CSI Project

Single Criteria Recommender System

Presented by:
Anish Agrawal (17ucs024)
Shatishay Jain (17ucs146)
Tarun Gupta (17ucs168)

Recommender System

The recommendation problem can be defined as estimating the response of a user for new items, based on historical information stored in the system, and suggesting to this user novel and original items for which the predicted response is high. Examples : Amazon.com, Movielens.org .

Collaborative Filtering

●Collaborative filtering is a method of making predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).
●Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future.
●A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see.

- Example :

ID	User	Item	Rating
240	u1	m1	2
222	u1	m3	3
256	u2	m1	5
300	u2	m2	2
211	u3	m1	3
278	u3	m2	3
198	u3	m3	1
266	u4	m2	2
200	u4	m3	2

Step 1: Write the user-item ratings data in a matrix form.The resultant matrix is :

	m1	m2	m3
u1	2	?	3
u2	5	2	?
u3	3	3	1
u4	?	2	2

Step 2 : We will use **Cosine Similarity** to calculate Item 2 Item similarity. To calculate similarity between items m1 and m2 .Here, both m1 and m2 have been rated by users u2 and u3. We create two item-vectors, v1 for item m1 and v2 for item m2, in the user-space of (u2,u3) and then find the cosine of angle between these vectors. A zero angle or overlapping vectors with cosine value of 1 means total similarity (or per user, across all items, there is same rating) and an angle of 90 degree would mean cosine of 0 or no similarity. Thus, the two item-vectors would be,

v1 = 5*u2 + 3*u3

v2 = 2*u2 + 3*u3

The cosine similarity between the two vectors, v1 and v2, would then be:

cos(v1,v2) = (5*2 + 3*3)/sqrt[(25 + 9)*(4+9)] = 0.90

Complete Item-to-Item similarity matrix:

Step 3: For each user, we next predict his ratings for items that he had not rated. We will calculate rating for user u1 in the case of item m2(target item). To calculate this we weigh the just-calculated similarity-measure between the target item and other items that user has already rated. The weighing factor is the ratings given by the user to items already rated by him. We further scale this weighted sum with the sum of similarity-measures so that the calculated rating remains within a predefined limits. Thus, the predicted rating for item m2 for user u1 would be calculated using similarity measures between (m2,m1) and (m2,m3) weighted by the respective ratings for m1 and m3: rating = (2 * 0.90 + 3 * 0.86)/(0.90+0.86) = 2.70

Predict Accuracy

1.Mean Absolute Error

$$\mathrm{MAE}(f) = \frac{1}{|\mathcal{R}_{\mathrm{test}}|} \sum_{r_{ui} \in \mathcal{R}_{\mathrm{test}}} |f(u,i) - r_{ui}|,$$

Where f(u,i) is a function defined to predict the rating of item i by user u.

r_{ui} is rating given by user u to item i.R_{test} is test set of ratings.

Mean Absolute Error

```
 RESTART: C:\Users\admin\Desktop\Recommender System\mini project\recommender.py
mae = 0.9589626831620537
mae = 0.9296321191950454
mae = 0.8654252099570696
mae = 0.899057516830418
mae = 0.9503594189176424
avg = 0.9206873896124458
```

Simulation & Result

- We have used MovieLens dataset which consists of 100,000 ratings, assigned by 943 users on 1682 movies.

- We have chosen 50 active users randomly to give recommendation.

- Training and testing dataset are created as 66% and 34% respectively.

- Accordingly Cosine similarity, and prediction formulas are used.

- Finally for movielens 100K dataset Mean Absolute Error ~ 0.9 is calculated.