(from Andrew’s lecture)

Content-based recommender systems

Define the model as, when the user click the page, we mark the user are interested in the content,

Content based, Theta is a content preference.

Feature vector Xi for the content( we can see this as an encoder for the object), but how can I get this one????(DRAWBACK)

If we change to collaborative filtering, we can let the algorithm to choose the feature to use to help user filter the content they want.

If we don’t know the feature, we can use CF.

Basic idea if we have theta, we can infer the X.

CF1

In this problem, we have all the theta because we can track user’s preferences as theta. Then the inner product of Theta and X represents the whole recommending score for each user, then we can take the threshold based on the space in the screen to recommend items in the database in descending order.

CF2

Vectorization

LOW Rank factorization, theta is usually low rank, we can save X and Theta separately and computer X\*Theta(transpose) more efficiently in vectorization form using GPU.

Sometimes X might not be human understandable because of the feature learning by computer instead of handcraft feature by people.

For people currently browsing some item, we can recommend similar item by finding the calculating ||xi-xj||.

For the database side, we use mongodb to store all the data. Since mongodb is nosql, It fits the best when store the low rank matrix of predicted ratings of user’s or user’s preference theta in Collaborative filtering. Because we don’t have real user visiting our website, we have to simulate the browsing history.

Definition of the problem:

To fit into the form of Collaborate filtering, we define the time for each user stayed on one item/website in the scale of 5.

0: close the item within 5 seconds.

1:5sec – 1min

2: 1 min – 2 minutes

3: 2 minutes- 3minutes

4:3 minutes – 4 minutes

5: above 4 minutes.

We set users’ preference matrix Theta, and the feature matrix X including all the items we have.

The object is to learn this Theta and X given low rank rating matrix Y, and we want to “fill in” all the missing part in the Y to possibly give each user a recommendation based on this calculation of using X\* Theta to approximate Y.

So our collaborative filtering algorithm is the following:

FC2 3.43

Click and track the users’ staying time can be done in the real world. So we are now focusing on predicting this low rank rating matrix given the simulated data (the real staying time on one item).

Finialy, I use d3.js to show user the recommended item in our database.

Database design:

Our database has three instances: user based information, item based information and the time user staying on one page (rating). (We pushed the rating at runtime to the database)

The following are the definition of the database:

UID name email pw

itemID url avatar

UID ItemID rating

(Underscore means primary key)

We save this file in json format in mongodb.

Work schedule

1. finish collaborative filtering algorithm in python( for the fastest using GPU or spark) future work migrate to gpu )

<http://dato.com/> this might be a good tool,1 distributed,2 restful

input = Low rank matrix Y (for data is a csv file user, item, rating)

output = json file for d3.visualization

1. d3.js to show recommendation
2. api from mongodb to collaborative algorithm
3. from CF to d3.js shows