Spark Lightning-fast cluster computing

Collaborative Filtering Recommendation Algorithm on Apache Spark

Liu Xingbang & Gao Yuan

Introduction

- What can Collaborative Filtering Recommendation Algorithm do?
  - Use information about a user's preferences to make personalized predictions about content, that they might find relevant.
- Challenge of the traditional recommendation algorithm
  - Difficult to compute recommendations quickly and accurately over a large dataset.
- Solution --- Parallel computation of neighborhood-based collaborative filtering algorithm on Spark
  - Allow the algorithm to scale linearly with a growing number of users, i.e. O(n).
  - Include two recommendation approaches-- User-based and Item-based
- Perspective
  - Effective at increasing engagement and purchasing
  - Many of the most heavily trafficked websites employ recommender systems, such as LinkedIn, Amazon, and Twitter

Sequential Algorithm of Collaborative Filtering

Problem Statement

- A list of n items I = {i1,i2,…,in} and a list of k users U = {u1,u2,…,uk}.
- Let M be a n × k matrix, where each M(u,i) represents the rating score of a user u about an item i.
- M(u,i) can either be a real number or missing
- Predict the items that will have the top N rating for a given user $u \in U$
- Based on u's rating scores and the preferences of users with similar interaction histories to u
- Assuming users with similar preferences will have the same rating for the same items

Sequential Algorithm of Collaborative Filtering --- Item-based

Mathematical Formulation

- Step one: Obtain M, user-item ratings matrix.

$$sim(\mathrm{i}_x, \mathrm{i}_y) = \frac{\sum_{u \in P_{i_x, i_y}} r_{u,i_x} r_{u,i_y}}{\sqrt{\sum_{u \in P_{i_x, i_y}} r_{u,i_x}^{\ 2}} \sqrt{\sum_{u \in P_{i_x, i_y}} r_{u,i_y}^{\ 2}}}$$

Formula 1

- Step two: Calculate the similarities between items by Formula 1 .

$$r_{u_x,\mathrm{i}} = \bar{r}_{u_x} + \frac{\sum_{i_y \in R_{u_x,i}} (r_{u_x,i_y} - \bar{r}_{u_x}) sim(\mathrm{i}, \mathrm{i}_y)}{\sqrt{\sum_{i_y \in R_{u_x,i}} sim(\mathrm{i}, \mathrm{i}_y)}}$$

Formula 2

- Step three: Calculate the predicted rating for each item that a given user u ∈ U has not yet rated by Formula 2.

$R(u_x, i)$ **represents the subset of items iy ∈ I other than i that the given user ux have rated**

Sequential Algorithm of Collaborative Filtering --- Item-based

Stages of Coding

1.Obtain the sparse user-item matrix:

user_id -> [(item_id_1, rating_1), (item_id_2, rating_2), ...]

2.Get all item-item pair combos:

(item1,item2) -> [(item1_rating,item2_rating),
(item1_rating,item2_rating), ...]

3.Calculate the cosine similarity for each item pair and select the top-N nearest neighbors:

①(item1,item2) -> similarity

②item1 ->[(item2,similarity), (item3,similarity), ...]

4.Calculate the top-N item recommendations for each user:

user_id -> [item1, item2, item3, ...]