtivemos também duas funções minhash que seriam usadas na aplicação. Fizemos ambas em duas funções diferentes. A função filmesminhash usa os nomes dos filmes, transforma-os em shingles, obtém o hashcode e com este forma o minHash que devolve juntamente com a e c. Optámos por fazer k-shingles de tamanho 3, pois os filmes possuem um tamanho bastante pequeno e só assim obteríamos um número satisfatório de shingles.

```
function [minh,hash, a, c]= filmesMinHash(filmes, nhash)
    j=1;

    shinglesl=cell(length(filmes),50);
    shingles=cell(1,1);
    x=0;
    for i=1:length(filmes)
        k=3;
        l=1;

        while(l<length(filmes{1,i})-k)
            shingles{i,l}= extractBetween(filmes{1,i},l,l+k);
            %shingleslength{i,l}=j;
            l=l+1;
            j=j+1;
        end
    end

    p=10000019;
    hash=zeros(length(shingles),nhash);
    a=randi([2,p-2],1,1682);
    c=randi([2,p-1],1,1682);


    for j=1:length(shingles)

hash(j,:)=mod(a(1,1:nhash)*sum(double(lower(shingles{j,1}{1})))+
c(1,1:nhash),p);
    end

    minh=Inf(length(filmes),nhash);
    l=1;

    for j=1:length(filmes)
        k=1;
        while k<max(sum(cellfun('length',shinglesl),2)) &&
not(isempty(shinglesl{j,k}))
            minh(j,:)=min(hash(shinglesl{j,k},:), minh(j,:));
            k=k+1;
            l=l+1;
         end
    end

end
```

A função minhash por sua vez não recorre aos shingles, devolvendo apenas o hashcode para ser mais tarde ser usado na aplicação.

```
function [ minh ] = minHash(users, movies, Set, size)


    p=10000019;
    hash=zeros(length(movies),1000);
    for i=1:1000

        a=randi([2,p-2]);
        c=randi([2,p-1]);
        hash(:,i)=mod(a*movies+c,p);
    end


    minh=zeros(size,length(users));

    for i=1:100
        for u=1:length(users)
            minh(i,u)=min(hash(Set{users(u)},i));
        end
    end
end
```

Escrita num novo ficheiro após manipulação dos dados

Para escrevermos estes dados usámos a tabela lida do ficheiro “u_item.txt” e adicionámos as colunas dos dados que necessitaríamos mais tarde.

```
d.min= min;
d.set= Set;
d.minh= hash;
d.a=a';
d.c=c';
writetable(d, "ola.txt");
```

app.m

Na app começámos por retirar novamente os dados, já processados:

```
d= readtable("ola.txt");% lê a tabela previamente escrita
moviesgenre= d{:,2:20};% retira
movies= d{:,1};          % nomes dos filmes
Nid=943;                 %quantidade de utilizadores
minend= 20+Nid;          %começo da minhash na tabela
userend= 1700;
t= d{1:100, 21:1700}; %minhash
Set= cell(Nid,1);        %cell com utilizadores e respetivos
filmes
for n=1: Nid

    Set{n}=[d{n,minend+1:userend}];


end
minhash= d{:,userend+1:end-2};
a= d{1:1000, end-1};
c= d{1:1000, end};
minhash= minhash';
```

Estando os dados lidos, começa a interação com o utilizador. Após lhe ser pedido o id, verificado por um simples while que só avança com um id válido, segue para as 4 opções: Ver os filmes assistidos, procurar sugestões, encontrar um filme e sair.

```
prompt= "Insert User ID (1 to 943):";
x= input(prompt);
while(x <1 || x >943)                %espera por um input dentro
do número de utilizadores
        fprintf("\nOut of bounds\n");
        prompt= "Try again:";

        x= input(prompt);
end

prompt="\n1 - Your Movies\n2 - Get Suggestions\n3 - Search
Title\n4 - Exit Select\n choice:";
y= input(prompt);
```

A opção de sair foi implementada com um while que espera um input do utilizador que seja igual a 4 para acabar a aplicação. Qualquer número que não esteja entre 1 e 4 pedirá um novo input por parte do utilizador.

```
while y~=4          %enquanto o input 4 não aparecer não
acaba
    while(y <1 || y >4)%rejeita qualquer input não desejado
        fprintf("\nOut of bounds\n");
        prompt= "Try again:";

        y= input(prompt);
    end
```

A primeira opção foi implementada com um for que percorre a célula do Set(previamente processado) associada ao id do utilizador, e com os ids dos filmes que o utilizador viu, retira os nomes da lista de filmes(processada anteriormente) e dá display aos mesmos.

```
if y==1     %lista os filmes do utilizador
        array= Set{x};
        array = array(~isnan(array))';
        for n=1: length(array)
            disp(d{array(n), 1});

        end
```

A segunda opção imprime no ecrã os diversos géneros de filmes contidos na base de dados, e após a escolha do utilizador, utiliza a minhash processada no ficheiro writing, para encontrar a distância entre o utilizador e o utilizador e os restantes utilizadores. Após processar todas as distâncias, encontra o utilizador mais similar com ele e retira-lhe os filmes com o género previamente escolhido pelo utilizador dando display aos mesmos.

```
prompt= "1- Action, 2- Adventure, 3- Animation, 4-
Children's\n5- Comedy, 6- Crime, 7- Documentary, 8-
Drama\n9- Fantasy, 10- Film-Noir, 11- Horror, 12-
Musical\n13- Mystery, 14- Romance, 15- Sci-Fi, 16-
Thriller\n17- War, 18- Western\nSelect choice:";
        k= input(prompt);

        sim=zeros( 1,length(t(1,:)));
        for i=1:length(t(1,:))
            sim(1,i)=sum(t(:,k)==t(:,i))/length(t(:,1));
        end
        sim(k)=0;
        [M,I] = max(sim);
        array= Set{I};
        array = array(~isnan(array))';
```

```
        for n=1: length(array)
            if(moviesgenre(array(n), k+1)== 1)
                disp(d{array(n),1});
            end
        end
```

Por fim a terceira opção que após receber o input do utilizador (string), utiliza a função searchfilme que transforma o texto input pelo utilizador numa minHash. Após comparar as distâncias todas entre a minhash criada pelo input e a processada no ficheiro writing irá dar display a 5 delas ou àquelas que satisfizerem a distancia mínima.

```
elseif y==3 %search engine de acordo com o input
        prompt= "\nSearch title :";
        title= input(prompt, 's');
        [s] = searchfilme(title, minhash, a , c);
        if (~isempty(s(1,:)))
            disp("\nFilmes com nome mais parecido: ");
            for j=1:5
                disp(movies1(1,s(1,j)));
            end
         else
            disp("Filme não encontrado");
        end
function [ notw] = searchfilmes( t, minh, a , c)

    threshold=0.3;

    [minh2]=filmeSingleMinHash(t, a, c);
    sim=minHashSingleDist(minh, minh2 );

    notw=zeros(2,length(find(sim>=threshold)));
    for i=1:length(find(sim>=0.2))
        m=find(sim==max(sim));
        notw(1,i)=m(1,1);
        notw(2,i)=max(sim);
        sim(notw(1,i))=0;
    end
end


function [ minh] = filmeSingleMinHash( str,a,c )
    shinglesname=cell(1,1);
    k=2;
    l=1;
    while(l<length(str)-k)
            shinglesname{l,1}= extractBetween(str,l,l+k);
```

```
            l=l+1;
    end
    k=1000;
    hash=zeros(length(shinglesname),k);
    p=10000019;
    for j=1:length(shinglesname)

hash(j,:)=mod(a(1,:)*sum(double(lower(shinglesname{j,1}{:})
))+c(1,:),p);
    end
    minh=Inf(k,1);
    for i=1:k
        for j=1:length(shinglesname)
            minh(i,1)=min(hash(j,i), minh(i,1));
        end
    end

end

function [ sim ] = minHashSingleDist( minh, minh2 )
    for i=1:length(minh(1,:))

sim(1,i)=sum(minh(:,i)==minh2(:,1))/length(minh(:,1));
    end

end
```

Conclusão

A criação desta app foi feita numa corrida contra o tempo, pois, embora tenhamos reservado bastante tempo para a mesma, necessitámos de rever bastante matéria e sentimos as dificuldades que advêm do ensino á distância. Não conseguimos então concluir a terceira opção da aplicação. A primeira e a segunda, embora não eficientemente, foram concluídas com sucesso.