Assignment 1- Collaborative Filtering and Gradient Descent

Shivakanth Thudi

02/01/2017

Part 1.4 - Evaluation

The Root Mean Squared Error and Mean Absolute Error on the testing set when using the full training set is as follows:

RMSE	MAE
0.8899	0.6975

Part 1.6 - The Cold Start Problem

In the scenarios below, we assume that we have a similarity matrix and look at collaborative filtering, but assume that we have no information about properties of items.

New User but Known Movie

We take the average of the ratings given to the known movie across all known users that have rated this known movie, and assign this rating to the new user.

New Movie but Known User

We take the average of all the ratings that this known user has given movies that he has watched, and assign this rating to the new movie.

New User and New Movie

We take the average of all ratings across all known users and all known movies, and use this as a prediction for the rating a new user would give a new movie.

Part 1.7 - Complexity

The complexity of this algorithm is $O(n^2)$.

Assume that there are n users and m movies. When we make a prediction on the rating of user i on movie k, we first look at the set of all users who have rated this movie k. We assume that every user has rated this movie k - this implies that we have to loop through the list of all users, which is of size n. Also, since we are computing similarities on the fly in this algorithm, for each of the users in this list, we need a similarity score between this user and the user i (user we are making the prediction for). For this similarity score, we look at the intersection of all movies these two users have seen, which at worst is all movies - assume this is of size m.

So the complexity is $O(mn)$, or rather, just $O(n^2)$.

Part 1.8 - Collaborative Filtering for Pandora

The algorithm from the previous section does not work because the Pearson-correlation coefficient does not work for implicit ratings. Here we do not have explicit ratings in the utility matrix, but rather 1's and 0's that signify whether or not a user has played a song. If we use the Perason-correlation for similarity, we will find that users who have seen atleast one song in common will always be assigned a similarity score of 1.0, while users who do not have any songs in common will always be assigned a score of 0.0.

For implicit ratings in Pandora, we can use Jaccard Similarity for making recommendations using collaborative filtering.

Pseudocode:

1) Given users i and j, let I_i and I_j be the set of songs that user i and j have played respectively. The similarity score, using Jaccard similarity will then be:

$$w_{ij} = \frac{|I_i \cap I_j|}{|I_i \cup I_j|}$$

2) Predict whether a user will listen to a song: For a particular song (column in utility matrix), we take a weighted sum of the similarity scores - the numerator will be the summation of the similarity scores multiplied by whether or not a user has listened to a song, while the denominator will be the summation of the similarity scores across all users. If this weighted sum is greater than a threshold value, say 0.5, predict that the user will listen to the song and recommend it to him. If it is less than the threshold value, we will not recommend the song to the user.

$$P_{ik} = \frac{1}{\sum w_{ij}} \sum w_{ij} * I_{jk}$$

- If $P_{ik} > 0.5$, predict 1 (user will listen to song), else predict 0 (user will not listen to song). Note that we do not need to use absolute values of the similarity scores here, since Jaccard similarity is between 0 and 1.

$$E(U, V, u^0, v^0) = \frac{1}{N} \sum_{(i,j):r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - u_i \cdot v_j \right)^2$$

$$\frac{\partial E}{\partial u_{ik}} = \frac{-2}{N} \sum_{j:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - \sum_{k=1}^{K} u_{ik} v_{jk} \right) v_{jk}$$

$$\frac{\partial E}{\partial v_{jk}} = \frac{-2}{N} \sum_{i:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - \sum_{k=1}^{K} u_{ik} v_{jk} \right) u_{ik}$$

$$\frac{\partial E}{\partial u_i^0} = \frac{-2}{N} \sum_{j:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - \sum_{k=1}^{K} u_{ik} v_{jk} \right)$$

$$\frac{\partial E}{\partial v_j^0} = \frac{-2}{N} \sum_{i:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - \sum_{k=1}^{K} u_{ik} v_{jk} \right)$$

COMPACT FORM :

$$u_i^{t+1} = u_i^t + \frac{2\eta}{N} \sum_{j:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - u_i^t v_j^t \right) v_j^t$$

$$v_j^{t+1} = v_j^t + \frac{2\eta}{N} \sum_{i:r_{ij}=1} \left(y_{ij} - u_i^0 - v_j^0 - u_i^t v_j^t \right) u_i^t$$

$$u_i^{0\,t+1} = u_i^{0\,t} + \frac{2\eta}{N} \sum_{j:r_{ij}=1} \left(y_{ij} - u_i^{0\,t} - v_j^{0\,t} - u_i^t v_j^t \right)$$

$$v_j^{0\,t+1} = v_j^{0\,t} + \frac{2\eta}{N} \sum_{i:r_{ij}=1} \left(y_{ij} - u_i^{0\,t} - v_j^{0\,t} - u_i^t v_j^t \right)$$