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

Para esse trabalho prático é implementado um sistema de recomendação baseado em filtragem colaborativa (FC). Para tanto, é implementado a filtragem colaborativa baseada em na similaridades de usuários e na similaridades de itens.

Na filtragem colaborativa baseada em usuários, a avaliação dada por usuários similares ao usuário alvo servem como base para serem feitas para o usuário alvo. Com o objetivo de determinar os usuários similares ao usuários alvo, sua similaridade com os demais usuários é computada. Em seguida, essa similaridade é utilizada para ponderar a predição final. Por outro lado, a filtragem colaborativo baseada em item para fazer as recomendações para o item alvo, o primeiro passo é determinas 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 utilizada para identificar as similaridade entre os itens. Em seguida, esse valor é utilizado para ponderar a predição final para o item alvo.

Esse trabalho tem foi implementado um sistema de recomendação colaborativa de filmes. Para tal, foi implementado o 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 Implementação são apresentados os detalhes de implementação juntamente com a análise de complexidade e na seção Resultados são apresentados os resultados da implementação.

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 utilizada é esparsa, a primeira implementação feita era baseada na estrutura `map`. Ou seja, a matriz de utilidade era implementada em `map`. Entretanto, tal implementação se mostrou lenta e excedia o tempo total de cinco

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

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

Assim, é a matriz de utilizada é implementada como uma matriz de bidimensional. Os detalhes da implementação serão mostrados a seguir.

2.1 FC baseada em itens

Para implementação dessa abordagem os seguintes componentes devem ser implementados: métrica de similaridade, agregação da avaliação e normalização dos dados. Dessa forma, cada componente e estruturas utilizadas nessa abordagem são detalhados a seguir.

Métrica de similaridade Conforme apresentado anteriormente, nessa abordagem é necessário computar a similaridade entre os itens. Como apresentado em (2), 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 mais altos, enquanto outros podem avaliara a maioria de forma negativa. Para isso nesse trabalho é utilizado a métrica *adjusted cosine*, em que é subtraída a média do usuários de seus *ratings*. A similaridade entre o item a e b pode ser calculada conforme definida na Equação 1.

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

Agregação das avaliações Para agregar as avaliações de cada item, foi utilizada a média ponderada para o calculo 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.

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

Escolha da vizinhança Para a escolha dos itens a serem utilizados para o calculo da predição, foram escolhidos os k itens mais similares ao item que se desejava fazer a predição para o usuário alvo. Dessa forma, afim de obter resultados mais satisfatórios torna-se necessária a variação de k.

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, como

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

para computar a similaridade é necessário subtrair a media de score do usuário com o valor de score que ele deu para o item, a matriz de utilidade é criada com tal diferença, e não com o valores reais de score.

Além disso, foi necessário mapear os usuários e os itens para linhas e colunas da matriz. Para tal, foi criado um identificador que variava entre 0 e quantidade total de usuários na base, para mapear os usuários. De forma similar, foi criado um identificador de 0 a quantidade total de itens na base, para mapear os itens.

Hash de itens Para o calculo do *adjusted cosine*, é necessário saber quais usuários avaliaram os itens que se deseja calcular a similaridade. Do contrário, o tempo para calcular a similaridade de dois itens seria muito alto, pois seria necessário a multiplicação de todas as linhas das duas colunas que representam os itens. Assim, é criado um `hash` de itens para armazenar a lista de usuários que o avaliaram.

Matriz de similaridade Para evitar que a similaridades de itens fossem computadas repetidamente, foi criada uma matriz para armazenar as similaridades computadas. 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 calculo é feito e armazenado na matriz.

2.2 FC baseada em usuário

A abordagem de filtragem colaborativa baseada em usuário é similar à abordagem baseada em item. A diferença direta é que a similaridade é calculada entre usuário 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 ...

Map de usuários

3. Resultados

4. Dificuldades

Referências

1

2 D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York, NY, USA: Cambridge University Press, 1st ed., 2010.