Item-based Collaborative Filtering Movie Recommendation

Hsing-Wen Hsu hh2916

1 Overview & Background

Item-based Collaborative Filtering is an algorithm that computes item similarity based on the ratings or interactions of items by users. With this algorithm, one will be able to predict the products that a user might like. Hence, it is widely used in recommendation systems. Large companies such as Amazon use Item-based Collaborative filtering to recommend products to their users.

To compute item-item similarity, a utility matrix is involved. In this example, there are 6 movies and 12 users, and we are trying to predict the rating of movie 1 by user 5. (Note: Some movies are not rated):

$i\backslash u$	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Steps to predict rating of movie 1 by user 5

Let S_{ij} be the similarity between movie i and j, r_{xj} be the rating of user x on movie j, $N(i;x)$ be the items rated by user x that are similar to i. The predicted rating for item i by user x is:

$$r_{xi} = \frac{\Sigma_{j \in N(i;x) S_{ij} \cdot r_{xj}}}{\Sigma_{j \in N(i;x) S_{ij}}} \tag{1}$$

1. Mean-center the rating of each movie. (For movie 1, the mean centered ratings are [-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]

2. Compute $N(1;5)$: Compute the cosine similarity between movie 1 and other movies (S_{1j}), and select the top k movies that are similar to movie 1. (For k = 2, the answers are movie 3 and 6).

3. Compute $r_{5,1}$: Predict movie 1's rating by taking the weighted average of the top k movies similar to movie 1 (Equation 1).

2 Problem Formulation

In this project, I will be using the MovieLens dataset, which provides 100000 ratings (943 users, 1682 movies). The program will take a user id as the input and recommend n movies for each movie that the user has already rated.

Deliverables

1. Implement a sequential version
2. Implement a parallelized version
3. Compare the difference between the two versions

Possible ways to implement parallelism or compare different methods

1. Store the utility matrix in different formats and compare the performance of parallelism among these formats. (Ex. Traditional 2D matrix vs. CSR format).
2. Parallelize the computation of item-item similarity. Computing the similarity between an item and all items sequentially will cost $O(|I||U|)$. Where $|I|$ is the number of items and $|U|$ is the number of users.

References and Resources

1. https://web.stanford.edu/class/cs124/lec/collaborativefiltering21.pdf
2. Item-based Collaborative Filtering :

   Build Your own Recommender System!: https://www.analyticsvidhya.com/blog/2021/05/item-based-collaborative-filtering-build-your-own-recommender-system/
3. Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets
4. G. Linden, B. Smith and J. York, ”Amazon.com recommendations: item-to-item collaborative filtering,” in IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.
5. MovieLens: https://grouplens.org/datasets/movielens/