

INTRODUCTION TO DATA SCIENCE

This lecture is
based on course by E. Fox and C. Guestrin, Univ of Washington

19/12/2017

WFAiS UJ, Informatyka Stosowana
II stopień studiów

Recommender system

2

Personalization is transforming our experience of the world



100 Hours a Minute

What do I care about?

Information overload



Browsing is "history"

– Need new ways to discover content

Personalization: Connects *users & items*

viewers

videos

Recommender system

3

Movie recommendations



Connect users with movies
they may want to watch

Recommender system

4

Product recommendations

amazon.com [Help](#) | [Close window](#)

Recommended for You

High Performance Web Sites: Essential Knowledge for Front-End Engineers
by Steve Souders (Author)
Our Price: **\$19.79**
Used & new from \$16.24

[Add to Cart](#) [Add to Wish List](#)

Because you purchased...

Programming Collective Intelligence: Building Smart Web 2.0 Applications (Paperback)
by Toby Segaran (Author)

Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to [see all recommendations](#)

Even Faster Web Sites: Performance... (Paperback) by Steve Souders
★★★★☆ (7) \$23.10
[Fix this recommendation](#)

Simply JavaScript (Paperback) by Kevin Yank
★★★★☆ (149) \$26.37
[Fix this recommendation](#)

The Art & Science of Java (Paperback)
★★★★☆ (1)
[Fix this recommendation](#)

[Any Category](#) Algorithms Boxed Sets Business & Culture Java
Networking Networks, Protocols & APIs New SQL

Recommendations combine global & session interests

Recommender system

5

Music recommendations



The screenshot shows the Pandora web interface. At the top, there's a search bar for a 'New Station' and a playback control bar. Below that, there are tabs for 'Now Playing', 'Music Feed', and 'My Profile'. A sidebar on the left lists various radio stations, with 'Singer-Songwriter Essen...' highlighted. The main content area displays the album cover for 'Norwegian Wood (This Bird Has Flown)' by The Beatles from their album 'Rubber Soul'. Below the album cover, the lyrics are shown: 'I once had a girl / Or should I say she once had me / She showed me her room / Isn't it good Norwegian wood?'. There are also buttons for 'Show...' and 'Buy...'.

Recommendations form
coherent & diverse sequence

Recommender system

6

Friend recommendations



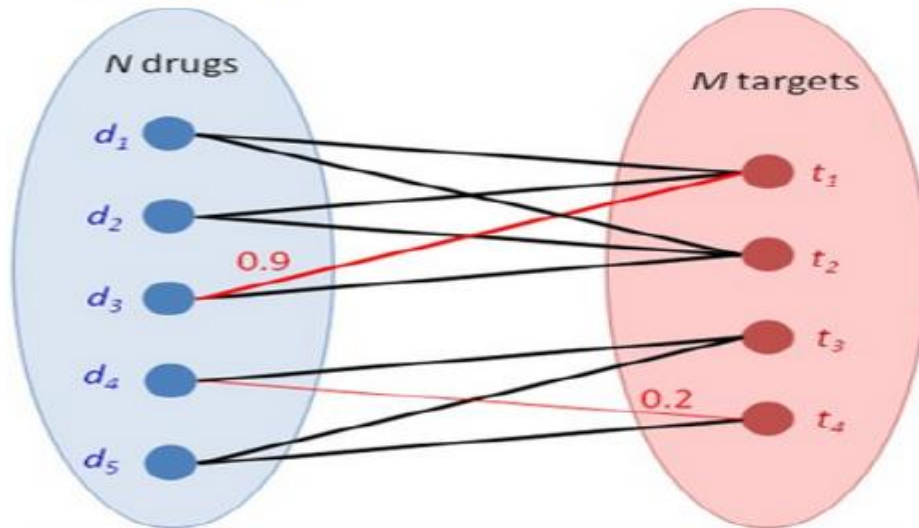
Users and "items"
are of the same "type"

Recommender system

7

Drug-target interactions

Cobanoglu et al. '13



What drug should we
"repurpose" for some disease?

Building a recommender system

Simplest approach: Popularity

9


- What are people viewing now?
 - Rank by global popularity
- **Limitation:**
 - No personalization

MOST POPULAR

E-MAILED BLOGGED SEARCHED

1. Really?: The Claim: Lack of Sleep Increases the Risk of Catching a Cold.
2. Magazine Preview: Coming Out in Middle School
3. Yes, We Speak Cupcake
4. Gossamer Silk, From Spiders Spun
5. Tie to Pets Has Germ Jumping to and Fro
6. Maureen Dowd: Where the Wild Thing Is
7. Maureen Dowd: Blue Is the New Black
8. The Holy Grail of the Unconscious
9. For Opening Night at the Metropolitan, a New Sound: Booing
10. Economic Scene: Medical Malpractice System Breeds More Waste

[Go to Complete List »](#)

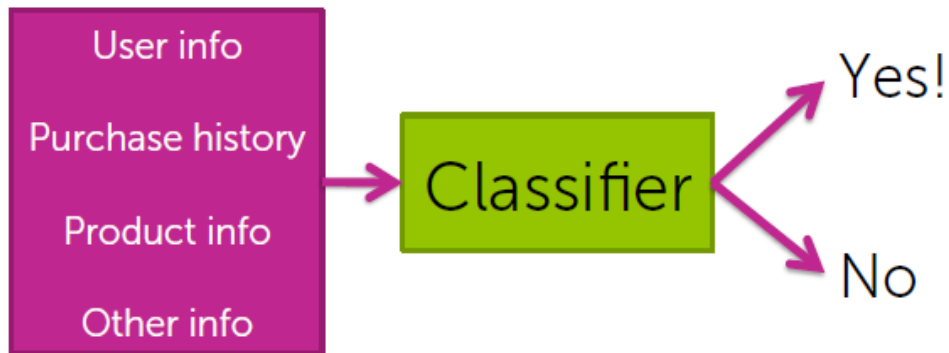
 **CUSTOMIZE HEADLINES**
Create a personalized list of headlines based on your interests. [Get Started »](#)



Solution 1: classification

10

What's the probability I'll buy this product?

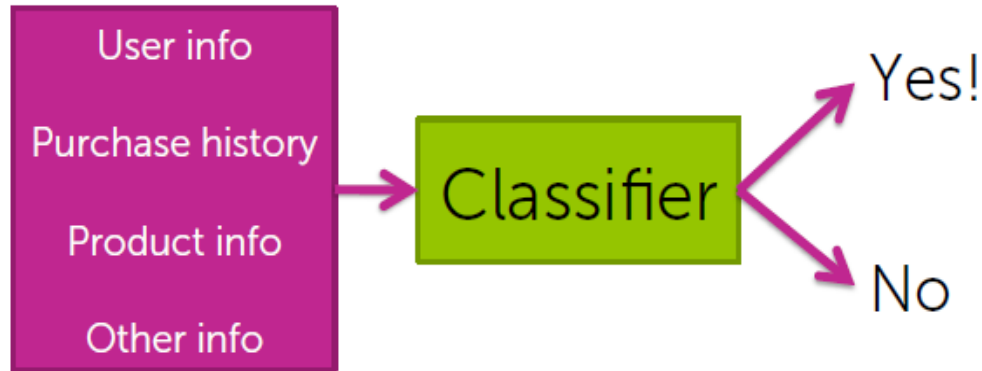


- **Pros:**
 - **Personalized:**
Considers user info & purchase history
 - **Features can capture context:**
Time of the day, what I just saw,...
 - **Even handles limited user history:** Age of user, ...

Solution 1: Classification

11

Limitations of classification approach



- Features may not be available
- Often doesn't perform as well as [collaborative filtering](#) methods (next)

Solution 2: People who bought this also bought...

12


Co-occurrence matrix

- People who bought *diapers* also bought *baby wipes*
- **Matrix C:**
store # users who bought both items *i* & *j*
 - (# items x # items) matrix
 - **Symmetric:** # purchasing *i* & *j* same as # for *j* & *i* ($C_{ij} = C_{ji}$)

Solution 2: People who bought this also bought...

13



Making recommendations using co-occurrences

- User  purchased *diapers*
 1. Look at *diapers* row of matrix
 2. Recommend other items with largest counts
 - *baby wipes, milk, baby food,...*

Solution 2: People who bought this also bought...

14

Co-occurrence matrix must be normalized

- What if there are very popular items?
 - Popular baby item:
Pampers Swaddlers diapers 
 - For any baby item (e.g., *i=Sophie giraffe* )
large count C_{ij} for *j=Pampers Swaddlers*
- Result:
 - Drowns out other effects
 - Recommend based on popularity

Solution 2: People who bought this also bought...

15

Normalize co-occurrences: Similarity matrix

- **Jaccard similarity**: normalizes by popularity
 - Who purchased *i and j* divided by who purchased *i or j*

- Many other similarity metrics possible, e.g., **cosine similarity**

Solution 2: People who bought this also bought...

16


Limitations

- Only current page matters, **no history**
 - Recommend similar items to the one you bought
- What if you purchased many items?
 - Want recommendations based on purchase history

Solution 2: People who bought this also bought...

17

(Weighted) Average of purchased items

- User  bought items $\{diapers, milk\}$
 - Compute user-specific score for each item j in inventory by combining similarities:

$$\text{Score}(\text{User}, baby\ wipes) = \frac{1}{2} (S_{baby\ wipes, diapers} + S_{baby\ wipes, milk})$$

- Could also weight recent purchases more
- Sort $\text{Score}(\text{User}, j)$ and find item j with highest similarity

Solution 2: People who bought this also bought...

18

Limitations



















- Does **not** utilize:
 - context (e.g., time of day)
 - user features (e.g., age)
 - product features (e.g., baby vs. electronics)
- Cold start problem
 - What if a new user or product arrives?

Solution 3: Discovering hidden structure by matrix factorization

19

Movie recommendation

- Users watch movies and rate them

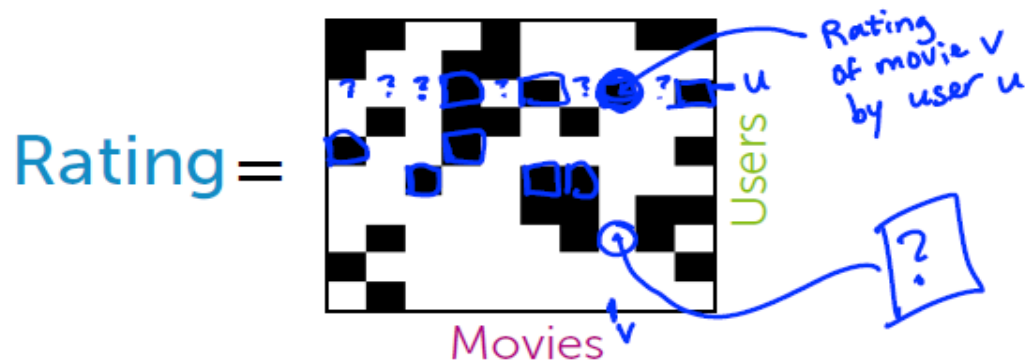
User	Movie	Rating
		★ ★ ★ ★ ☆
		★ ★ ★ ★ ★
		★ ★ ☆ ☆ ☆
		★ ★ ☆ ☆ ☆
		★ ★ ★ ★ ☆
		★ ☆ ☆ ☆ ☆
		★ ★ ★ ☆ ☆
		★ ★ ★ ★ ★
		★ ★ ★ ★ ☆

Each user only watches a few of the available movies

Solution 3: Discovering hidden structure by matrix factorization

20

Matrix completion problem



- **Data:** Users score some movies

$Rating(u,v)$ known for black cells
 $Rating(u,v)$ unknown for white cells


- **Goal:** Filling missing data?



Solution 3: Discovering hidden structure by matrix factorization

21

Suppose we had d topics for each user and movie

- Describe movie v  with topics R_v
 - How much is it **action**, **romance**, **drama**,...

$$R_v = [0.3 \quad 0.01 \quad 1.5 \quad \dots]$$

- Describe user u  with topics L_u
 - How much she likes **action**, **romance**, **drama**,...

Estimate

$$L_u = [2.5 \quad 0 \quad 0.8 \quad \dots]$$

- $Rating(u, v)$ is the product of the two vectors

$$R_v = [0.3 \quad 0.01 \quad 1.5 \quad \dots]$$

$$L_u = [2.5 \quad 0 \quad 0.8 \quad \dots]$$

$$L_{u'} = [0 \quad 3.5 \quad 0.01 \quad \dots]$$

$$\rightarrow 0.3 * 2.5 + 0 + 1.5 * 0.8 + \dots = 7.2 > 5$$

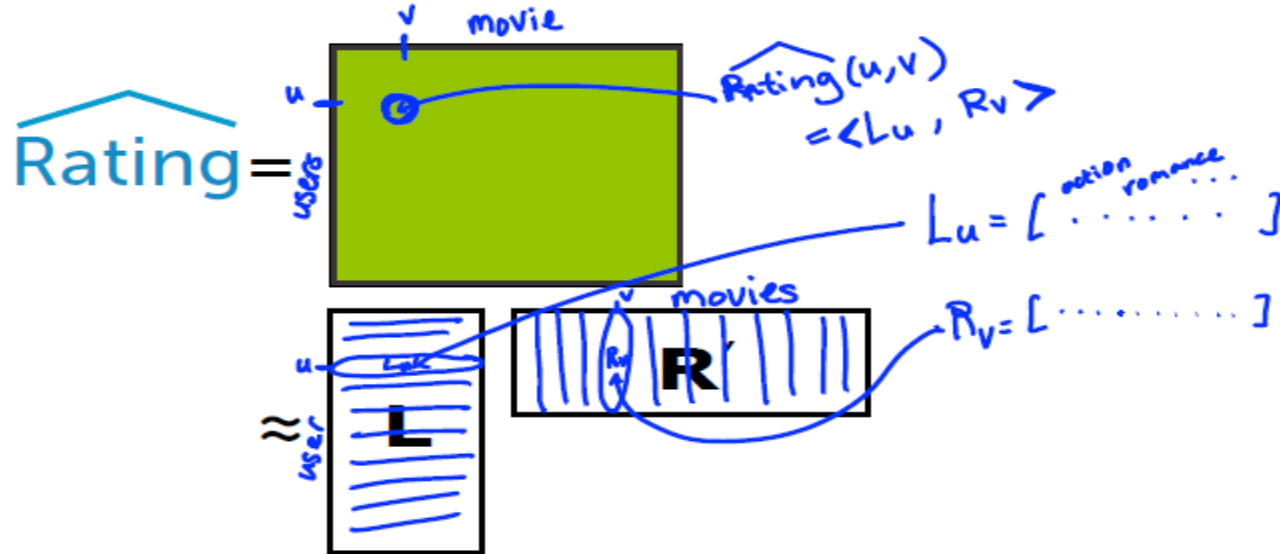
$$\rightarrow 0 + 0.01 * 3.5 + 1.5 * 0.01 + \dots = 0.8$$

- Recommendations:** sort movies user hasn't watched by $Rating(u, v)$

Solution 3: Discovering hidden structure by matrix factorization

22

Predictions in matrix form

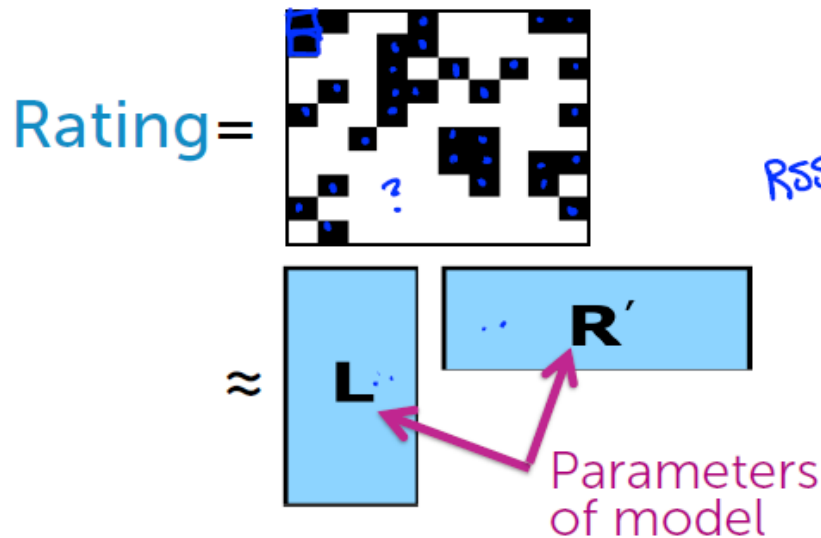


But we don't know topics of users and movies...

Solution 3: Discovering hidden structure by matrix factorization

23

Matrix factorization model: Discovering topics from data



$$RSS(L, R) =$$

$$\left(\text{Rating}(u, v) - \langle L_u, R_v \rangle \right)^2$$

↑ predicted rating

+ [include all (u, v) pairs where

Rating (u, v) are available]

Many efficient algorithms for factorization

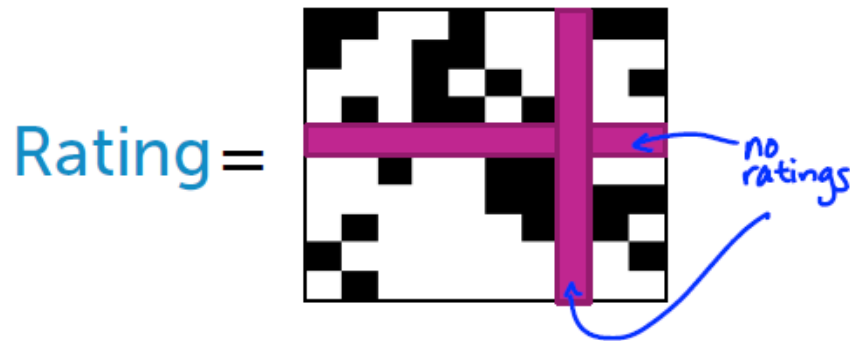
- Only use observed values to estimate "topic" vectors \hat{L}_u and \hat{R}_v
- Use estimated \hat{L}_u and \hat{R}_v for recommendations

Solution 3: Discovering hidden structure by matrix factorization

24

Limitations of matrix factorization

- Cold-start problem
 - This model still cannot handle a new user or movie



Featurized matrix factorization

25

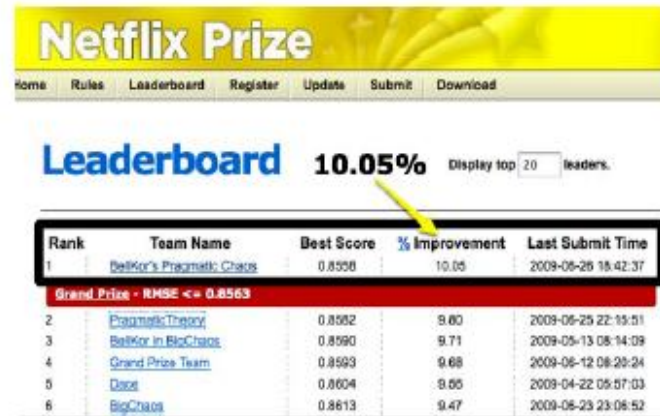
Combining features and discovered topics

- Features capture **context**
 - *Time of day, what I just saw, user info, past purchases,...*
- Discovered topics from matrix factorization capture **groups of users** who behave similarly
 - *Women from Seattle who teach and have a baby*
- **Combine** to mitigate cold-start problem
 - Ratings for a new user from **features** only
 - As more information about user is discovered, matrix factorization **topics** become more relevant

Blending models

26

- Squeezing last bit of accuracy by blending models
- Netflix Prize 2006-2009
 - 100M ratings
 - 17,770 movies
 - 480,189 users
 - Predict 3 million ratings to highest accuracy
 - **Winning team blended over 100 models**



The screenshot shows the Netflix Prize Leaderboard interface. At the top, there is a yellow banner with the text "Netflix Prize". Below the banner, there are navigation links: "Home", "Rules", "Leaderboard", "Register", "Update", "Submit", and "Download". The main heading is "Leaderboard" followed by "10.05%" and "Display top 20 leaders." A yellow arrow points to the "10.05%" value. Below this is a table with the following data:

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8508	10.05	2009-09-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	Pragmatic Theory	0.8562	9.80	2009-09-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-09-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-09-12 08:20:24
5	Dacor	0.8604	9.55	2009-04-22 09:57:03
6	BigChaos	0.8613	9.47	2009-08-23 23:08:52

Performance metrics

27

The world of all baby products



Performance metrics

28

User likes subset of items



Performance metrics

29

Why not use classification accuracy?

- Classification accuracy =
fraction of items correctly classified
(*liked* vs. *not liked*)
- Here, **not** interested in what a person
does not like
- Rather, how quickly can we discover the
relatively few *liked* items?
 - (Partially) an imbalanced class problem

Performance metrics

30

How many liked items were recommended?

A central purple stick figure points to a collection of baby products. The products are annotated with blue 'X' marks and pink circles. The pink circles highlight three items: a baby monitor, a car seat, and a stroller. The blue 'X' marks are placed over a crib, a hanging mobile, a pair of baby shoes, a box of baby wipes, a baby hat, a baby blanket, a baby bottle, and a baby toy. A blue box on the right contains the text 'Recall' and the formula $\frac{\# \text{ liked \& shown}}{\# \text{ liked}}$. Below the box is a hand-drawn blue box containing the fraction $\frac{3}{5}$.

Recall

$\frac{\# \text{ liked \& shown}}{\# \text{ liked}}$

$= \frac{3}{5}$

Performance metrics

31

Maximize recall: Recommend everything



Recall

$$\frac{\# \text{ liked \& shown}}{\# \text{ liked}}$$

$$= 1 \leftarrow \frac{10}{10} \checkmark$$

Performance metrics

32

Resulting precision?



Precision

$\frac{\# \text{ liked \& shown}}{\# \text{ shown}}$

*small,
maybe very
small*

Performance metrics

33

Optimal recommender

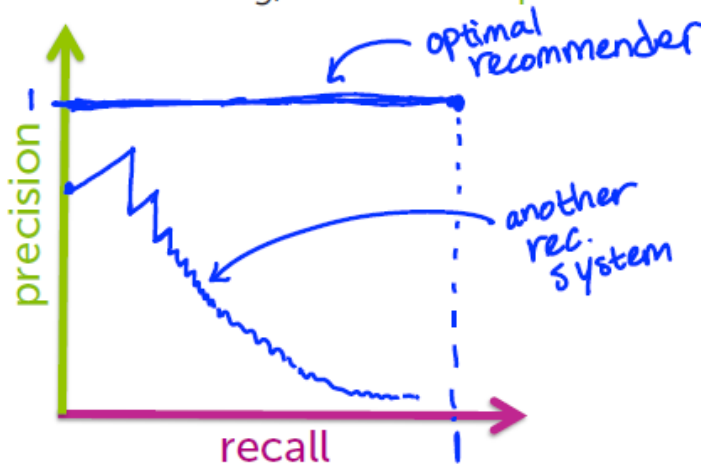


Performance metrics

34

Precision-recall curve

- **Input:** A specific recommender system
- **Output:** Algorithm-specific precision-recall curve
- To draw curve, vary threshold on # items recommended
 - For each setting, calculate the precision and recall

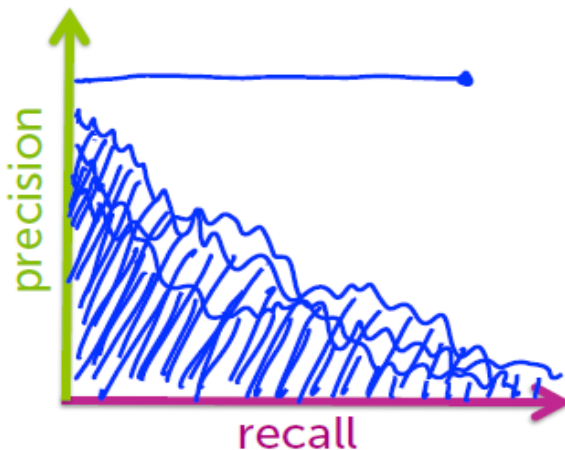


Performance metrics

35

Which Algorithm is Best?

- For a given **precision**, want **recall** as large as possible (or vice versa)
- One metric: largest **area under the curve (AUC)** ★
- Another: set desired recall and maximize precision (precision at k)



What you can do now ...

36

- Describe the goal of a recommender system
- Provide examples of applications where recommender systems are useful
- Implement a co-occurrence based recommender system
- Describe the input (observations, number of "topics") and output ("topic" vectors, predicted values) of a matrix factorization model
- Exploit estimated "topic" vectors (algorithms to come...) to make recommendations
- Describe the cold-start problem and ways to handle it (e.g., incorporating features)
- Analyze performance of various recommender systems in terms of precision and recall
- Use AUC or precision-at-k to select amongst candidate algorithms