CSE 8803RS: Recommendation Systems

Lecture 2: Memory-Based Collaborative Filtering

Hongyuan Zha

School of Computational Science & Engineering
College of Computing
Georgia Institute of Technology

Basic Problem Formulation

Rating based paradigm

- Users: $u, v \in \mathcal{U}$; Items: $i, j \in \mathcal{I}$
- Ratings: r_{ui} indicating degree of preference of user u for item j, higher values $\Rightarrow$ stronger preference
- **Problem.** Ratings are not defined over all $\mathcal{U} \times \mathcal{I}$, need to predict those missing ratings
- Incomplete rating matrix

	Casablanc	God Father	Harry Potter	Lion King
David	5	4	2	?
John	3	2	?	5
Jenny	5	2	5	?

The Duality Between Users and Items

- Users and items are *dual* of each other. However, your viewpoint can either be
  — User centric: for a given user with past purchasing and/or rating history, how to recommend new items to her?
  — Item centric: for a given item that was bought and/or rated by some users before, to which other users should we recommend it?
- CF has been exclusively focused on the user-centric viewpoint. Thus the heavy emphasis on item-based methods
- Asymmetry still exists in real-world examples
  — similarity of items more stable than similarity of users

In the User Centric World

Problem. For a given user with past purchasing and/or rating history, how to recommend new items to her?

- User-based methods
  — For a given user, find other similar users, and recommend items those similar users liked in the past
- Scaling issues: complexity $O(MN)$, where M # of users, and N # of items; in practice more like $O(M)$
- Some remedies:
  — sampling users
  — clustering users
  — offline computation of user similarity: frequent change of user activities

User-based Methods

The notion of an active user a, and the prediction for r_{ai}

- For any user u, let $I_u = \{i \mid r_{ui} \neq ?\}$
- Mean user rating:

$$\bar{r}_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui}$$

- Prediction for r_{ai}

$$\hat{r}_{ai} = \bar{r}_a + \kappa \sum_u sim(a,u)(r_{ui} - \bar{r}_u)$$

where u is over the set of neighbors, κ normalization factor

Some Subtle Points

- As is written the set of neighbors is *fixed* independent of the item to be predicted
- The best k neighbors may not even have an opinion about the particular item
- Dynamically select k best neighbors who have rated the item

Similarity

- Correlation:

$$sim(a,u) = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2} \sqrt{\sum_i (r_{ui} - \bar{r}_u)^2}}$$

where the summation is over $i \in I_a \cap I_u$

Extensions

- Default voting: assume default values, and expand the summation over $i \in I_a \cup I_u$ or beyond
  — assume some number of items that both would like/dislike
- Inverse user frequency: down-weight items that appear in many I_u
  — analogous to inverse document frequency in IR
  — many variations on this: $\log(M/M_i), M_i$ # of I_u that item i appeared
- Case amplification: making $sim(a, u)$ more extreme

In the User Centric World

Problem. For a given user with past purchasing and/or rating history, how to recommend new items to her?

- Item-based methods
  — For a given user, find items that are similar to those that the user has purchased or rated, then combines those similar items into a recommendation list
- Offline computation of item similarity: complexity $O(MN^2)$. However, most of the entries will be zero $\Rightarrow$ fast method
- Online look-up of similar items does not depend on M or N
  — but rather how many the user purchased/rated in the past
- Works for user with limited data, even just one item purchase/rating

Fast Method for Item Similarity

Problem. Computing Item Similarity table offline: complexity $O(MN)$ — there might be items bough by most users, but each user only bought a small number of items

```
For each item in product catalog, I_1
  For each customer C who purchased I_1
    For each item I_2 purchased by
       customer C
      Record that a customer purchased I_1
       and I_2
  For each item I_2
    Compute the similarity between I_1 and I_2
```

Evaluation Metrics

r_{ui} vs. $\hat{r}_{ui}$, and $\mathcal{T}$ is the test set

- Root mean squared error (RMSE):

$$\left(\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (r_{ui} - \hat{r}_{ui})^2 \right)^{1/2}$$

- Mean absolute error (MAE):

$$\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |r_{ui} - \hat{r}_{ui}|$$

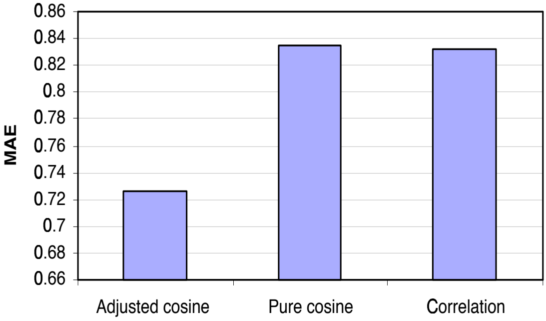
- Metrics based on binary classification

Experiments on MovieLens Data

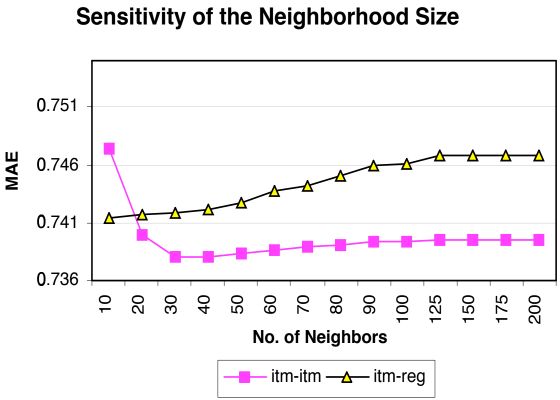
- 43,000 users and 3,500+ movies
  — users with 20+ ratings
  — used 100,000 ratings with a 943×1682 user-item matrix
- Public data: 1 million ratings for 3,900 movies by 6,040 users. About 4% of the ratings are observed. The ratings are integers ranging from 1 (bad) to 5 (good).

Compare Similarity

- Pure cosine for rating vectors
- Correlation
- Adjusted cosine

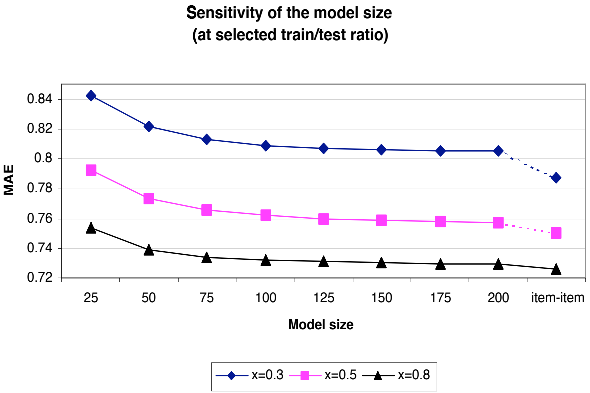


User Neighborhood Size



Item Neighborhood Size

of items to keep in item similarity table



Item-based vs. User-based