PA #2 - Content-based Movie Recommendation

Ramon Gonçalves Gonze

14 de maio de 2018

Resumo

Este trabalho apresenta um recomendador de filmes baseado em conteúdo. O objetivo é predizer notas que usuários dariam para filmes que ainda não assistiram. Através de um dataset contendo várias informações sobre filmes como gênero, atores e diretores, é calculada a similaridade entre os gostos do usuário e o filme em questão, e então é predita uma nota.

1 Introdução

A abordagem *content-based* é utilizada principalmente em situações de *cold-start*, onde não se tem nenhuma histórico de avaliações do item. Para o contexto de predições de notas para filmes, é calculada a similaridade entre as preferências de filmes do usuário e o filme em questão. As seções 2.1 e 2.2 apresentam o modo como esses vetores são construídos. A métrica utilizada para avaliação do recomendador é o *RMSE* (Root Mean Square Error).

A Seção 5 apresenta os resultados de diversas combinações de características utilizadas dos filmes, e seus respectivos erros alcançados. Por fim a última seção conclui o trabalho e apresenta as dificuldades encontradas.

2 Modelagem

Há 3 matrizes principais que representam os dados:

- Filme X Descrição
- Usuário X Descrição
- Usuário X Filme (histórico de notas de usuários para filmes)

A descrição de cada filme são informações sobre gêneros, atores, diretor(es), roteirista(s), o país de produção do filme e o seu enredo. A descrição do usuário é a combinação de todos os filmes que este já assistou e deu uma nota maior que 4. Está sendo considerado que as características dos filmes com nota menor ou igual a 4 não são relevantes, pois elas não representam o gosto do usuário. Seja F o conjunto de filmes, U o conjunto de usuários e R_{uf} a nota que um usuário u deu para um filme f (presente na matriz Usuário X Filme).

2.1 Vetores de filmes

Os vetores que representam um filme f são $\vec{f_g}$ para gêneros, $\vec{f_{adr}}$ para atores, diretor e roteirista e $\vec{f_p}$ para países. Todos esses vetores são binários, onde, no vetor de gêneros por exemplo, haverá 1 se o filme é daquele gênero ou 0 caso contrário. A Tabela 1 contém um exemplo de representação do vetor de gêneros.

Já para o enredo de um filme, o vetor $\vec{f_e}$ não é binário. Cada dimensão do vetor é uma palavra em todo o conjunto de palavras possíveis que existem nos enredos de todos filmes. Para atribuir o valor de cada palavra presente no enredo do filme, foi utilizado o método *TF-iDF*, onde $TF(p, f)$ é a frequência da palavra p no enredo do filme f, e $iDF(p)$ é a frequência inversa da palavra p em todos os filmes, isto é

$$iDF(p) = log\left(\frac{|F|}{|F|_p}\right) \tag{1}$$

onde $|F|_p$ é o número de filmes onde p aparece em seu enredo.

	f_{g1}	f_{g2}	...	$f_{g\|F\|}$
Terror	1	1		1
Drama	0	0		1
Comédia	1	0		0
Aventura	0	0		0
...				

Tabela 1: Representação do vetor de gêneros de cada filme. A coluna f_{gi} representa o vetor de gêneros do i-ésimo filme.

2.2 Vetores de usuários

Os vetores dos usuários são combinações dos vetores de todos os filmes que ele viu e avaliou (com nota maior que 4, como já justificado anteriormente). Similarmente aos filmes, há vetores binários $\vec{u}_g$, $\vec{u}_{adr}$ e $\vec{u}_p$ para um usuário u. Cada um desses três vetores é a soma dos vetores dos filmes que u assitiu. Por exemplo, o vetor $\vec{u}_g$ é definido por

$$\vec{u}_g = \vec{f}_{g1} + ... + \vec{f}_{gi} \tag{2}$$

onde $\vec{f}_{gi}$ é o vetor de gênero do i-ésimo filme que o usuário u assistiu. A equação (2) também se aplica para os vetores $\vec{u}_{adr}$ e $\vec{u}_p$.

Já para o vetor de enredo $\vec{u}_e$, foi utilizada a recomendação de Rocchio. O vetor é definido por

$$\vec{u}_e = \frac{1}{|R_u|} \sum_{f \in R_u} R_{uf} \cdot \vec{f}_e \tag{3}$$

onde R_u é o conjunto dos filmes avaliados por u.

2.3 Similaridade entre filmes e usuários

Em relação aos vetores binários, seja t a menor quantidade de dimensões com o valor 1, ou seja, $t = min(|\vec{u}|^2, |\vec{f}|^2)$. Para vetores de gênero por exemplo, t é o filme/usuário que é classificado pela menor quantidade de gêneros. Seja b a quantidade de gêneros de filmes existentes (ou seja, o valor máximo para algum $|\vec{v}_g|^2$). A similaridade de dois vetores de gêneros $sim(\vec{u}_g, \vec{f}_g)$ onde $\vec{u}_g = (u_{g1}, ..., u_{gb})$ e $\vec{f}_g = (f_{g1}, ..., f_{gb})$, é definida por

$$sim(\vec{u}, \vec{f}) = \frac{\sum_{i=1}^{b} u_{gi} \cdot f_{gi}}{t} \tag{4}$$

A similaridade descrita na equação (4) é bem semelhante ao $cos(\vec{u}, \vec{f})$, porém, ao se utilizar o cosseno, a similaridade entre um filme e um usuário será distorcida. Por exemplo, suponha que o filme f_1 é do gênero terror, e o filme f_2 é do gênero terror, halloween e suspense. O valor de $cos(f_{g1}, f_{g2}) = 0,57$. Intuitivamente, percebe-se que $f1$ e $f2$ são muito similares, porém o cosseno

nos diz que eles são somente 57% semelhantes. Já a similaridade $sim(f_{g1}, f_{g2}) = 1$, sendo assim adotada como função de similaridade. O mesmo raciocínio também se aplica para o vetor de países e para o vetor de atores, diretor e roteirista.

Para a similaridade de dois vetores $sim(\vec{u}_e, \vec{f}_e)$ sendo $\vec{u} = (u_1, ..., u_n)$ e $\vec{f} = (f_1, ..., f_n)$ onde n é o número total de palavras distintas, foi utilizada a distância euclidiana normalizada. Portanto, $sim(\vec{u}_e, \vec{f}_e) = d(\vec{u}_e, \vec{f}_e)$, e

$$d(\vec{u}_e, \vec{f}_e) = \sqrt{\sum_{i=1}^{n} \frac{(u_i - f_i)^2}{n}} \tag{5}$$

2.4 Predição da nota

Para realizar a predição da nota $p(u, f)$ de um usuário u para um filme f, calcula-se as similaridades entre os quatro vetores de ambos, e depois calcula-se a média ponderada das similaridades de acordo com pesos G (gêneros), ADR (atores, diretor e roteirista), P (países) e E (enredo). Portanto

$$p(u, f) = \frac{sim(\vec{u}_g, \vec{f}_g) \times G + sim(\vec{u}_{adr}, \vec{f}_{adr}) \times ADR + sim(\vec{u}_c, \vec{f}_c) \times C + sim(u_p, f_p) \times P}{G + ADR + C + P} \tag{6}$$

O valor de $p(u, f)$ retornará um número real no intervalo $[0, 1]$. Como a nota de um usuário para um filme varia de 0 a 10 (para o dataset utilizado neste trabalho), a nota final retornada será $p(u, f) \times 10$.

O único caso onde não é possível realizar a predição acima é quando o usuário em questão não avaliou nenhum filme. Neste caso a nota retornada será o IMDb Rating (descrito na Seção 5) do filme.

3 Análise de Complexidade

O programa principal faz primeiramente a leitura do conteúdo dos filmes. Para cada filme lido, sua descrição é processada, e são construídos os vetores dos filmes. Supondo que a descrição de gêneros, atores, diretores e roteiristas, país de produção e enredo não é maior que 10^4 caracteres cada uma, o processamento da descrição de cada filme é constante. Logo a complexidade da leitura dos conteúdos dependerá somente da quantidade de filmes, sendo assim $O(|F|)$.

Após a leitura dos conteúdos, é feita a leitura da matriz Usuário X Filmes. Seja r o total de avaliações. No pior caso, teremos $r = |U| \cdot |F|$, quando todos os usuários avaliaram todos os filmes. Portanto, a complexidade será $O(|U| \cdot |F|)$.

Por fim, o programa principal faz a leitura dos pares `usuário:filme`, os quais deve-se predizer a nota. Seja m a quantidade de pares lidos, u e f o usuário e filme de algum par `usuário:filme`. Para cada par, são calculadas as similaridades dos quatro vetores de u com os quatro vetores de f. Para os vetores binários, o cálculo da função $sim()$ depende da quantidade de dimensões dos vetores, ou seja, a quantidade de gêneros distintos, atores, diretores e roteiristas distintos e países distintos. Claramente a quantidade de atores, diretores e roteiristas é superior. Sendo a essa quantidade, gasta-se $O(a)$ para cacular a similaridade. Já o vetor de enredo, serão feitas n operações, que é o limite do somatório da equação 4. A complexidade do cálculo de similaridades será então $O(|U| \cdot |F| + a + n)$.

A complexidade final do programa principal é dada pela quantidade de pares m e o gasto para predizer a nota para cada par. Sendo assim, a complexidade é $O(m \cdot (|U| \cdot |F| + a + n))$.

4 Dataset e Execução

O dataset utilizado neste trabalho se consiste em três arquivos no formato CSV:

- content.csv, contendo 22.080 pares *(filme,descrição)*
- ratings.csv, contento o histórico de notas com 336.672 tuplas *(usuário,filme,nota)*

- targets.csv, contendo 77.276 pares *(usuário,filme)* para predição

O trabalho foi implementado em C++, utilizando a biblioteca padrão e uma biblioteca externa, a **rapidjson**[1], para tratar os dados em JSON do arquivo `content.csv`. Há um arquivo principal denominado *recommender*, que possui a função principal do programa, e mais um módulo denominado *predict*, que possui a implementação das funções para a predição de uma nota. Há um *Makefile* que compila o programa e gera um executável `recommender`. O algoritmo recebe três parâmetros por linha de comando:

```
$./recommender <arquivo_conteúdo> <histórico_de_notas> <pares_perdições>
```

Exemplo de execução do programa:

```
$./recommender content.csv ratings.csv targets.csv
```

A saída do programa é feita pela saída padrão **stdout**.

5 Experimentos

Foram realizados testes variando os valores de G, ADR, P e E da equação (6). Quando $E = 0$, ou seja, não considerar a similaridade entre os enredos, foi alcançado um erro menor do que quando $E > 0$. A causa desse resultado pode ser a não utilização de algoritmos mais sofisticados de processamento de linguagem natural para tratar o enredo de cada filme. As únicas técnicas aplicada para tratar os enredos foi a remoção de *stop words*, que são palavras como artigos e pronomes que não são relevantes para comparar textos e a utilização do método *TF-iDF*.

A melhor combinação de pesos encontradas foi $G = 50$, $ADR = 25$, $P = 25$ e $E = 0$. O valor do RMSE para essa combinação foi de **1.86944**.

Uma outra característica disponível no conteúdo de cada filme é o *IMDb Rating*, que é a média das notas de todos os usuários que assistiram aquele filme e o avaliaram no IMDb. Ou seja, podemos tratar essa nota como a popularidade do filme. Ao realizar uma média ponderada da nota predizida por $p(u, f)$ (com peso 2) e o IMDb Rating de f (com peso 8), foi alcançado um RMSE de **1.71746**. Com isso pode se dizer que levar a popularidade de um filme em consideração para predizer uma nota é uma opção viável.

6 Conclusão

Pode-se concluir com esse trabalho que a abordagem *content-based* consegue alcançar bons resultados, porém, é de consenso da comunidade científica que a filtragem colaborativa é mais efetiva quando se possui histórico de avaliações. Contudo, como a filtragem colaborativa não trata problemas de cold-start, quando por exemplo um filme acaba de ser lançado no catálogo, a abordagem content-based é bem-vinda. Um sistema híbidro que implementa ambas obterá um melhor resultado. A principal dificuldade encontrada foi a de modelagem do problema, onde foi preciso testar diferentes funções (como por exemplo a de similaridade) para encontrar o melhor resultado possível.

[1] https://github.com/Tencent/rapidjson