

# Recommender System

Spring 2024

Hongchang Gao

# Review

- Step 1: Mean subtraction

$$\tilde{X} = X - \frac{1}{n}X\mathbf{1}\mathbf{1}^T$$

- Step 2: Compute the covariance matrix

$$A = \tilde{X}\tilde{X}^T$$

- Step 3: Eigen-decomposition

$$A = U\Sigma U^T$$

- Step 4: Keep the largest k eigenvectors

$$W = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k] \in \mathbb{R}^{d \times k}$$

# Introduction

- Recommender Systems
  - A particular type of personalized Web-based applications
  - Provide users personalized recommendations about content they may be interested
- Example:
  - Amazon: product recommendation
  - Netflix: movies recommendation
  - Google: news recommendation



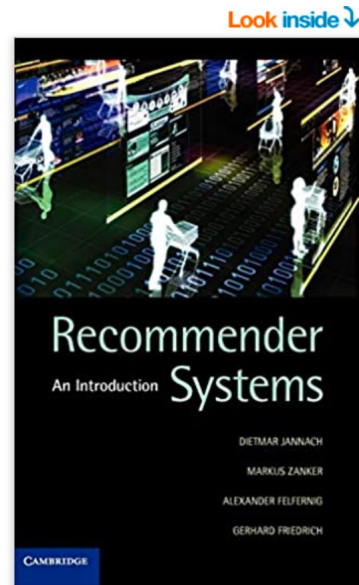
# Example

- Book Recommendation: a lot of sales from recommendations

## Recommender Systems (An Introduction) 1st Edition

by [Dietmar Jannach](#) (Author)

★★★★☆ 16 ratings



ISBN-13: 978-0521493369

ISBN-10: 0521493366

[Why is ISBN important?](#)

Have one to sell?

[Sell on Amazon](#)

Kindle   
\$21.73 - \$41.33

**Hardcover**  
\$62.33 - \$76.99

Paperback  
\$23.72

☐ Buy used:

☒ Buy new:

**Only 5 left in stock - order soon.**

Ships from and sold by Amazon.com.

May be available at a lower price from [other sellers](#), potentially

### More Buying Choices

11 new from \$76.91 | 16 used from \$40.99

### Products related to this item

Sponsored

Just released



**Transformers for Natural Language Processing:** Build innovative deep neural network ...  
Denis Rothman  
★★★★☆ 20  
Paperback  
**\$39.96** ✓prime



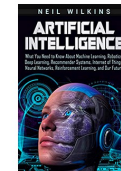
**Trustworthy Online Controlled Experiments** (A Practical Guide to A/B Testing)  
Ron Kohavi  
★★★★☆ 140  
Paperback  
**\$34.99** ✓prime



**Hands-On Recommendation Systems with Python:** Start building...  
Rounak Banik  
★★★★☆ 20  
Paperback  
**\$29.99** ✓prime



**The Economics of Data, Analytics, and Digital Transformation:** The theorems, laws, a...  
Bill Schmarzo  
★★★★☆ 24  
Paperback  
**\$29.99** ✓prime



**Artificial Intelligence:** What You Need to Know About Machine Learning, Robotics, De...  
Neil Wilkins  
★★★★☆ 13  
Paperback  
**\$13.34** ✓prime

Best seller

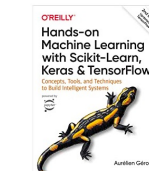


**System Design Interview - An insider's guide, Second Edition**  
Alex Xu  
★★★★☆ 495  
Paperback  
**\$24.99** ✓prime



**Building a Recommendation System with R**  
Suresh K. Gorakala  
★★★★☆ 14  
Paperback  
**\$29.99** ✓prime

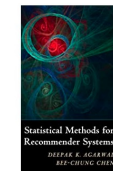
### Customers who viewed this item also viewed



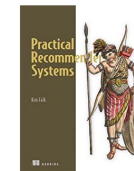
**Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow**  
> Aurélien Géron  
★★★★☆ 1,832  
Hardcover  
**#1 Best Seller** in  
Computer Vision & Pattern Recognition



**Recommender Systems: The Textbook**  
> Charu K. Aggarwal  
★★★★☆ 42  
Hardcover  
**\$59.97**  
Prime FREE Delivery  
In stock on April 26, 2021.



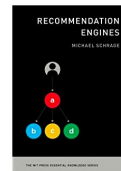
**Statistical Methods for Recommender Systems**  
Deepak K. Agarwal  
★★★★☆ 8  
Hardcover  
**\$53.99**  
✓prime FREE Delivery



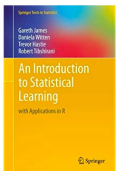
**Practical Recommender Systems**  
> Kim Falk  
★★★★☆ 20  
Hardcover  
**\$45.49**  
✓prime FREE Delivery



**Hands-On Recommendation Systems with Python:**...  
> Rounak Banik  
★★★★☆ 20  
Paperback  
**\$29.99**  
✓prime FREE Delivery



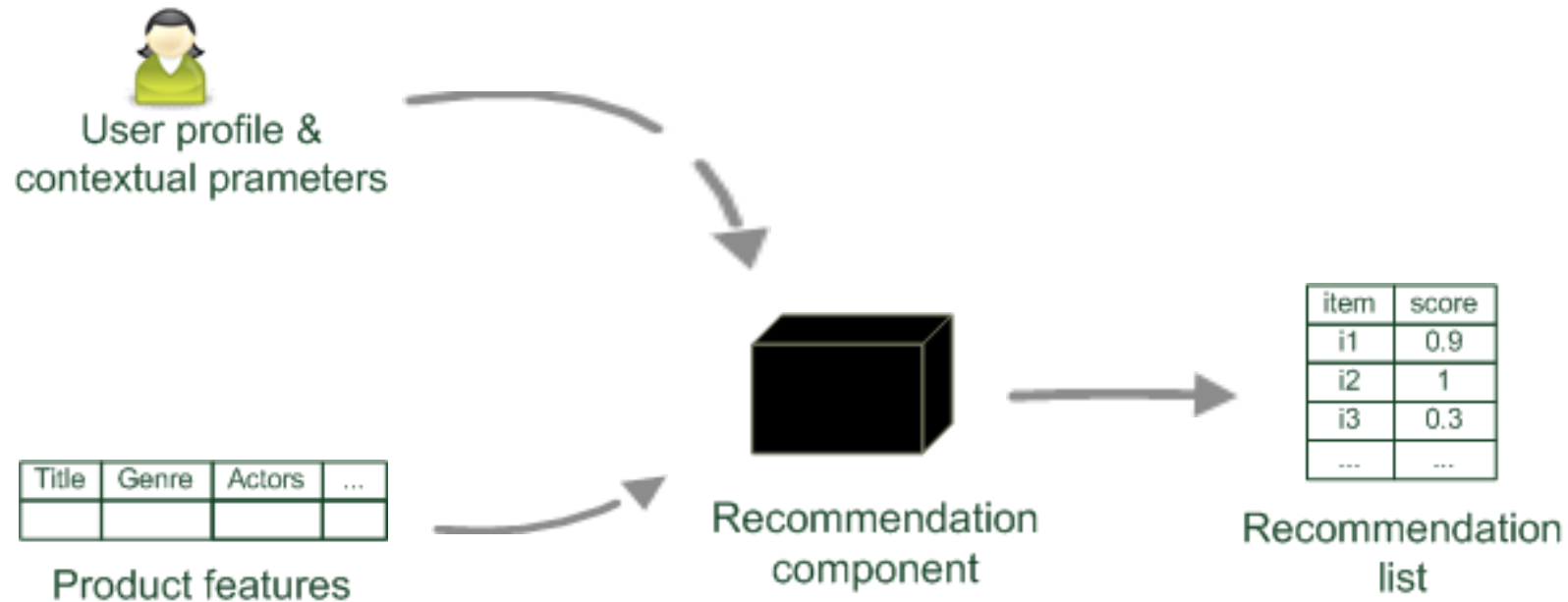
**Recommendation Engines (The MIT Press Essential Knowledge...)**  
> Michael Schrage  
★★★★☆ 40  
Paperback  
**\$14.97**  
✓prime FREE Delivery



**An Introduction to Statistical Learning: with Applications in R** (Springer Texts in...)  
> Gareth James  
★★★★☆ 1,086  
Hardcover  
**\$43.95**

# Recommender System

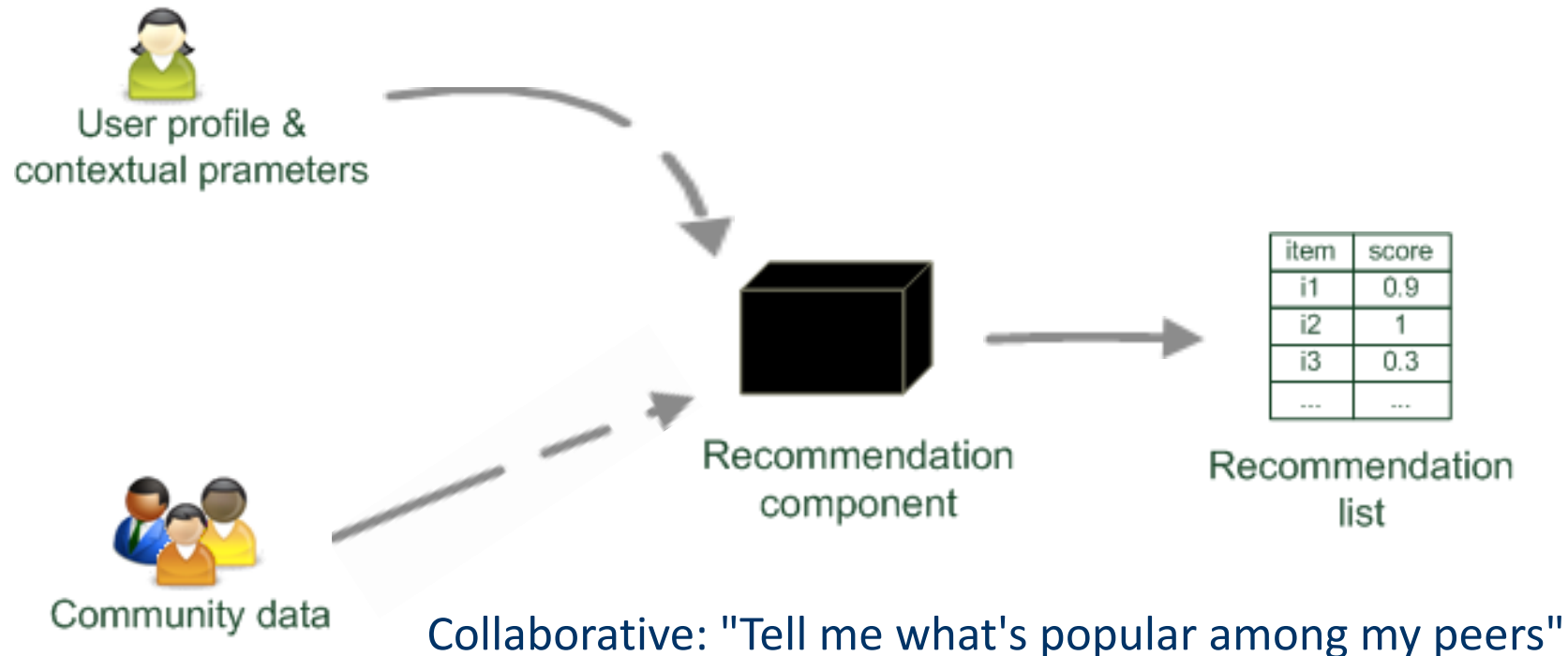
- 1. Content-based filtering
  - recommendations based on item descriptions/features



Content-based: "Show me more of the same what I've liked"

# Recommender System

- 2. Collaborative filtering
  - Look at the ratings of like-minded users to provide recommendations
  - Users who have expressed similar interests in the past will share common interests in the future.

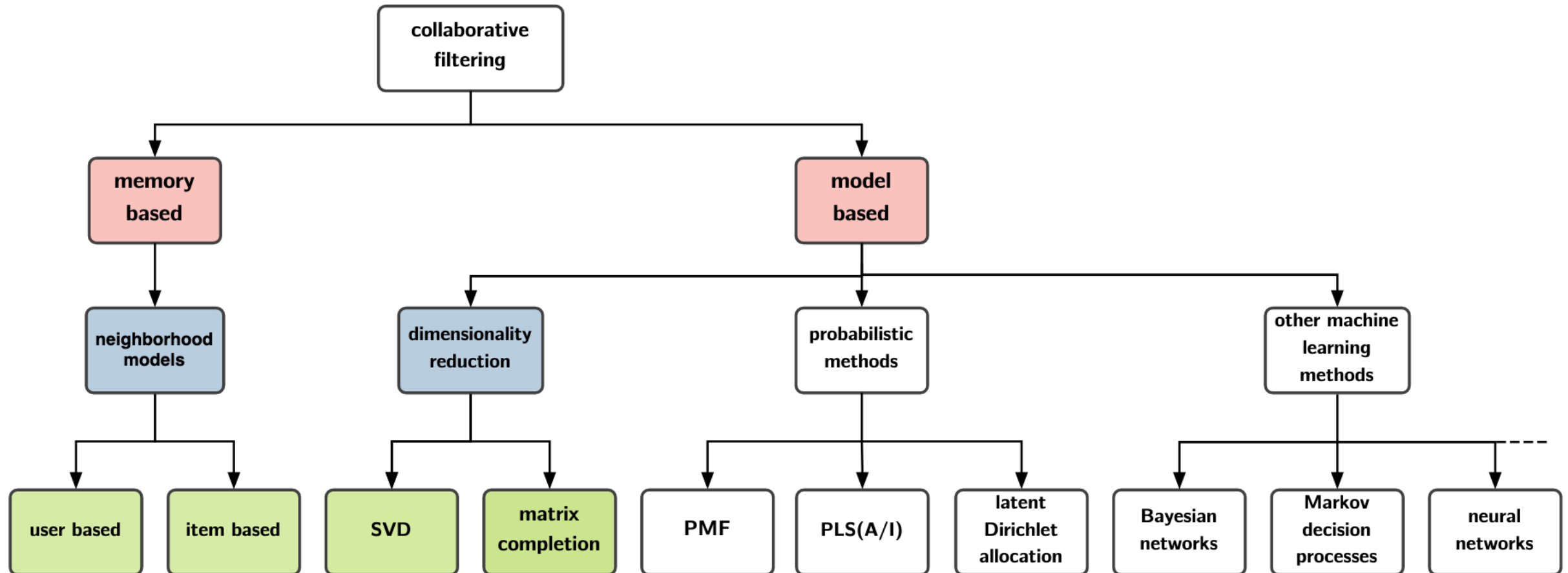


# Collaborative Filtering

- Given
  - Set of users
  - Set of items (movies, books, news, ...)
  - Feedback (ratings, ...)
- Predict the preference of each user for each item
  - Assumption: **similar feedback**  $\leftrightarrow$  **similar taste**

	Avatar	The Matrix	Up
Marco	?	4	2
Luca	3	2	?
Anna	5	?	3

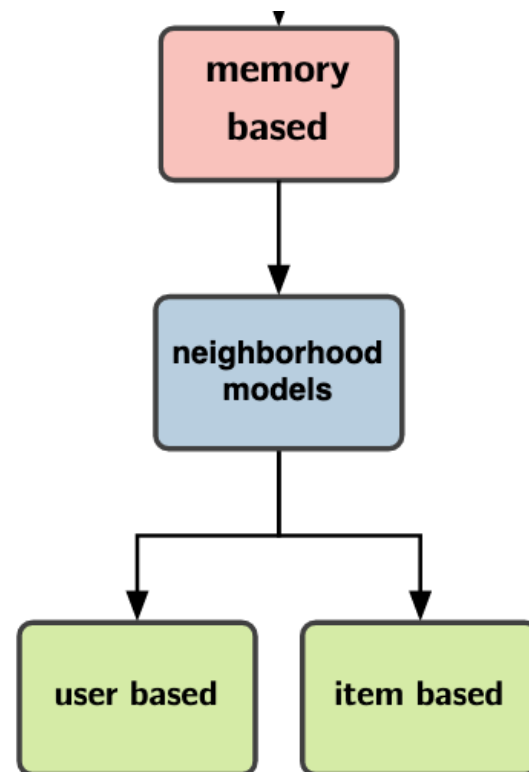
# Collaborative Filtering





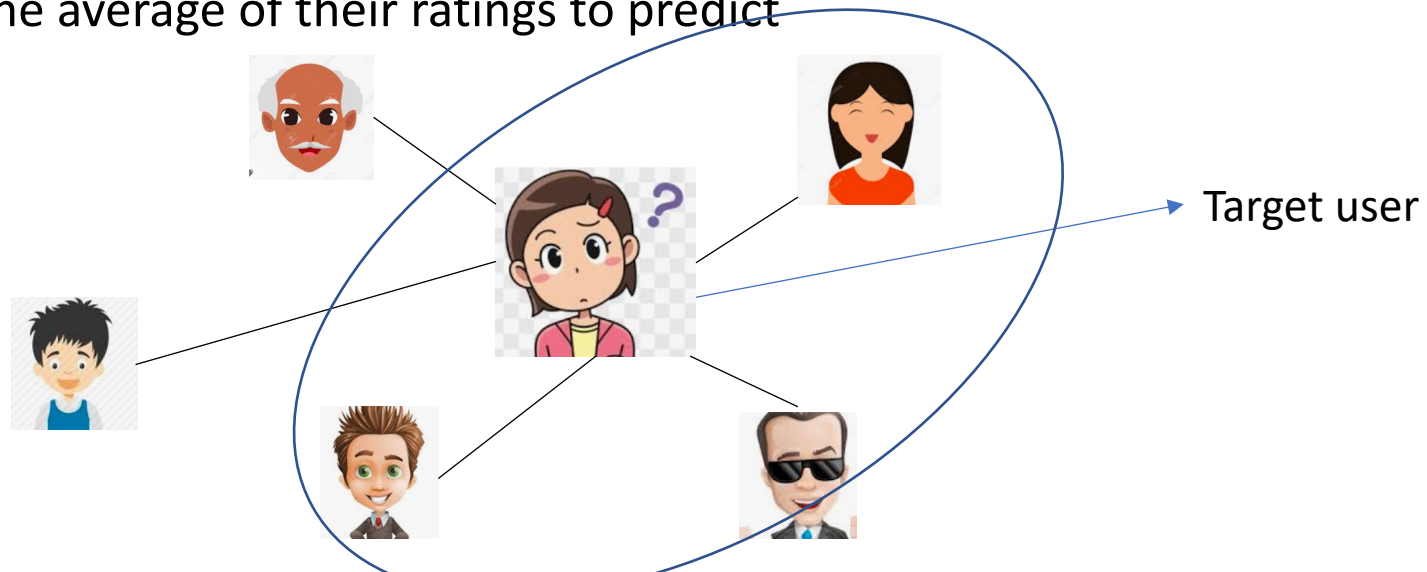
# Memory based CF

- Memory based collaborative filtering
  - User based CF
  - Item based CF



# Memory based CF: User-CF

- User-CF:
  - Idea
    - If users have similar tastes in the past, they will have similar tastes in the future
  - Recommend item  $i$  to Alice?
    - Find a set of users (peers/nearest neighbors) who liked the same items as the target user (Alice) in the past and who have rated item  $i$
    - Use the average of their ratings to predict



# Memory based CF: User-CF

- Illustration:
  - A database of ratings of the current user, Alice, and some other users is given
  - Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

	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

# Memory based CF: User-CF

- 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

# Memory based CF: User-CF

- How to measure the user similarity?
  - Pearson correlation
    - $a, b$  : users
    - $r_{a,p}$  : rating of user  $a$  for item  $p$
    - $P$  : set of items, rated by both  $a$  and  $b$
  - Possible similarity values between  $-1$  and  $1$

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

# Memory based CF: User-CF

- Pearson correlation

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

	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

# Memory based CF: User-CF

- How to make predictions?
  - Use similarity threshold or fixed number of neighbors

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a, b)}$$

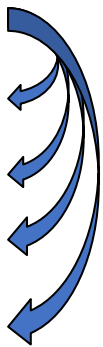
- Calculate, whether the neighbors' ratings for the unseen item are higher or lower than their average
- Combine the rating differences – use the similarity with  $a$  as a weight
- Add/subtract the neighbors' bias from the active user's average and use this as a prediction

# Memory based CF: User-CF

- Prediction

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

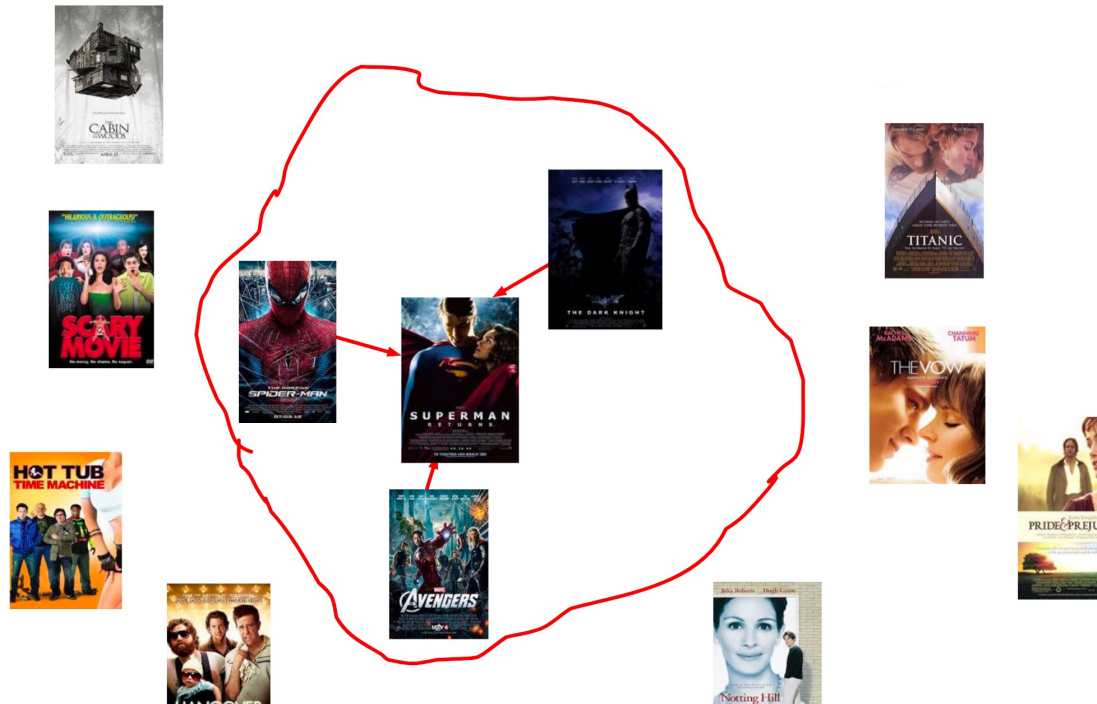
	Item1	Item2	Item3	Item4	Item5	
Alice	5	3	4	4	?	
User1	3	1	2	3	3	sim = 0,85
User2	4	3	4	3	5	sim = 0,00
User3	3	3	1	5	4	sim = 0,70
User4	1	5	5	2	1	sim = -0,79





# Memory based CF: Item-CF

- Idea:
  - Use the similarity between items (and not users) to make predictions



# Memory based CF: Item-CF

- Idea:
  - Use the similarity between items (and not users) to make predictions
- Steps:
  - Look for items that are similar to item5
  - Take Alice's ratings for these items to predict the rating for item5

	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

# Memory based CF: Item-CF

- How to measure the item similarity?
  - Cosine similarity

$$\text{sim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

- Ratings are seen as vector in n-dimensional space

	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

# Memory based CF: Item-CF

- How to make prediction?

$$\text{pred}(u, p) = \frac{\sum_{i \in \text{ratedItem}(u)} \text{sim}(i, p) * r_{u,i}}{\sum_{i \in \text{ratedItem}(u)} \text{sim}(i, p)}$$

- Neighborhood size is typically also limited to a specific size
- Not all neighbors are taken into account for the prediction

	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

# Evaluation

- 1. MAE and RMSE

- Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - r_i|$$

- Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2}$$

# Evaluation

Actually good	Recommended (predicted as good)
Item 237	Item 345
Item 899	Item 237
	Item 187

- 2. Precision and Recall

- Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved
  - E.g. the proportion of recommended movies that are actually good

$$\text{Precision} = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|all\ recommendations|}$$

- Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items
  - E.g. the proportion of all good movies recommended

$$\text{Recall} = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$