Analytics for Customization

Customer Analytics

Tilburg University

Customer lifecycle

Customer **development**: change in behavior over time: buying more (up-selling) or different things (cross-selling)

Customer **acquisition**: how customers are "born" or first contact with the firm.

Customer **retention**: preventing customer "death" or churn.

Marketing is about acquiring, developing and retaining customers

Cross-selling

Getting customers to buy other products of the firm that customer has not already bought

Up-selling

Getting customers to more expensive variants or add-ons to products

Next product to buy (NPTB)

- Idea: use a model to predict which product the customer will buy next, and target cross-selling at that customer for that product.

- Use data on previous product ownership ($\mathrm{Own}_{ikt-1} = \{0,1\}$) for each individual i, time t, and product k as well as some demographics (Z_i), to predict next period ownership (Own_{ikt})
  - If there are K products, there are $K+1$ alternatives including no choice

- We can also estimate K separate binary logistic regressions. The probability of buying product j (vs. not buying j):

$$P(\mathrm{Own}_{ijt} = 1) = \frac{\exp\left(\beta_{0j} + \sum_{k=1}^{K} \beta_{kj}\,\mathrm{Own}_{ikt-1} + Z_i\gamma_j\right)}{1 + \exp\left(\beta_{0j} + \sum_{k=1}^{K} \beta_{kj}\,\mathrm{Own}_{ikt-1} + Z_i\gamma_j\right)}$$

Next-product-to-buy models for cross-selling applications

Odds ratios = $\exp(\beta_{kj})$

Current period

Table 21.2 Odds-ratios for next-product-to-buy (Adapted from Knott et al. 2002)

Last period

	Product j				
Product k	Base checking	No-fee checking	Base savings	No-fee savings	CDs
Base checking	2.16[a]	0.66[b]	2.29	0.73	0.36
No-fee checking	0.68	2.66	1.55	1.48	0.69
Base savings	1.67	1.09	0.83	0.96	1.36
No-fee savings	1.47	0.12	0.30	2.54	1.66
CDs	0.63	0.45	0.44	0.51	4.94

[a] To be read: Owning base checking increases the odds of next purchasing another base checking account by 116%.
[b] To be read: Owning base checking decreases the odds of next purchasing no-fee checking by 34% (1–0.66).

For most products, owning it in t-1 strongly increases the probability that customers buy it again in t.

BKN p. 522

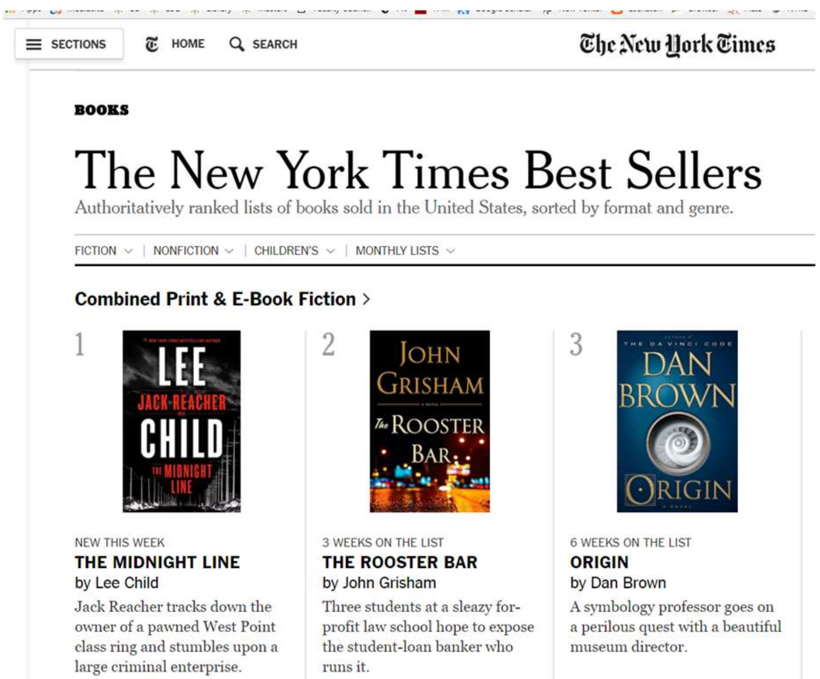
Many products

NPTB models break down when the number of products gets large

Market	New products introduced in 2014	Total products available in 2014
Books	300,000	29,000,000
Music albums	75,000	4,000,000
Movies	700	325,000
PC video games	2,700	6,000
iOS apps	400,000	2,000,000
Android apps	500,000	2,400,000
Restaurants in Manhattan	1,200	10,000

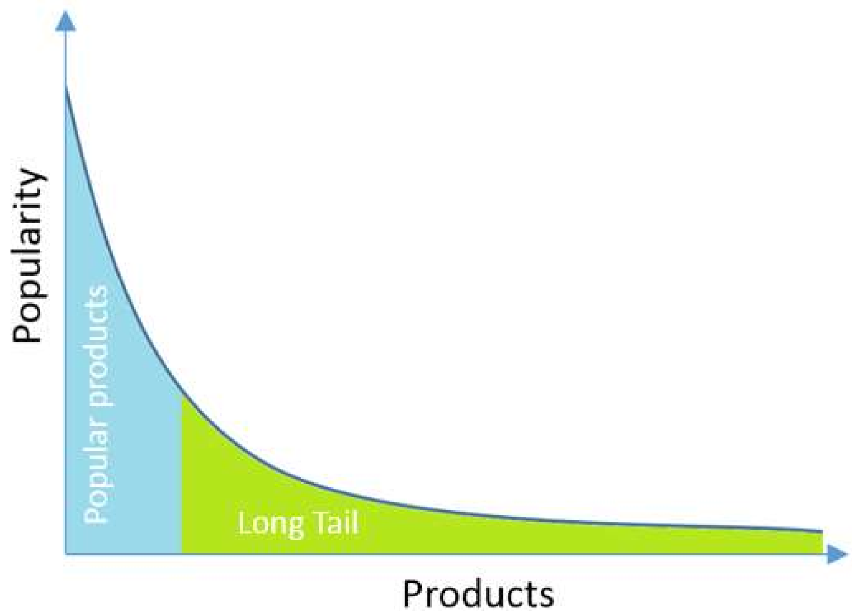
Traditional product discovery

What's popular



Reviews

Traditional discovery leaves most products left out



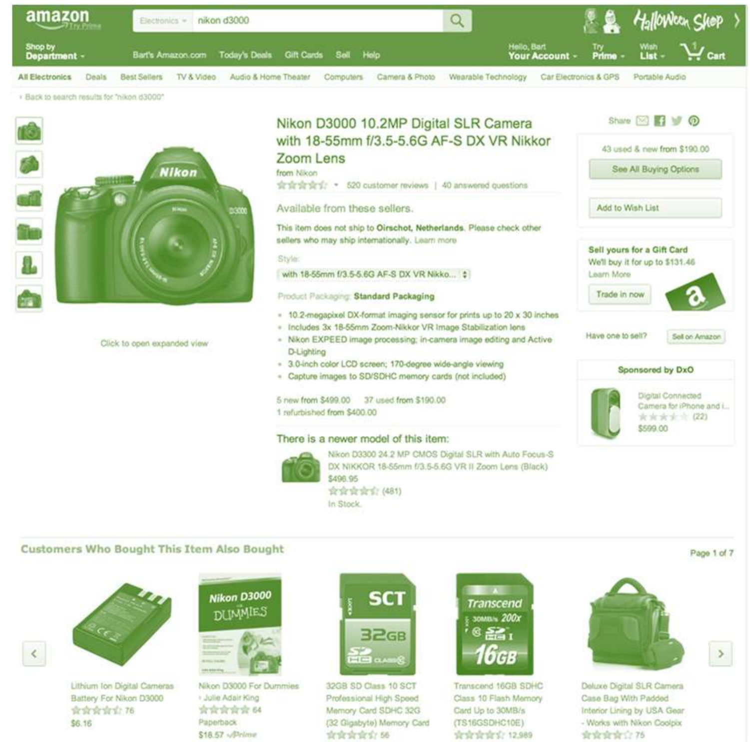
Can automated product discovery – recommender systems – do better?

Recommender Systems

Inbound customization

Use big data on <u>consumer views</u>, <u>purchases</u> and <u>reviews</u>

Use methods to make <u>customized dynamic</u> recommendations



Key issues

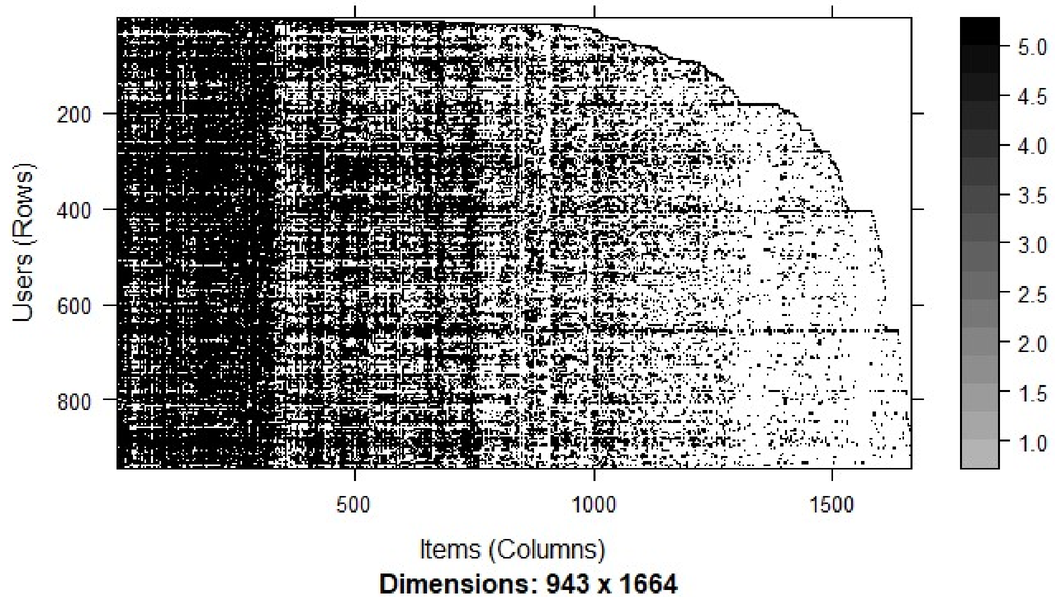
1. What type of data do you use to build the RS?

2. How do you make predictions?

3. How do you measure success or performance of the RS?

Most important data challenge: sparsity

The data was collected through the MovieLens web site (movielens.umn.edu) during Sept 1997 - Apr 1998. The data set contains ~100k ratings (1-5) from 943 users on 1664 movies. Each user has rated at least 19 movies.



6% of entries non-missing

Matrix is sparse: most users haven't rated most movies

1. Collecting data: ratings

	Jungle Book	Civil War	Deadpool	Zootopia
Adam	5	???	1	???
Ben	???	4	???	3
Chris	2	???	4	???
David	???	???	???	2

"Cold start" problem

- **New items** have **no ratings/viewings**
- **New users** have **no history**

1. Collecting data

Explicit

	Jungle Book	Civil War	Deadpool	Zootopia
Adam	5		1	
Ben		4		3
Chris	2		4	
David				2

Simple: ask people to rate items

Not scalable: most users don't leave ratings

Implicit
(base on actions like consuming or viewing)

	Jungle Book	Civil War	Deadpool	Zootopia
Adam	1	0	1	0
Ben	0	1	0	1
Chris	1	0	1	0
David	0	0	0	1

Scalable: learn ratings from user actions

- Purchase implies high rating

Hard to learn low ratings

- Does non-purchase mean not aware or aware but does not like?

Prediction: a few approaches

1. **Content filtering**: making recommendations based on item attributes and user preferences for attributes.

   Recommend action movies to teenage males

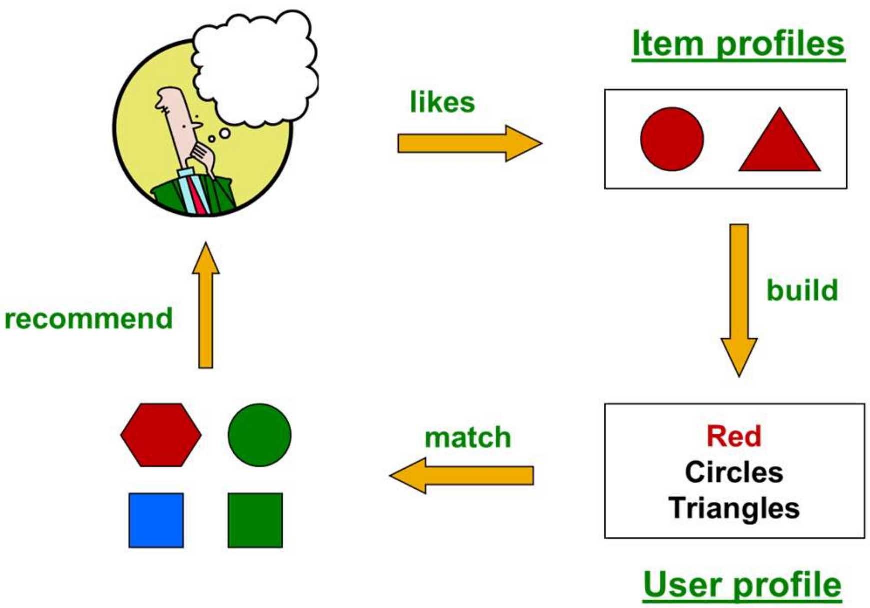
2. **Collaborative filtering**: based on similarities between users and products.

   Recommend X if liked Y, because others who liked Y liked X

3. **Matrix factorization:** use low dimensional factors to approximate sparse matrix

Content-based recommendations

Recommend items to customer similar to previous items rated highly by customer



J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

[Recommendation Systems]

Building item profiles

Profile is a set of features

Movies: author, title, actor,
Images, **videos:** metadata, tags
People: set of friends

Can also use tags, scripts or summaries, e.g., "surprise"

	Item profile #1	Item profile #2
	Pretty Woman	**Total Recall**
AS	0	1
JR	1	0
"surprise"	0.1	0.4

User Profiles

movies

attributes

$V(i,j)$

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator
AS	0	1	0	1	1
JR	1	0	1	0	0
"surprise"	0.1	0.4	0.1	0	0.1

Adam's ratings	3	1	5	2	4

r	Adam's normalized ratings	0	-2	2	-1	1

User profile

Find average change in ratings for characteristic, e.g., on average Adam rates movies with JR 1 point higher.

Of the rated movies: take the inner product of characteristic matrix and normalized reviews and divide by total characteristics

sum across movies

$$u_i = \frac{\sum_j v_{ij}\ r_j}{\sum_j v_{ij}}$$

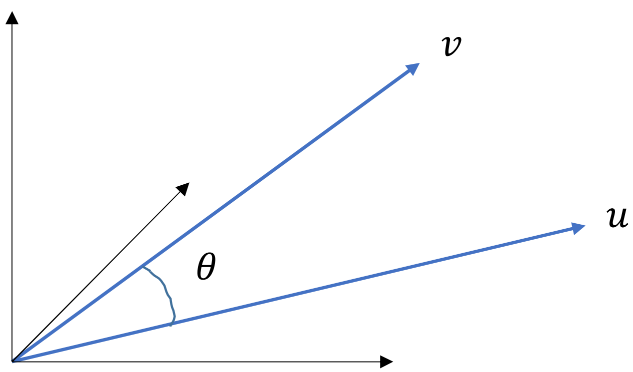
for characteristic i across movies j.

	Adam
AS	-0.67
JR	1
"surprise"	-0.71

Vector space model

- We now have an item profile and a user profile in terms of I characteristics

- The similarity between them is measured as the angle between the vectors



$$CS_{uv} = \cos(\theta) = \frac{u \cdot v}{\|u\| \|v\|} = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2} \sqrt{\sum_i v_i^2}}$$

If $\cos(\theta) = 1$, angle $\theta = 0$

Example: Making predictions

	True Lies	Notting Hill	Adam
AS	1	0	-0.67
JR	0	1	1
“surprise”	0.1	0	-0.71

$$CS_{Adam,TL} = \frac{(1)(-0.67) + (0)(1) + (0.1)(-0.71)}{\sqrt{(1)^2 + (0)^2 + (0.1)^2}\sqrt{(-0.67)^2 + (1)^2 + (-0.71)^2}} = -0.53$$

$$CS_{Adam,NH} = \frac{(0)(-0.67) + (1)(1) + (0)(-0.71)}{\sqrt{(0)^2 + (1)^2 + (0)^2}\sqrt{(-0.67)^2 + (1)^2 + (-0.71)^2}} = 0.72$$

Content-based approach

Positives

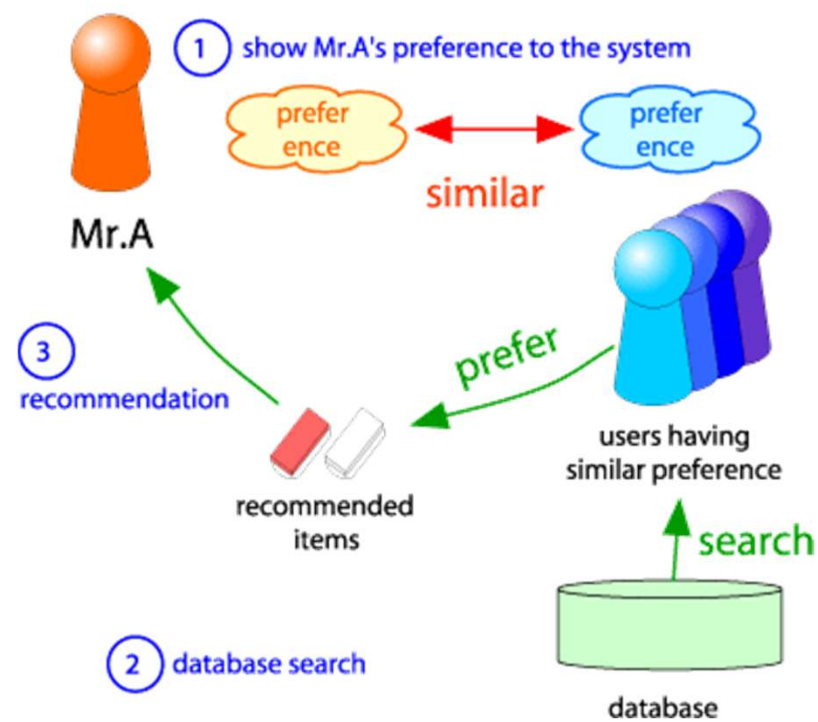
- No need for data on other users
  - Good for unique tastes
- Able to recommend new & unpopular items
  - No cold-start for items problem
- Can explain why something was recommended (JR +)

Negatives

- Finding features is hard
  - Hard to classify experience goods based on artists, genres, etc.
- Overspecialization
  - Never recommends items outside content profile
- Cold-start problem for users
  - How do you build a profile for a new user?

Collaborative Filtering

- Consider Mr. A
- Find other users whose ratings are similar to Mr. A's
- Estimate Mr. A's ratings based on ratings of similar users.



(This is called "user-based" collaborative filtering; also item-based)

User-based collaborative filtering

We're going to guess correlations

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator	Notting Hill
Adam	2	5	4	2	?	?
Ben	5	1	2		1	
Cindy	5	5	5	5	5	5
Dave	2	5		3		
Emily	5	3	5	3		5
Fred	1	5				1
George	2		5		5	

Adam's ratings are positively correlated to those of Fred and George

User-based collaborative filtering

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator	Notting Hill
Adam	2	5	4	2	?	?
Ben	5	1	2		1	
Cindy	5	5	5	5	5	5
Dave	2	5		3		
Emily	5	3	5	3		5
Fred	1	5				1
George	2		5		5	

Adam's ratings are negatively correlated to those of Ben

User-based collaborative filtering

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator	Notting Hill
Adam	2	5	4	2	?	?
Ben	5	1	2		1	
Cindy	5	5	5	5	5	5
Dave	2	5		3		
Emily	5	3	5	3		5
Fred	1	5				1
George	2		5		5	

Adam's rating for Predator is positively related to George and negatively to Ben.

Adam's rating for Notting Hill is positively related to Fred.

Item-based collaborative filtering

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator	Notting Hill
Adam	2	5	4	2	?	?
Ben	5	1	2		1	
Cindy	5	5	5	5	5	5
Dave	2	5		3		
Emily	5	3	5	3		5
Fred	1	5				1
George	2		5		5	

Predator’s ratings are positively correlated to those of Total Recall and Erin Brockovich

Item-based collaborative filtering

	Pretty Woman	Total Recall	Erin Brockovich	Terminator 2	Predator	Notting Hill
Adam	2	5	4	2	?	?
Ben	5	1	2		1	
Cindy	5	5	5	5	5	5
Dave	2	5		3		
Emily	5	3	5	3		5
Fred	1	5				1
George	2		5		5	

Notting Hill's ratings are positively correlated to those of Pretty Woman

Item vs. User based CF

- In theory, item- and user-based CF are dual approaches
- In practice item-based predicts better than user-based
- Why? Items are "simpler" than users
  - Items belong to a small set of "genres", users have varied tastes
    - Adam can like baroque organ music and acid rock
    - A song is unlikely to be to be categorized as both
  - Item similarity is more meaningful than user similarity

Collaborative Filtering

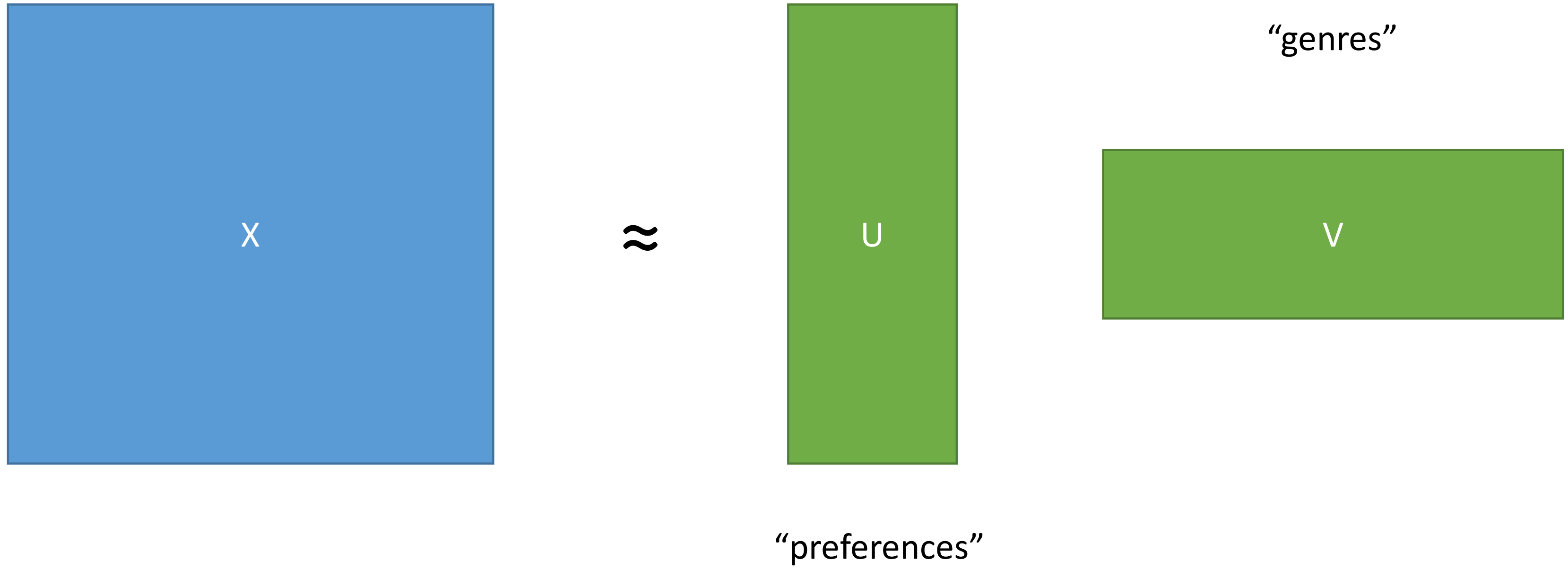
Positives

- Works for any kind of item
  - No feature selection needed

Negatives

- Cold start
  - Need enough users in the system to find a match
  - Cannot recommend unrated item
  - New items, esoteric items
- Sparsity
  - User matrix is sparse
  - Hard to find users that have rated the same set of items
- Popularity bias
  - Tends to recommend popular items
  - "Harry Potter effect"

Matrix factorization



1000 x 1600 matrix

= 1600000

1000 x d matrix

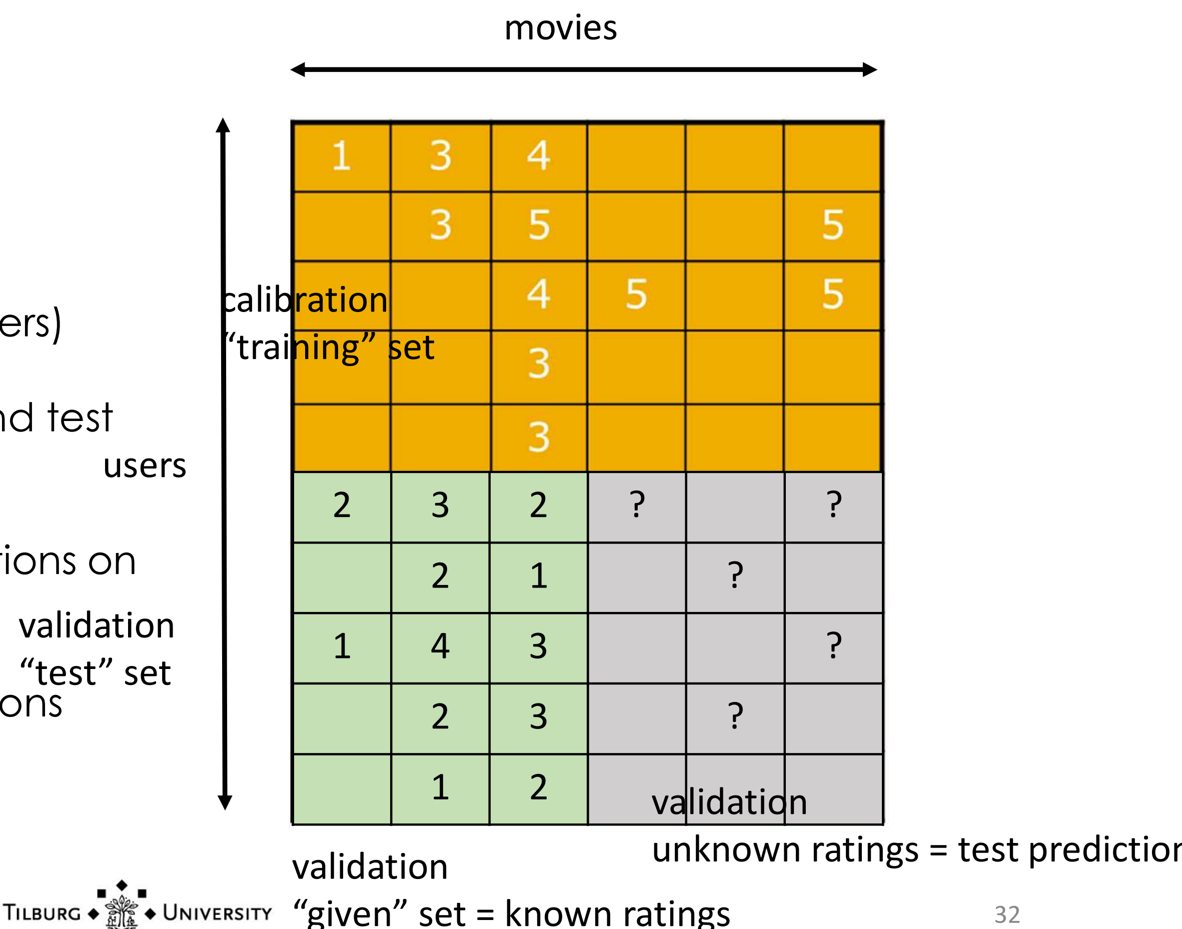
d x 1600 matrix

= 5000 + 8000 if d = 5

3. How do you evaluate RS?

How well does model predict out-of-sample, in the "test" or validation set.

- Calibration/training: fit model (80% of users)
- Validation/training: make predictions and test (20%)
  - Known/given: used to make predictions on other part
  - Unknown: used to evaluate predictions



Challenges with recommender systems

- Why recommend a product the customer was going to buy anyhow?
  - Consider the counterfactual: what would have happened had the user not seen the recommendation?
- Not enough diversity in recommendations
  - If I like one Harry Potter movie, my recommendations is all Harry Potter movies
- I no longer need the product recommended after I buy in the category (preferences change; see Amazon recommendations)

Wrapping up

- Using analytics to customize: which product to which customer? (when and over what channel)
  - Cross-selling, up-selling and recommender systems
- Important drivers: product ownership, similarity to others, attributes of products