Memory-Based Collaborative Filtering

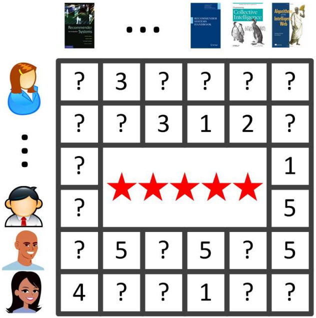
Weike Pan

College of Computer Science and Software Engineering
Shenzhen University

Outline

Recommendation with Explicit Feedback

- We may represent users' explicit feedback in a ***matrix*** form:



- If we can estimate the missing values (denoted as "?") in the matrix, we can make recommendations for each user.
- It's a rating prediction problem.

Assumption and Idea

- "Wisdom of the crowd"
- Users with similar tastes in the past will have similar tastes in the future
  - User-based CF
- A user will like some similar items to those he liked before
  - Item-based CF

Some Questions

- Q1. How to calculate the similarity between two users or items?
  - A1. Similarity Measurement
- Q2. How to select some similar users or items?
  - A2. Neighborhood Selection
- Q3. How to predict the rating based on the information of similar users or items?
  - A3. Prediction Rule

Notations

- $\mathcal{I}_u$ is a set of items rated by user u
- $\mathcal{I}_w$ is a set of items rated by user w
- $\mathcal{U}_j$ is a set of users who rated item j

A1. Similarity Measurement

Pearson correlation coefficient (PCC) between user u and user w,

$$s_{wu} = \frac{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{uk} - \bar{r}_u)(r_{wk} - \bar{r}_w)}{\sqrt{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{wk} - \bar{r}_w)^2}} \quad (1)$$

Notes:

- $-1 \leq s_{wu} \leq 1$

A2. Neighborhood Selection

- Similarity threshold
- Top-K most nearest neighbors
  - Step 1. Obtain the neighbors of user u where $s_{wu} \neq 0$, i.e., $\mathcal{N}_u$
    - In practice, we usually use a large $\mathcal{N}_u$ as candidate users (**instead of all the neighbors**) due to the high space cost
  - Step 2. Obtain the users who rated item j, i.e., $\mathcal{U}_j$
  - Step 3. Obtain a set of top-K nearest neighbors of user u from $\mathcal{U}_j \cap \mathcal{N}_u$ (when estimating the rating of $\hat{r}_{uj}$), i.e., $\mathcal{N}_u^j \subseteq \mathcal{U}_j \cap \mathcal{N}_u$ with $|\mathcal{N}_u^j| = K$

A3. Prediction Rule

Predicted rating of user u on item j,

$$\hat{r}_{uj} = \bar{r}_u + \frac{\sum_{w \in \mathcal{N}_u^j} s_{wu}(r_{wj} - \bar{r}_w)}{\sum_{w \in \mathcal{N}_u^j} s_{wu}} \quad (2)$$

Notes:

- sometimes, we will use the following prediction rule,

$$\hat{r}_{uj} = \bar{r}_u + \frac{\sum_{w \in \mathcal{N}_u^j} s_{wu}(r_{wj} - \bar{r}_w)}{\sum_{w \in \mathcal{N}_u^j} |s_{wu}|}$$

- the default value is $\bar{r}_u$ if $\mathcal{N}_u^j = \emptyset$
- $\mathcal{N}_u^j$ is dependent on both user u and item j

Bug!

- Step 1. Obtain the top-K nearest neighbors of user u, i.e., $\hat{\mathcal{N}}_u$ with $|\hat{\mathcal{N}}_u| = K$
- Step 2. Predict the rating of user u on item j,

$$\hat{r}_{uj} = \bar{r}_u + \frac{\sum_{w \in \mathcal{U}_j \cap \hat{\mathcal{N}}_u} s_{wu}(r_{wj} - \bar{r}_w)}{\sum_{w \in \mathcal{U}_j \cap \hat{\mathcal{N}}_u} s_{wu}}$$

Notes:

- We may have $|\mathcal{U}_j \cap \hat{\mathcal{N}}_u| < K$, and thus it is a bug
- Some references indeed use such a prediction rule ($\times$)

Notations

- $\mathcal{U}_k$ is a set of users who rated item k
- $\mathcal{U}_j$ is a set of users who rated item j
- $\mathcal{I}_u$ is a set of items rated by user u

A1. Similarity Measurement

Adjusted Cosine similarity between item k and item j,

$$s_{kj} = \frac{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uk} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uj} - \bar{r}_u)^2}} \quad (3)$$

Notes

- $-1 \leq s_{kj} \leq 1$
- Cosine similarity between item k and item j

$$s_{kj} = \frac{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uk} r_{uj}}{\sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uk}^2} \sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uj}^2}}$$

A2. Neighborhood Selection

- Similarity threshold
- Top-K most nearest neighbors
  - Step 1. Obtain the neighbors of item j where $s_{kj} \neq 0$, i.e., $\mathcal{N}_j$
    - In practice, we usually use a large $\mathcal{N}_j$ as candidate items (**instead of all the neighbors**) due to the high space cost
  - Step 2. Obtain the items rated by user u, i.e., $\mathcal{I}_u$
  - Step 3. Obtain a set of top-K nearest neighbors of item j from $\mathcal{I}_u \cap \mathcal{N}_j$ (when estimating the rating of $\hat{r}_{uj}$), i.e., $\mathcal{N}_j^u \subseteq \mathcal{I}_u \cap \mathcal{N}_j$ with $|\mathcal{N}_j^u| = K$
  - K is a parameter needs to be tuned, e.g., $K \in \{20, 30, 40, 50, 100\}$

A3. Prediction Rule

Predicted rating of user u on item j,

$$\hat{r}_{uj} = \frac{\sum_{k \in \mathcal{N}_j^u} s_{kj} r_{uk}}{\sum_{k \in \mathcal{N}_j^u} s_{kj}} \tag{4}$$

Notes:

- the default value is $\bar{r}_u$ if $\mathcal{N}_j^u = \emptyset$
- $\mathcal{N}_j^u$ is dependent on both item j and user u

Bug!

- Step 1. Obtain the top-K nearest neighbors of item j, i.e., $\hat{\mathcal{N}}_j$ with $|\hat{\mathcal{N}}_j| = K$
- Step 2. Predict the rating of user u on item j,

$$\hat{r}_{uj} = \frac{\sum_{k \in \mathcal{I}_u \cap \hat{\mathcal{N}}_j} s_{kj} r_{uk}}{\sum_{k \in \mathcal{I}_u \cap \hat{\mathcal{N}}_j} s_{kj}}$$

Notes:

- We may have $|\mathcal{I}_u \cap \hat{\mathcal{N}}_j| < K$

Linear Combination of UCF and ICF

Predicted rating of user u on item j,

$$\hat{r}_{uj} = \lambda^{UCF}\hat{r}_{uj}^{UCF} + (1 - \lambda^{UCF})\hat{r}_{uj}^{ICF} \tag{5}$$

where $0 \leq \lambda^{UCF} \leq 1$ is a tradeoff parameter.

Data Set

- We use the files u1.base and u1.test of MovieLens100K[1] as our training data and test data, respectively.
- user number: $n = 943$; item number: $m = 1682$.
- u1.base (training data): 80000 rating records, and the density (or sparsity) is $80000/943/1682 = 5.04\%$.
- u1.test (test data): 20000 rating records.

[1] http://grouplens.org/datasets/

Evaluation Metrics

- Mean Absolute Error (MAE)

$$MAE = \sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} |r_{ui} - \hat{r}_{ui}| / |\mathcal{R}^{te}|$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} (r_{ui} - \hat{r}_{ui})^2 / |\mathcal{R}^{te}|}$$

- Performance: the smaller the better.

Implementation Details

- When we can not find K nearest neighbors, use as many neighbors as possible
- Post-processing for $\mathbb{G} = \{1, 2, 3, 4, 5\}$
  - if $\hat{r}_{ui} > 5$, set it as 5
  - if $\hat{r}_{ui} < 1$, set it as 1

Results

Table: Prediction performance of User-based CF, Item-base CF and Hybrid CF with $K = 50$, $\lambda^{UCF} = 0.5$ on MovieLens100K (u1.base, u1.test).

Method	RMSE	MAE
User-based CF	**0.9554**	**0.7480**
Item-based CF	0.9901	0.7801
Hybrid CF	0.9562	0.7538

Observation: Hybrid CF and user-based CF perform better than item-based CF on this data.

A1. Similarity Measurement

- When the number of co-rated items by user u and user w (i.e., $|\mathcal{I}_u \cap \mathcal{I}_w|$) is small, the similarity may be not reliable
- Similar ratings on popular items is less reliable than similar ratings on unpopular ones
- Item similarities are supposed to be more stable than user similarities
- There are many other similarity measurement and related techniques such as normalization

A2. Neighborhood Selection

- We may combine the strategies of *similarity threshold* and *top-K nearest neighbors*

A3. Prediction Rule

- Can we design a more sophisticated prediction rule?

In A Big Picture

- Collaborative filtering
  - Memory-based collaborative filtering
    - User-based collaborative filtering
    - Item-based collaborative filtering
  - Model-based collaborative filtering
- Content-based recommendation
- ...

Conclusion

- Basic assumptions and ideas of memory-based methods, including user-based CF and item-based CF
- Three basic questions of memory-based methods
  - Similarity measurement
  - Neighborhood selection
  - Prediction rule
- Hybrid CF

Homework

- Implement user-based CF, item-based CF and hybrid CF, and study their performance on u2.base, u2.test of MovieLens100K
  - Check the performance using different values of K and λ^{UCF}
- Reading: chapter 4 of *Recommender Systems Handbook*