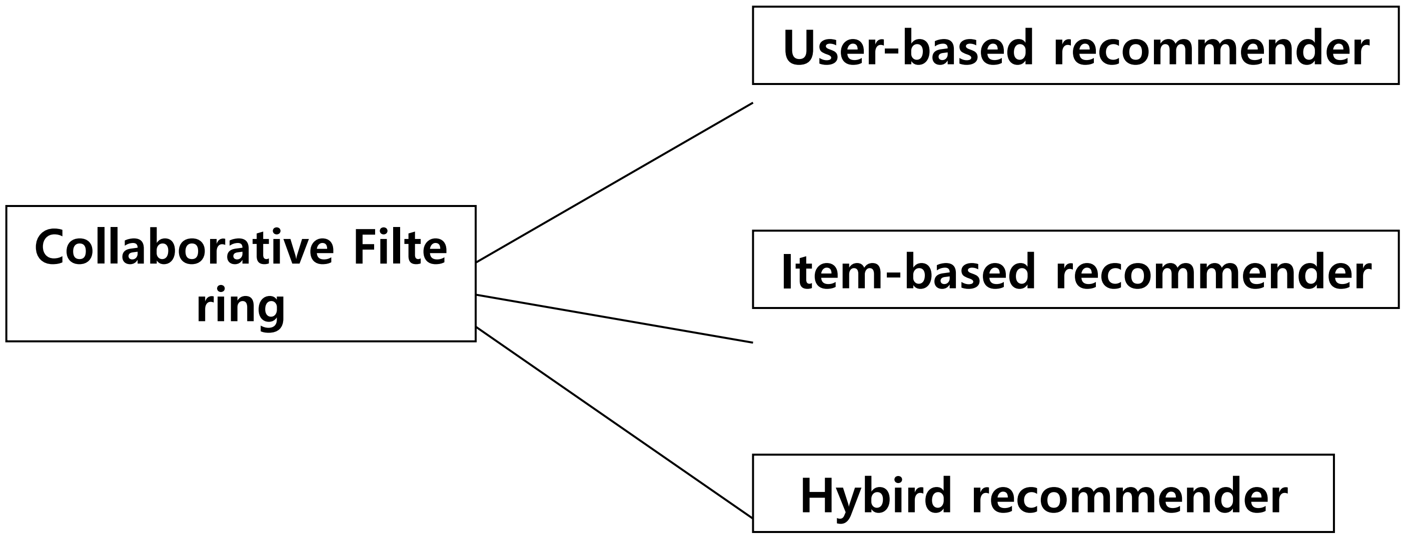
User-based collaborative filtering

Wang Jianfang
王建芳(in Chinese)

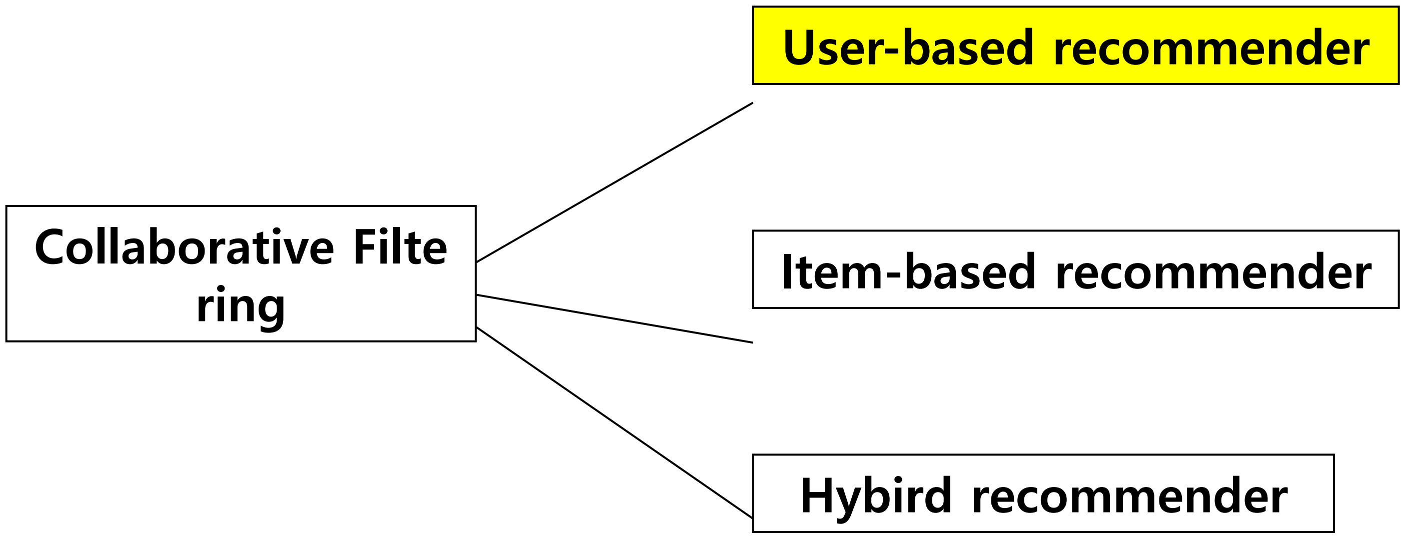
Agenda

- Algorithms (user-based)
- Code
- Installing environment
- MovieLens

Algorithms


User-based recommender
Collaborative Filte ring
Item-based recommender
Hybird recommender

Algorithms



Collaborative Filte ring
User-based recommender
Item-based recommender
Hybird recommender

Algorithms : **User-Based**

- **User-Based Nearest Neighbor**
  - Neighbor = similar users
  - Generate a prediction for an item *i* by analyzing ratings for *i* from users in *u*'s neighborhood

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} sim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} sim(u,n)}$$

Algorithms : **Item-Based**

- **Item-Based Nearest Neighbor**
  - Generate predictions based on similarities between items.
  - Prediction for a user u and item i is composed of a weight ed sum of the user u's ratings for items most similar to i.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} sim(i,j) \cdot r_{ui}}{\sum_{j \in ratedItems(u)} sim(i,j)}$$

User-based nearest-neighbor collaborative filtering (1)

- **The basic technique**
  - Given an "active user" (Alice) and an item i not yet seen by Alice
    - find a set of users (peers/nearest neighbors) who liked the same items as Alice in the past **and** who have rated item i
    - use, e.g. the average of their ratings to predict, if Alice will like item i
    - do this for all items Alice has not seen and recommend the best-rated
- **Basic assumption and idea**
  - If users had similar tastes in the past they will have similar tastes in the future
  - User preferences remain stable and consistent over time

User-based nearest-neighbor collaborative filtering (2)

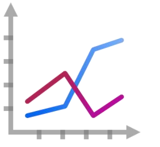
- Example
  - A database of ratings of the current user, Alice, and some other users is given:

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

  - Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

User-based nearest-neighbor collaborative filtering (3)

- Some first questions
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?



	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Measuring user similarity (1)

- **A popular similarity measure in user-based CF: Pearson correlation**

  a, b : users

  $r_{a,p}$: rating of user a for item p

  P : set of items, rated both by a and b

  – Possible similarity values between -1 and 1

$$\boldsymbol{sim(a,b)} = \frac{\sum_{\boldsymbol{p} \in \boldsymbol{P}} (\boldsymbol{r_{a,p}} - \boldsymbol{\bar{r}_a})(\boldsymbol{r_{b,p}} - \boldsymbol{\bar{r}_b})}{\sqrt{\sum_{\boldsymbol{p} \in \boldsymbol{P}} (\boldsymbol{r_{a,p}} - \boldsymbol{\bar{r}_a})^2} \sqrt{\sum_{\boldsymbol{p} \in \boldsymbol{P}} (\boldsymbol{r_{b,p}} - \boldsymbol{\bar{r}_b})^2}}$$

Measuring user similarity (2)

- **A popular similarity measure in user-based CF: Pearson correlation**

  a, b : users

  $r_{a,p}$: rating of user a for item p

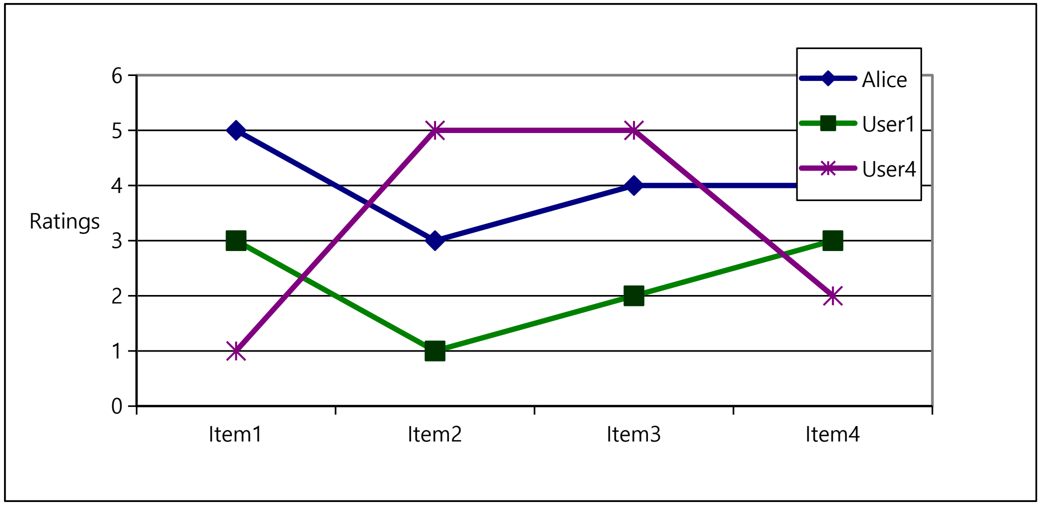
  P : set of items, rated both by a and b

  – Possible similarity values between -1 and 1

	Item1	Item2	Item3	Item4	Item5	
Alice	5	3	4	4	?	
User1	3	1	2	3	3	sim = 0,85
User2	4	3	4	3	5	sim = 0,00
User3	3	3	1	5	4	sim = 0,70
User4	1	5	5	2	1	sim = -0,79

Pearson correlation

- Takes differences in rating behavior into account



- Works well in usual domains, compared with alternative measures
  - such as cosine similarity

Making predictions

- A common prediction function:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b\in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b\in N} sim(a,b)}$$

- Calculate, whether the neighbors' ratings for the unseen item i are higher or lower than their average
- Combine the rating differences – use the similarity with a as a weight
- Add/subtract the neighbors' bias from the active user's average and use this as a prediction

similarity function

- Pearson

$$\boldsymbol{sim}(\boldsymbol{a},\boldsymbol{b}) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

- Cosine Similarity

$$w_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

- Jaccard

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}}$$

similarity function

- Pearson

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

- Pearson(simpler)

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}$$

Pratise

- How to get the similarity?

$$sim(a,b) = \frac{\sum_{p\in P}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p}-\bar{r}_b)^2}}$$

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0,85

sim = 0,00

sim = 0,70

sim = -0,79

similarity function

- Pearson

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

- Pearson correlation coefficient formula is used to calculate the correlation coefficient of XY to verif y the correctness of the code

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(X-\mu_Y)]}{\sigma_X \sigma_Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E[X^2] - E[X]^2}\sqrt{E[Y^2] - E[Y]^2}}$$

Homework

- Verifing the code in your computer.

  RS-1-2-User-based collaborative filtering.py

- Using Pearson similarity formulas to calculate the results in code 2.

Questions and Comments?

Thank you!!