Sistema de Recomendação de Filmes com Abordagem Content-Based com TD-IDF

Luiz Henrique de Melo Santos**
luiz.melo@dcc.ufmg.br
Departamento de Ciência da Computação, UFMG
Belo Horizonte, Minas Gerais, BR

ABSTRACT

Com o avanço dos serviços de streaming e o aumento das vendas de produtos e serviços pela internet, a exigência por sistemas que promovam uma recomendação personalizada de itens para seus usuários tem se tornado cada vez maior. Existem várias arquiteturas que possibilitam suas implementações, cada uma com suas especificidades quanto às formas de abordagem dos dados que são utilizados. Neste Trabalho Prático foi elaborado um sistema recomendador de itens com abordagem de análise content-based, e com uso de TD-IDF para a extração de features de colunas contendo linguagem natural.

KEYWORDS

recommendation system, content-based, td-idf, datasets, feature engineering, feature extraction

1 INTRODUÇÃO

Em certos ambientes ou ferramentas, o desenvolvedor possui acesso a várias informações acerca do itens que devem ser recomendados, ou não, para um determinado usuário, e nestes casos elas podem ser utilizadas a fim de otimizar a análise dos itens e a acurácia do recomendador. Por garantir uma excelente abstração destas informações, a recomendação com abordagem Content-Based é fortemente utilizada nestes casos.

O objetivo deste Trabalho Prático é a modelagem, desenvolvimento e implementação de um sistema recomendador com abordagem Content-Based que consiga predizer, com a maior acurácia possível, a avaliação que um determinado usuário dará para um determinado item. O desenvolvimento contou com uma extensa extração e tratamento de features existentes, a fim de torná-las legíveis pelo algoritmo de análise de similaridades, que faz uso da fórmula de *Similaridade do Cosseno*.

*Matrícula: 2017014464

A extração de features é feita por meio de uma análise e tratamento manual de cada uma delas, com uso de técnicas como One-Hot Encoding, transformação para features numéricas, e TF-IDF.

2 MODELAGEM

Boa parte da extração de features contou o tratamento de dados categóricos. O uso de One-Hot Encoding permitiu a criação de várias fatures numéricas a partir delas, por meio da criação de features binárias que indicam a presença ou não dela em determinado item do dataset. Nas demais features categóricas foi feito uma categorização simples dos dados - como por exemplo na feature '*Language*', na qual foram atribuídos os seguintes valores:

- '1' para '*English*';
- '2' para '*French*';
- '3' para '*Japanese*';
- '4' para '*Spanish*';
- '0' para as demais línguas, por possuírem um número extremamente distribuído de ocorrências.

Nas features contendo elementos de linguagem natural foi utilizado o algoritmo de TF-IDF (*Term Frequency – Inverse Document Frequency*) (1), que faz uso da frequência com que um determinado token (palavra) é utilizado combinado com a frequência inversa deste mesmo token em todo o dataset. A fórmula que permite esta análise é formulada por meio das operações de *Term Frequency* (2) e *Inverse Document Frequency* (3).

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{1}$$

$$tf(t,d) = f_{t,d} \div \sum_{t' \in d} f_{t',d} \tag{2}$$

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{3}$$

A estrutura de dados utilizada para a representação dos itens e de suas features foi de matriz densa, a fim de facilitar a implementação

do algoritmo e reduzir o tempo de desenvolvimento empregado. Um dos pontos negativos desta representação é o consumo de memória requerido, fazendo com que não seja possível o uso de todas as features extraídas, sendo limitadas em um número máximo de 15.000 vindas do TF-IDF e 34 vindas de fatures categóricas, entretanto tem como ponto positivo um maior desempenho, uma vez que o tempo de acesso a seus elementos é reduzido.

3 ALGORITMO

A avaliação da similaridade foi feita por meio da fórmula de *Similaridade do Cosseno* (4). Como todos os valores de similaridades entre os vetores dos itens apresentaram pouquíssima variação (todas próxima de 0.9999), foi necessária a aplicação de um método customizado (5) para a predição das classificações que serão dadas pelos usuários. Este método se resume na média ponderada entre as avaliações dadas por um determinado usuário, no qual os pesos são as similaridades obtidas entre cada um dos itens já avaliados pelo usuário com o item alvo fornecido. Este método de predição permitiu uma ótima redução do RMSE (de 0.4 unidades) observado no algoritmo.

$$cossim(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{4}$$

$$pred(i, U) = \frac{\sum_{j \in U} [cossim(i, U_j) \cdot U_i]}{\sum_{j \in U} cossim(i, U_j)} \tag{5}$$

Nos casos de *cold-start*, que é quando um usuário ainda não está presente na base de dados, é retornado o valor presente na feature '*imdbRating*', uma vez que ela já é a média de várias avaliações feitas externamente por vários usuários, bem como apresenta um intervalo semelhante ao observado nas predições que devem ser feitas. Nos casos em que um item do '*cold-start*' não possui dado para '*imdbRating*' é retornado o valor padrão de 6.53 (valor que apresentou o menor RMSE dente os testados no intervalo [6.5-6.8] com variações de 0.01).

4 EXPERIMENTAÇÃO

O algoritmo final utilizado para a geração da submissão deste Trabalho foi executado em máquinas do CRC-DCC-UFMG, com as seguintes configurações: Intel Core i7 4ª geração, RAM 16Gb DDR3 166MHz, HD 1TB e sistema operacional Ubuntu 16.04.

Os tempos de execução obtidos com os treinamento oscilaram no intervalo de 2:10min-2:40min, entretanto sem nunca ultrapassar a marca de 3:00min. A métrica de avaliação utilizada para a determinação da eficiência do algoritmo foi o RMSE (Root Mean Square Error). Em testes iniciais foram feitas predições sobre a própria base de dados de *ratings*, de forma a facilitar a análise do comportamento do algoritmo.

Em decorrência do uso de matriz densa, o algoritmo pode exigir uma grande quantidade de memória RAM disponível, entretanto ela fica limitada a valores inferiores a 9GB, o que permite a sua execução nas máquinas do CRC sem maiores problemas.

5 CONCLUSÃO

Por meio deste Trabalho Prático foi possível a abstração, modelagem e desenvolvimento de um sistema de recomendação de filmes, bem como os problemas e dificuldades associados à análise Content-Based.

Grande parte do tempo empregado nele foi gasto para o processo de tratamento dos dados, extração de features e pré-processamento da matriz resultante. Entretanto isto permitiu com que as etapas posteriores pudessem ser finalizadas em pouco tempo e com um nível de complexidade reduzido.

6 TRABALHOS FUTUROS

Uma possível otimização que pode ser feita neste projeto é a simplificação e otimização na extração e tratamento das features, o que poderia reduzir consideravelmente o tempo de execução do algoritmo. Além disto também pode ser feita a implementação do uso de uma matriz esparsa, o que permitiria uma redução do uso da memória e, consequentemente, permitiria a adição de todas as features disponíveis extraídas por meio do TF-IDF das features com linguagem natural.

Outra melhoria que pode ser feita é a alteração da fórmula de similaridade utilizada para a *Similaridade de Jaccard*, uma vez que ela promoveu uma considerável redução do RMSE global, possivelmente pelo melhor desempenho com esta base de dados específica. Entretanto esta similaridade possui uma relação inversamente proporcional na fórmula de predição utilizada, ou seja devemos considerar (6).

$$pred(i, U) = \frac{\sum_{j \in U} [\frac{1}{jacsim(i, U_j)} \cdot U_i]}{\sum_{j \in U} \frac{1}{jacsim(i, U_j)}} \tag{6}$$

Este formulação não foi utilizada neste Trabalho pois o tempo de execução deste algoritmo é muito superior ao tempo limite permitido.