Trabalho prático 1: Collaborative Movie Recommendation

Harlley Augusto de Lima

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas

harlley@dcc.ufmg.br

1. Introdução

Nesse trabalho prático é proposto um sistema de recomendação baseado em filtragem colaborativa (FC) aplicado no contexto de filmes. Para tanto, é implementada a filtragem colaborativa baseada em similaridades entre usuários e na similaridades entre itens.

Na filtragem colaborativa baseada em usuários, os *ratings* dados pelos usuários similares ao usuário alvo servem como base para realizar as predições. Com o objetivo de determinar os usuários similares ao usuário alvo, a similaridade entre esses deve ser computada. Em seguida, essa similaridade é utilizada para ponderar a agregação dos *scores*, gerando a predição final. Por outro lado, também foi implementada a filtragem colaborativa baseada em item. Nessa abordagem, o objetivo é fazer recomendações para o item alvo e o primeiro passo é determinar o conjunto de itens que são similares ao item alvo. Da mesma forma que FC baseada em usuário, é necessário implementar uma métrica de similaridade a ser utilizada para identificar a similaridade entre os itens. Em seguida, esse valor de similaridade é utilizado para ponderar a predição final do item alvo.

Nesse trabalho foi implementado um sistema de recomendação colaborativa de filmes. Para tal, foi implementada a FC baseada em item e baseada em usuário, sendo que a implementação baseada em item alcançou melhores resultados no sistema Kaggle[1]. A seguir, na Seção 2 são apresentados os detalhes de implementação juntamente com a análise de complexidade, na seção Seção 3 a avaliação experimental do método. Por fim, na Seção 4 é apresentada a conclusão e as dificuldades encontradas no trabalho.

2. Implementação

Essa seção mostra os detalhes de implementação dos componentes da FC baseada em item e em usuários. Para a implementação do sistema de recomendação foram feitas duas implementações. Como a matriz de utilidade é esparsa, a primeira implementação feita foi baseada na estrutura `hash`. Ou seja, inicialmente a matriz de utilidade era implementada em `hash`. Entretanto, tal implementação se mostrou lenta e excedia o

[1] https://www.kaggle.com/

tempo total de cinco minutos imposto na especificação. Dessa forma, essa implementação não foi considerada, mas pode ser acessada no repositório do presente trabalho[2].

Assim, a matriz de utilidade é implementada como uma matriz bidimensional e os demais detalhes da implementação serão mostrados a seguir.

2.1 FC baseada em itens

Para o desenvolvimento dessa abordagem os seguintes componentes devem ser implementados: métrica de similaridade, agregação da avaliação, normalização dos dados e outros componentes adicionais. Dessa forma, as estruturas utilizadas e a análise de complexidade de cada componente são detalhados a seguir. Para a análise de complexidade, considere que existam m usuários e n itens.

Métrica de similaridade Conforme mencionado anteriormente, nessa abordagem é necessário computar a similaridade entre os itens. Como apresentado em (1), a métrica de similaridade que apresenta melhores resultados é o cosseno. O maior problema com essa abordagem é que usuários podem prover *ratings* com escalas diferentes. Por exemplo, um usuário pode avaliar a maioria dos itens com valores altos, enquanto outros podem avaliar a maioria de forma negativa. Portanto, nesse trabalho é utilizado a métrica *adjusted cosine*, em que é subtraída a média de *score* do usuário de seus *ratings*. Assim, a similaridade entre o item a e b pode ser calculada conforme definida na Equação 1.

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r_a})(r_{b,p} - \bar{r_b})}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r_a})^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r_b})^2}} \tag{1}$$

A ordem de complexidade para calcular a similaridade dos itens é $O(m)$.

Hash de itens Para o cálculo do *adjusted cosine*, é necessário saber quais usuários avaliaram os itens para os quais se deseja calcular a similaridade. Do contrário, o tempo para calcular a similaridade de dois itens seria muito alto, pois seria necessário multiplicar todas as linhas das duas colunas que representam os itens. Assim, é criado um `hash` de usuários para armazenar a lista de usuários que avaliaram um determinado item. Dessa forma, apenas serão multiplicados na computação da similaridade os *ratings* dados pelos usuários que avaliaram os itens que estão sendo comparados. A estrutura de `hash` foi utilizada por prover um acesso rápido à lista de usuários que avalariam o item.

Escolha da vizinhança Para a escolha dos itens a serem utilizados para o cálculo da predição, foram escolhidos os k itens mais similares de acordo com o *adjusted cosine* ao item que se desejava fazer a predição para o usuário alvo. Dessa forma, afim de obter

[2]Branch master com a implementação baseada em `map`:
https://github.com/harlleyaugusto/collaborativeMovieRecommendation

resultados mais satisfatórios torna-se necessária a variação da quantidade k de itens. A complexidade da escolha da vizinhança é de $O(n)$ para ordenar o vetor de similaridade.

Matriz de utilidade Como mostrado na Equação 2, a todo momento é necessário acessar o *rating* dado por um usuário em um determinado item. Para facilitar tal acesso, a matriz de utilidade é implementada com uma matriz bidimensional. Além disso, para computar a similaridade é necessário subtrair a média dos *score* do usuário com o valor de *score* que ele deu para o item. Dessa forma, a matriz de utilidade é criada com tal subtração já realizada, e não com o valores reais de *score*.

Por fim, foi necessário mapear os usuários e os itens para linhas e colunas da matriz. Para tal, foi criado um identificador que varia entre 0 e a quantidade total de usuários na base, sendo mapeados nas linhas da matriz. De forma similar, foi criado um identificador de 0 a quantidade total de itens na base para mapear os itens nas colunas. A estrutura de matriz provê um acesso rápido aos valores de *ratings*, visto que para acessar tais valores basta o identificadores do item e do usuário.

Agregação das avaliações Para agregar as avaliações de cada item, foi utilizada a média ponderada para o cálculo da predição final. Assim, a ponderação da nota de cada item é feita utilizando a similaridade do item avaliado posteriormente pelo usuário alvo com o item que se deseja fazer a predição. Assim, a predição de um item p para um usuário a é dada pela Equação 2.

$$pred(a,b) = \bar{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \bar{r_b})}{\sum_{b \in N} |sim(a,b)|} \tag{2}$$

A complexidade final para a predição final é $O(m^2 n)$, que corresponde a calcular a similaridade de todos os itens e agregar as avaliações.

Matriz de similaridade Para evitar que a similaridades de itens fossem computadas repetidamente, foi criada uma matriz para armazenar as similaridades já computadas. Essa matriz é quadrática e tem como dimensões a quantidade de itens total. Assim, para obter a similaridade de dois itens, é verificado se a similaridade de tais itens já não foi calculada e armazenada na matriz de similaridade. Caso não tenha sido calculada, o cálculo é feito e posteriormente armazenada na matriz de similaridades. Especificamente, essa matriz reduziu o tempo de execução do sistema, pois evitava que a similaridade de dois itens fosse calculada repetidamente. Além disso, o acesso a matriz era rápido, visto que necessitava apenas dos identificadores de cada item.

Como a FC baseada em itens é uma abordagem de ordem quadrática, a criação dessa matriz diminui o tempo de processamento da predição final.

Com a matriz de similaridade	Sem a matriz de similaridade
1m25.090s	2m54.018

Tabela 1. Tempo de execução com e sem a matriz de similaridade, considerando toda a vizinhança.

2.2 FC baseada em usuário

A abordagem de FC baseada em usuário é similar à abordagem baseada em item. A diferença direta é que a similaridade é calculada entre o usuário alvo e os demais usuários que avaliaram o item que se deseja calcular a predição. Sendo assim, as mesmas estruturas apresentadas anteriormente foram utilizadas nessa abordagem, com a exceção que o *adjusted cosine* é calculado entre usuários. Além disso, foi necessário criar um *hash* para armazenar a lista de itens avaliados por cada usuário. Tal estrutura agiliza a computação de similaridade, pois não é necessária a multiplicação de todas as colunas dos usuários que estão sendo comparados. Ou seja, a computação da similaridade considera apenas os itens avaliados em comum pelos dois usuários.

Visto que essa abordagem é semelhante à baseada em item, a complexidade é também semelhante, sendo um algoritmo de ordem quadrática.

3. Resultados

Nessa seção são apresentados os resultados referentes a abordagem baseada em item, visto que tal implementação obteve melhor resultado no Kaggle.

Na implementação do sistema, foi proposta a utilização da matriz de similaridade para evitar a computação de similaridades de itens já calculados anteriormente. A Tabela 1 mostra o tempo de execução do algoritmo com a utilização da matriz de similaridade e sem a matriz de similaridade, considerando toda a vizinhança. Como pode ser visto, a diferença entre o tempo de execução é cerca de 90 segundos a menos utilizando a matriz de similaridade, trazendo uma melhora significativa.

A Tabela 2 mostra o tempo de execução do sistema a medida que o número k de itens semelhantes aumenta, tanto com a matriz de similaridade quanto sem tal matriz. Comparando as duas colunas fica claro que a abordagem que utiliza a matriz de similaridade possui sempre um tempo de processamento menor. Além disso, a medida que k aumenta, o tempo de execução do sistema se mantém praticamente constante, utilizando a matriz de similaridade. Tal fato também ocorre sem a utilização da matriz, ainda que com k igual a 120 o tempo de execução aumenta quase 16 segundos em relação a k igual a 1.

Em relação a qualidade das predições obtido no Kaggle, o melhor resultado obtido foi com a abordagem baseada em item, que alcançou RMSE igual a 2.30395. Ainda tentando melhor os resultados, foram feitas variações no tamanho k da vizinhança, não acarretando em melhora significativa. Diante de tal resultado não satisfatório, foi implementada a FC baseada em usuário, que também não melhorou a qualidade das predições.

k	Com a matriz de similaridade (s)	Sem a matriz de similaridade (s)
1	74.006	168.470
5	74.196	168.179
10	74.857	175.885
15	74.861	170.551
30	74.386	164.456
60	73.206	171.873
90	78.649	169.900
120	83.895	184.972

Tabela 2. Tempo de execução com e variando o tamanho k da vizinhança.

4. Conclusão e dificuldades

Nesse trabalho foi desenvolvido um sistema de recomendação de filmes. Para tal, foram desenvolvidas duas abordagens: filtragem colaborativa baseada em item e baseada em usuário. Sendo que a abordagem baseada em item obteve melhores resultados de acordo com os resultados apresentados no Kaggle. Para implementação de tal sistema as seguintes dificuldades foram encontradas:

1. **Implementação utilizando C++.** A falta de conhecimento pleno na linguagem tornou o tempo de implementação ainda maior. Em particular, a passagem de parâmetro padrão nessa linguagem é por cópia. Isso deixava a execução do sistema desenvolvido bastante lento. Entretanto, passando os parâmetros por referência diminuiu consideravelmente o tempo de execução. Esse ponto atrasou consideravelmente o desenvolvimento do trabalho.
2. **Otimização do código.** Na primeira versão do sistema, a matriz de utilidade foi implementada utilizando `hash`. Entretanto, tal implementação demorava cerca de 8 minutos para gerar todas as predições do arquivo de entrada. Assim, foi implementada uma nova versão em que a matriz de utilidade é implementada com uma matriz bidimensional, reduzindo o tempo de processamento. Isso também atrasou a implementação do sistema.
3. **Otimização dos resultados.** A primeira abordagem implementada foi a FC baseada em item. Como tal, abordagem não apresentou bons resultados no Kaggle, foi também implementada a FC baseada em usuário, que também não obteve bons resultados. Diante disso, foram feitos vários esforços para melhorar a qualidade das predições que não surtiram efeitos.

Apesar dos resultados das predições não serem satisfatórios, com esse trabalho foi possível passar por todas as etapas de desenvolvimento da abordagem colaborativa baseada em item e em usuário, sendo elas: seleção da vizinhança, agregação dos *ratings* e normalização dos dados. Além disso, foi feito um esforço considerável para manter o tempo de execução baixo e, conforme apresentado, o tempo de execução considerando toda a vizinhança na abordagem de FC baseada em item é de cerca 1m25s.

Referências

1 JANNACH, D. et al. *Recommender Systems: An Introduction*. 1st. ed. New York, NY, USA: Cambridge University Press, 2010. ISBN 0521493366, 9780521493369.