Programming Assignment #2
Item-based Recommendation

Felipe Moraes Gomes[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

felipemoraes@dcc.ufmg.br

1. Introdução e Definição

O objetivo desta trabalho é implementar um sistema de recomendação baseado em item (Item-based Recommendation) simples para filmes. Esse sistema é baseado em filtragem colaborativa. Neste sistema os usuários indicam atráves de avaliações, o quanto eles gostam de terminados items. Avaliando as avaliações conhecidas, o sistema implementado neste trabalho prevêem qual será a avaliação de um usuário para um item ainda não avaliado.

2. Implementação e Organização

2.1. Estrutura de dados e Algoritmos

Este trabalho foi implementado utilizado a linguagem de programação Java e várias estruturas de dados padrões dessa linguagem foram utilizados durante o trabalho. Três classes principais foram implementadas Indexer, Recommender e Evaluator. A seguir os atributos e os métodos utilizados nas classes principais são descritas e também uma análise de complexidade computacional para os principais métodos.

2.1.1. Class Indexer

Esta classe é a primeira principal classe implementada para desenvolver este trabalho. Segue abaixo os métodos implementados nessa classe.

Algorithm 1: Métodos classe Indexer

average; similarity;

O primeiro método calcula em $O(n)$ a média dos ratings de n itens de um usuário. O método similarity calcula em $O(n)$ a similaridade entre dois items, sendo n o número de usuários.

2.1.2. Class Recommender

Esta classe é a segunda principal classe implementada para desenvolver este trabalho. Ela possui apenas o método principal, em que carrega os arquivos de entrada e calcula a predição de ratings para cada item não classificado pelo usuário chamando um método da classe ItemBasedRecommender. A complexidade computacional desse método é $O(N * M * K)$ onde N é o número de usuários e M o número de itens, K é a complexidade do método predictingItemRating descrito a seguir.

2.1.3. Class ItemBasedRecommender

Esta classe é chamada segunda principal classe implementada para desenvolver este trabalho. Ela possui apenas um método em que carrega as similaridades calculadas com a princimeira classe principal Indexer e um método que calcula a predição de um rating para um usuário. A complexidade computacional desse método é $O(K)$ onde K é o número de itens na vizinha do item que pretende-se predizer o rating.

Algorithm 2: Métodos classe ItemBasedRecommender

predictingItemRating;
itemsSimilarities;

2.1.4. Class Evaluator

Esta classe é a terceira principal classe implementada para desenvolver este trabalho. Ela possui apenas um método principal que chama um método da classe AveragePrecision e imprimi o resultado na saída padrão.

2.1.5. Class AveragePrecision

Esta classe é chamada pela terceira principal classe implementada para desenvolver este trabalho. Ela possui dois métodos principal, o primeiro calcula a métrica Average Precision para um usuário com complexidade $O(n)$ onde n é o tamanho do raking que deseja considerar. O segundo método apenas chama o primeiro m usuários tendo complexidade $O(n * m)$.

Algorithm 3: Métodos classe AveragePrecision

getAPForUser;
getMAP;

2.1.6. Class LoadInput

Esta classe apenas realiza o carregamento dos dados dos arquivos de entrada. A análise de complexidade dessa classe não será analisada.

2.2. Fluxo de Execução

O fluxo de execução do sistema de recomendação implemetado é dado da seguinte forma. Primeiramente é chamado o primeiro programa Indexer onde as similaridades entre os items são calculados. Em o programa Recommender é chamado e calculado para as predições dos ratings dos usuários. Após esse calculo, o programa Evaluator pode ser chamado para avaliar os resultados preditos dado um arquivo com os dados reais das predições.

3. Controle e E/S

3.1. Execução e Compilação

O programa foi compilado com Java 8 e executado em uma máquina MacOS 10.9.5 utilizando a IDE Eclipse versão Luna. A seguinte linha pode ser usada para compilar este trabalho:

- $ java -jar tp2.jar Indexer ¡users¿ ¡items¿ ¡ratings¿
- $ java -jar tp2.jar Recommender ¡users¿ ¡items¿ ¡ratings¿
- $ java -jar tp2.jar Evaluator ¡ground-truth¿ ¡output-recommendations¿

  java -jar tp1.jar Recommender users items ratings

4. Resultados e Discussão

Este trabalho foi amplamente revisado utilizando o livro-texto [Jannach et al. 2010] e transparências da disciplina.

4.1. Tempo de Execução

Comparando ao trabalho prático anterior, em que foi implementado um User-based Recommendation, o tempo de execução para o recomendado foi de um minuto e 2 segundos. Como o trabalho implementado aqui computa primeiramente as similaridades entre os itens, essa primeira fase teve tempo de execução de 12 minutos. Em seguida, o recomendador utilizando a primeira fase prediz os itens recomendados para os usuários em tempo 41 segundos. Foi utilizado um k com valor de 100. Em resumo, o recomendador x foi melhor que o recomendador y.

4.2. Mean Average Precision

A métrica Mean Average Precision (MAP) [1] foi utilizada para avaliar a eficácia dos recomendadores implementados até agora. Utilizando a saída do recomendador baseado em usuário, o MAP encontrado foi 0.30223. Já para o recomendador implementado neste trabalho o MAP de 0.02863. Como podemos observar, o MAP do primeiro recomendador foi muito melhor que o MAP do segundo recomendador, o que pode ser um sinal que a métrica foi implementada de forma errada e que até o momento da entrega deste trabalho o aluno não conseguiu encontrar o erro de implementação.

5. Conclusão

Este trabalho teve como objetivo a familiarização com um dos mais famosos algoritmos de recomendação utilizados hoje. Esse algoritmo é baseado em filtragem colaborativa.

Os objetivos foram alcançados, mas várias melhorias na implementação poderiam ser realizadas. Foram utilizadas Hashmaps para guardar as avaliações dos usuários sobre os itens. Essa ainda não é uma boa estrutura de dados devido que faz uma busca. Uma estrutura de matriz simples poderia ter sido utilizada que tem acesso O(1) sempre.

Para os próximos trabalhos pretende-se realizar uma avaliação do sistema implementado com várias métricas vistas em sala de aula.

References

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

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