Avaliação PL04

Métodos Probabilísticos para Engenharia Informática

Turma P8

Francisco Cardita 97640

Pedro Ferreira 98620

Ano letivo 2022/2023

Introdução

Na realização deste guião, foram criados dois scripts principais, um de inicialização (init.m) e outro que age como *main* (trabalho2.m). O primeiro tem de ser executado antes do segundo, já que a sua função é ler os ficheiros “u.data” e “u_item.txt” (encontrados no mesmo diretório que o resto dos ficheiros) e guardar em estruturas de dados as informações neles contidas, relevantes para a aplicação.

Do “u.data” guardam-se apenas as três primeiras colunas, contendo ID do *user*, ID do filme e avaliação do filme pelo utilizador, respetivamente. Após esta inicialização, pode executar-se o outro script, que, após a introdução do ID do *user* pretendido, permite selecionar opções entre ver os filmes correspondentes ao respetivo *user* (usando um *Counting Bloom filter*), sugestões de filmes baseados noutros utilizadores ou filmes já avaliados pelo utilizador e procura do nome de um filme. As três últimas opções recorrem a um método *MinHash*.

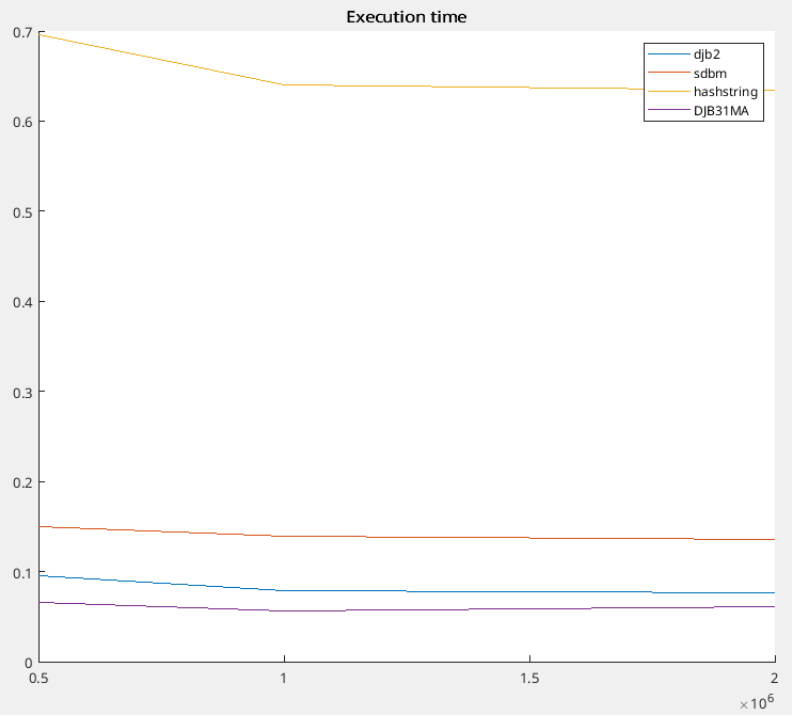
Opção 1

Para listar os títulos dos filmes do utilizador atual, foi usada a função “display_movies”, que acede ao *cell array* “reviews” para retornar os filmes avaliados por esse *user* e, através do *Counting Bloom Filter*, verificar o número de vezes que cada filme foi avaliado por todos os utilizadores. O valor do k ótimo do *Bloom Filter* foi determinado recorrendo à fórmula apresentada nos slides da disciplina (onde se encontra a devida dedução):

$$k = \frac{n \,.\, \ln(2)}{m}$$

Para o filtro, foi usada a função de *hash* “DJB31MA”, usada na aula prática, com um conjunto de *seeds* geradas aleatoriamente.

Esta *hash function foi escolhida pela* sua eficiência, não comprometendo a sua aleatoriedade (ver figura abaixo).



Opção 2

Para determinar os três utilizadores mais similares ao atual (com avaliação maior ou igual a 3), recorreu-se à função “similar_users”, que tem como *inputs* o ID de utilizador e a *MinHash*. Para a *MinHash*, o valor k utilizado foi 100, isto porque este valor foi o que permitiu a obtenção de melhores resultados.

Após o retorno do conjunto de utilizadores similares, é utilizada a função “get_users_suggestions”, que cria uma lista, à qual vai adicionando a uma lista todos os filmes avaliados pelos users similares que não tenham sido avaliados pelo user.

Resta, então, ir buscar os títulos dos filmes através dos ID’s e dar *display*.

Opção 3

Neste caso, as sugestões são baseadas nos filmes que o utilizador já avaliou (uma vez mais com avaliações iguais ou superiores a 3), portanto, usa-se uma função que aceita como parâmetros: ID de utilizador, os seus filmes, *MinHash* (desta vez usando os géneros de filme, para cada filme) e o “threshold” da distância de Jaccard entre os filmes que o user avaliou e os restantes.

Para cada filme do user, é criado um conjunto de filmes em que a distância de Jaccard seja inferior a 0.9.

No fim, é contado o número de vezes que cada filme aparece em todos os conjuntos encontrados e devolvido os 2 mais frequentes.

Nesta opção, não se usou um *Counting Bloom Filter*, porque os conjuntos de filmes gerados dependem do *user* ID escolhido inicialmente, não sendo código que possa ser executado num ficheiro de inicialização à parte.

Opção 4

Por fim, dado um uma certa string, a aplicação devolvia os 5 nome de filmes mais similares à mesma.

Para tal, foi usado um MinHash, em que a assinatura de cada filme se baseava na hashing de shingles (divisão de uma string em várias strings do mesmo tamanho).

Após o user inserir uma string válida, essa string é repartida e transformada em shingles, com o mesmo tamanho de shingle usado no MinHash. A seguir, cada shingle foi passado pela função hash, e para cada função hash foi guardado o valor mínimo obtido (método usado na criação de todos os MinHashes).

Com isto, temos a assinatura da string inserida pelo utilizador, que se compara com todas as linhas do MinHash, para obter a similaridade.

Por fim, retiramos os 5 IDs de filmes mais similares à pesquisa, e damos display dos respetivos nomes.

Nota Final

É importante ter em conta que, na opção 4, os valores de hash obtidos não são verdadeiramente aleatórios, pelo que os resultados da pesquisa são inválidos (são devolvidos sempre os mesmos resultados, independentemente da pesquisa).