universidade de aveiro deti departamento de electrónica, telecomunicações e informática

Relatório
Algoritmos Probabilísticos

Métodos Probabilísticos para Engenharia Informática
Relatório Trabalho 1 - Turma P4
Prof. Amaro Sousa
2022/23

Maria Rafaela Alves Abrunhosa, 107658
Matilde Moital Portugal Sampaio Teixeira, 108193

Índice

Introdução

Neste relatório foram realizadas as questões referentes à seção de avaliação sobre Algoritmos Probabilísticos.

O objetivo deste guião deste 3º guião tem como foco desenvolver uma aplicação em MATLAB com funcionalidades de um sistema online de disponibilização de diversos filmes. Este projeto consistia em a partir de um ficheiro [u.data], previamente disponibilizado, onde foi nos atribuídos uma lista de utilizadores (que vão ser identificados por um ID) e uma lista de filmes (também identificados por um ID) e de um ficheiro [u_item.txt], também previamente disponibilizado, também nos era fornecido as informações referentes a cada filme visto por cada utilizador.

Desenvolvimento da Aplicação

Como pretendido, criámos dois *scripts*: um para armazenar os nossos dados - **read_files.m** - e outro para implementar a aplicação em si - **main_test.m**. Isto porque, ao separarmos as duas componentes, tornamos o código mais simples de ser executado, uma vez que pudemos facilmente carregar os dados necessários para o *script* da aplicação utilizando o comando *load* e o save para salvar as variáveis utilizadas.

Menu

Nesta parte do código, este pede ao utilizador para inserir um ID (que deve estar entre 1 e 943). Em seguida, verifica se o ID é válido. Se o ID não for um número ou estiver fora do intervalo válido (entre 1 e 943), o utilizador é notificado e é pedido que insira um novo ID. Este processo é repetido até que o utilizador insira um ID válido.

O código usa a função input para ler a entrada do utilizador como uma string, e depois usa a função str2double para converter a string para um número. A função isnan é usada para verificar se o número é válido. Se o número for válido e estiver no intervalo esperado, a variável validate é definida como 1, o que faz com que o ciclo while seja interrompido.

Seleção de Opções

Se o ID for válido, é usado um comando switch para determinar qual ação deve ser executada. As opções incluem exibir os filmes avaliados pelo utilizador atual, exibir filmes sugeridos para o utilizador baseado em outros utilizadores similares, exibir filmes sugeridos para o utilizador baseado em títulos similares, e realizar uma pesquisa por títulos de filmes.

Para cada opção, são executadas as ações apropriadas, como calcular a similaridade entre utilizadores, exibir filmes sugeridos, ou pesquisar títulos de filmes.

Option 1

Quando a opção 1 é selecionada, a função my_movies vai ser executada. Esta função exibe os filmes avaliados pelo utilizador com o ID que foi inserido previamente. Esta função recebe três argumentos de entrada:

Id: o ID do utilizador cujos filmes devem ser exibidos.
movies_ids: um vetor de cell arrays, onde cada cell array contém os IDs dos filmes avaliados por um utilizador específico.
dic_movies: um dicionário que mapeia os IDs dos filmes para os títulos dos filmes.

A função começa por percorrer os IDs dos filmes avaliados pelo utilizador, usando um loop for. Em cada iteração, o ID do filme atual é exibido juntamente com o título do filme, usando o dicionário dic_movies.

O operador ' é usado para transpor o vetor movies_ids{id}, o que significa que cada iteração do loop será um vetor linha em vez de um vetor coluna.

Option 2

Caso a opção 2 seja selecionada, esta vai sugerir filmes para o utilizador com o ID especificado, baseando-se em outros utilizadores similares. A função recebe quatro argumentos de entrada:

id: o ID do utilizador para o qual as sugestões de filmes devem ser feitas.
dic_movies: um dicionário que mapeia os IDs dos filmes para os títulos dos filmes.
movies_ids: um vetor de cell arrays, onde cada cell array contém os IDs dos filmes avaliados por um utilizador específico.
minhash_movies: uma matriz de MinHash dos filmes avaliados pelos utilizadores.
A função começa calculando a similaridade entre o utilizador atual e todos os outros utilizadores, usando a matriz de MinHash. Em seguida, os utilizadores são classificados de acordo com a similaridade e os dois utilizadores mais similares são selecionados.

Em seguida, são comparados os filmes avaliados pelo utilizador atual com os filmes avaliados pelos dois utilizadores mais similares. Qualquer filme avaliado pelos utilizadores similares mas não pelo utilizador atual é adicionado à lista de sugestões. Por fim, a lista de sugestões é exibida ao utilizador.

Option 3

Quando é acionada a opção 3, este irá selecionar os filmes com distância de Jaccard inferior a 0.8 (para géneros cinematográficos), e que ainda não tenham sido visualizados pelo utilizador

Assim é utilizada a função minhash para filmes do conjunto de dados "u.data", separando-os pelo id de cada usuário. Além disso, ele lê os filmes por id do arquivo "films.txt" e armazena os ids dos filmes avaliados por cada usuário em um cell array chamado "movies_ids".

Depois, o código realiza a função minhash para os filmes avaliados pelos usuários e armazena o resultado em uma matriz chamada "minhash_movies", com o número de linhas igual ao número de usuários e o número de colunas igual a "k2", que é um parâmetro definido no início do código e representa o número de funções hash a serem usadas.

Por fim, o código realiza a função minhash para os gêneros de cada filme e armazena o resultado em uma matriz chamada "minhash_genres", com o número de linhas igual ao número de filmes e o número de colunas igual a "k3", que é outro parâmetro definido no início do código e também representa o número de funções hash a serem usadas.

Tudo o descrito anteriormente é utilizado no read_files. No ficheiro main é realizada função sug_users. Esta deve ter 3 argumentos de entrada:

id: id do utilizador atual
dic_movies: um cell array com informação sobre os filmes, incluindo o seu título.
movies_ids: um cell array que contém as listas de ids dos filmes avaliados por cada utilizador.
Ela imprime sugestões de filmes para o utilizador atual com base na avaliação de filmes por utilizadores similares.

Para determinar os utilizadores similares, a função calcula a distância de Jaccard entre o utilizador atual e cada um dos outros utilizadores. A distância de Jaccard é definida como o número de itens distintos que os dois conjuntos têm em comum dividido pelo número total de itens distintos entre os dois conjuntos.

Depois de calcular a distância de Jaccard entre o utilizador atual e cada um dos outros utilizadores, a função ordena os utilizadores por ordem crescente de distância. Os dois utilizadores com menor distância são considerados os mais similares. A função imprime então os títulos dos filmes avaliados pelos dois utilizadores mais similares, mas que ainda não foram avaliados pelo utilizador atual.

Opção 4

Neste passo, a função minHashTitles é usada para calcular o minHash de uma série de títulos de filmes. A função recebe como entrada os títulos dos filmes (titles), o número de funções hash a serem utilizadas (numHash) e o tamanho dos shingles (shingleSize). A função retorna uma matriz (matrizMinHashTitles) com o minHash dos títulos dos filmes, onde cada linha corresponde a um título e cada coluna corresponde a uma função hash.

No ficheiro main, esta função faz uma busca de títulos de filmes similares a um título de entrada. Ela usa minHash para calcular a distância entre o título de entrada e os títulos de filmes já conhecidos, e retorna os títulos mais similares.

A função começa calculando o minHash do título de entrada, armazenando os resultados em minHashSearch. Em seguida, ela chama uma outra função chamada filterSimilar, passando como parâmetro o valor de threshold, os títulos de filmes conhecidos, a matriz de minHash dos títulos

de filmes, o minHash do título de entrada e o número de funções hash. Essa função deve retornar um vetor de títulos de filmes similares, um vetor de distâncias e um inteiro k que representa o número de títulos encontrados.

Se k for igual a 0, a função imprime "No results found" (nenhum resultado encontrado). Se k for maior que 5, k é definido como 5. Em seguida, a função converte o vetor de distâncias de cell para matriz, classifica as distâncias e imprime os títulos de filmes mais similares, junto com as respectivas distâncias.

Foi também utilizada a função filterSimilar que foi usada para filtrar títulos de filmes que são considerados similares a um título de filme de pesquisa dado. É dado um limiar de similaridade, threshold, e os títulos de filmes, titles, juntamente com a sua representação MinHash, matrizMinHashTitles, e a representação MinHash do título de pesquisa, minHash_search. O número de funções hash utilizadas é dado por numHash.

A função retorna um conjunto de títulos de filmes similares, similarTitles, as distâncias entre os títulos de filmes similares e o título de pesquisa, distancesTitles, e o número de títulos de filmes similares encontrados, k.

Para cada título de filme, a distância é calculada como 1 menos a proporção de elementos iguais entre a representação MinHash do título de filme e a representação MinHash do título de pesquisa. Se a distância for menor que o limiar de similaridade, o título de filme é considerado similar e é adicionado à lista de títulos de filmes similares.

Além disto foi ainda tentado realizar a implementação de um Counting bloom filter, com funções para adicionar elemento e para fazer o filtro, porém este ficou inacabado devido à falta de tempo.

Conclusão

Ao longo da resolução deste trabalho, deparámo-nos não só com diferentes problemas como também alguns sucessos. Posto isto, podemos afirmar que melhorou a nossa capacidade de resolução de problemas uma vez que nos permitiu a procura de soluções distintas que melhor satisfizesse o objetivo do trabalho.

Conseguimos compreender a importância de MinHash, dos k utilizados, Distâncias de Jaccard, Shingles entre outros conceitos.