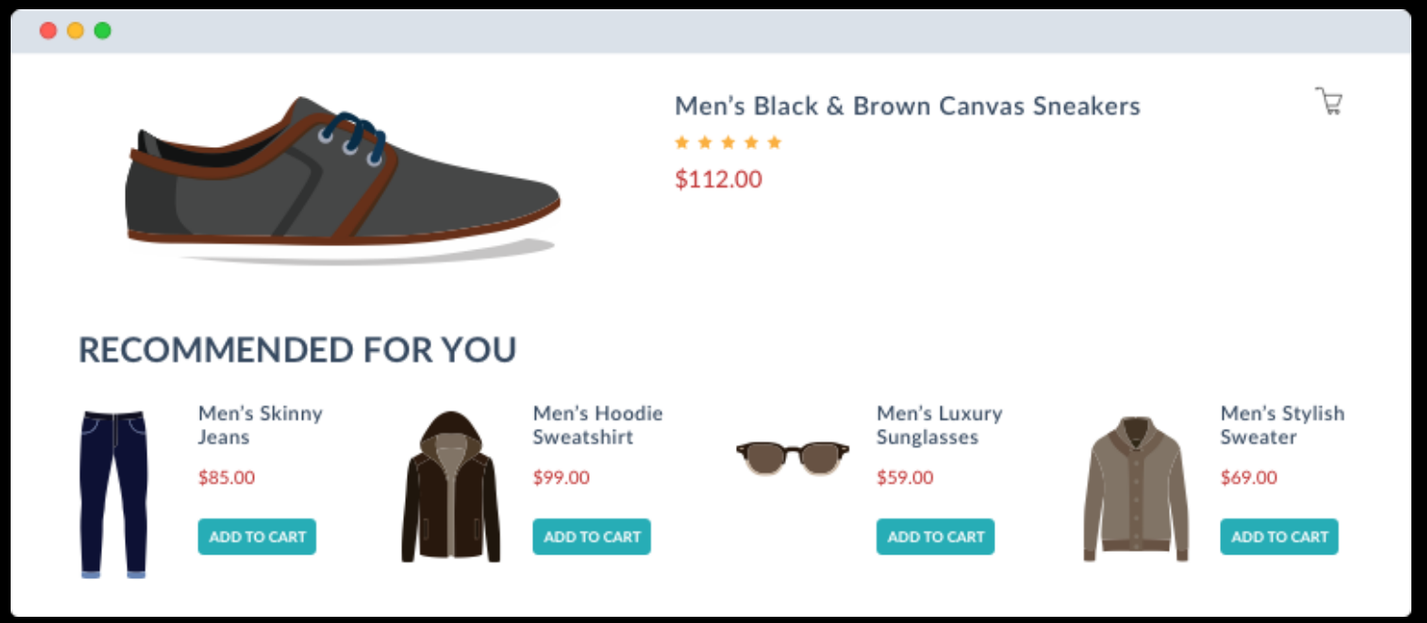
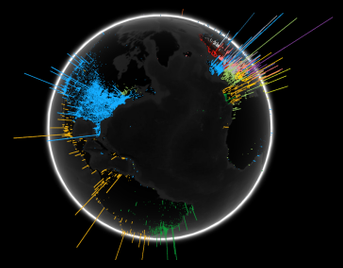
Recommender Systems



Hari Sundaram
Associate Professor (CS, ADV)
hs1@illinois.edu

thanks: Andrei Broder, Vanja Josifovski

Introduction

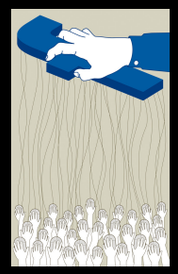


Web search

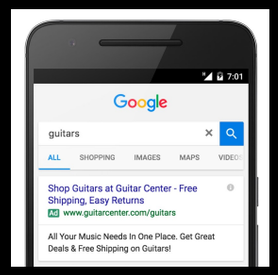
Game Theory

Auctions

Data flows



Privacy



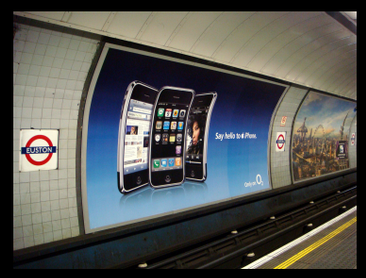
Text Ads

Display Ads

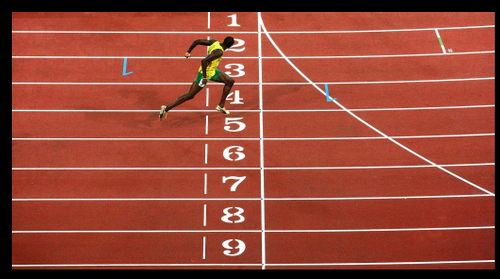
NETFLIX

Recommender systems

Behavioral targeting



Emerging areas



Final Presentations

thus far, we've examined the similarity between ads based on search logs

we've identified similar ads based on random walks on a click graph

but, in general, we can ask "**what other ads may be of interest?**"

might a person interested in buying a sports car, and having watched a BMW ad, be interested in an Audi ad?

A set of techniques to recommend items based on explicit (rating) or implicit (page visits, ad clicks)

It collects the user responses and assumes the items are not accessible

what are recommender systems?

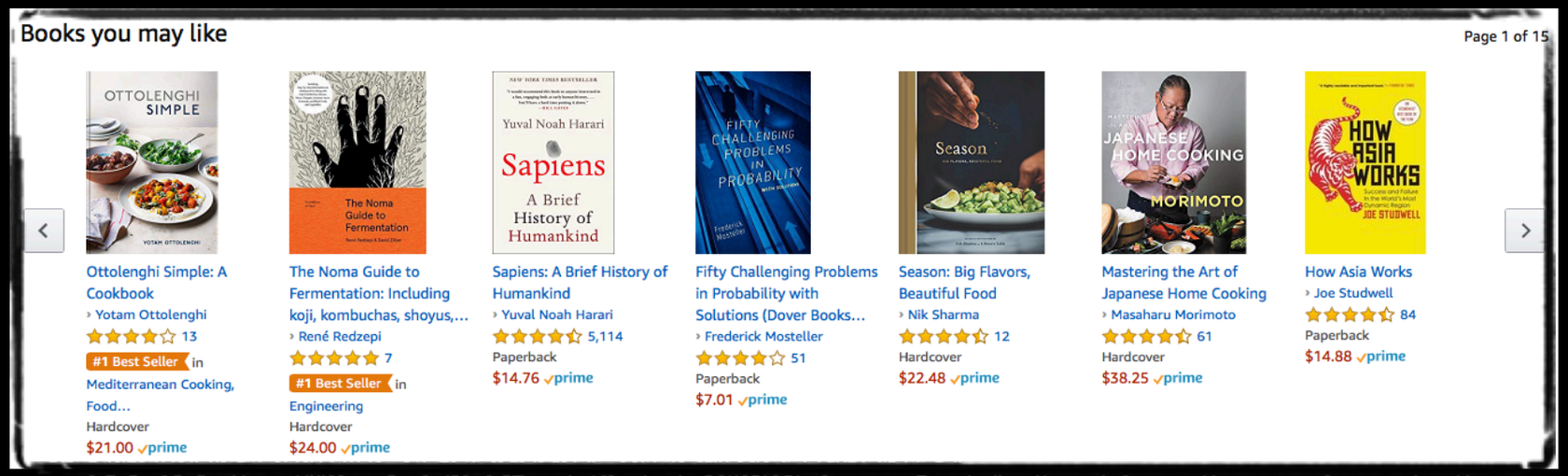
Usually does not take in account the content of the item

- Opposed to matching of ads we have discussed so far
- Recent techniques combine the two

While our discussion is on movie ratings, we will discuss the similarities and differences with the ad domain

Recommend items based on past transactions of many users
Analyze relations between users and items
Specific data characteristics are irrelevant
Domain-free:
    user/item attributes are not necessary
    Can identify elusive aspects

collaborative filtering



movie ratings

Training data

user	movie	date	score
1	21	5/7/02	1
1	213	8/2/04	5
2	345	3/6/01	4
2	123	5/1/05	4
2	768	7/15/02	3
3	76	1/22/01	5
4	45	8/3/00	4
5	568	9/10/05	1
5	342	3/5/03	2
5	234	12/28/00	2
6	76	8/11/02	5
6	56	6/15/03	4

Test data

user	movie	date	score
1	62	1/6/05	?
1	96	9/13/04	?
2	7	8/18/05	?
2	3	11/22/05	?
3	47	6/13/02	?
3	15	8/12/01	?
4	41	9/1/00	?
4	28	8/27/05	?
5	93	4/4/05	?
5	74	7/16/03	?
6	69	2/14/04	?
6	83	10/3/03	?

a matrix view

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

challenges

1

Countless factors may affect preferences

- Genre, movie/TV series/other
- Style of action, dialogue, plot, music et al.
- Director, actors

2

Large imbalances

- Most user-item preferences are unknown
- Number of ratings per user or item may vary by several orders of magnitude
- Information to estimate individual parameters varies widely

3

Scalability

- Datasets contain millions of users/items

r_{ui} rating by user u to item i

$\hat{r}_{ui}$ predicted rating by user u to item i

conventions

error metric:

$$E(S) = \sqrt{\frac{\sum_{(u,i)\in S} (r_{ui} - \hat{r}_{ui})^2}{|S|}}$$

dataset

Ad matrix a lot sparser

As with movies, no information does not mean a negative response

- We could determine negative responses by analysis of user history

mapping to the Ad problem

Ranking metrics might be better option

- AUC of ROC curve

Need to limit to the top-k items

- We cannot show every ad to every user

In practice–combine Recommender Systems methods with predictive modeling for best performance

neighborhood methods

early, popular collaborative filtering (CF) method

Derive unknown item ratings from those of "**similar**" items (item-item variant)
A user-user flavor: rely on ratings of **like-minded users**

neighborhood based CF

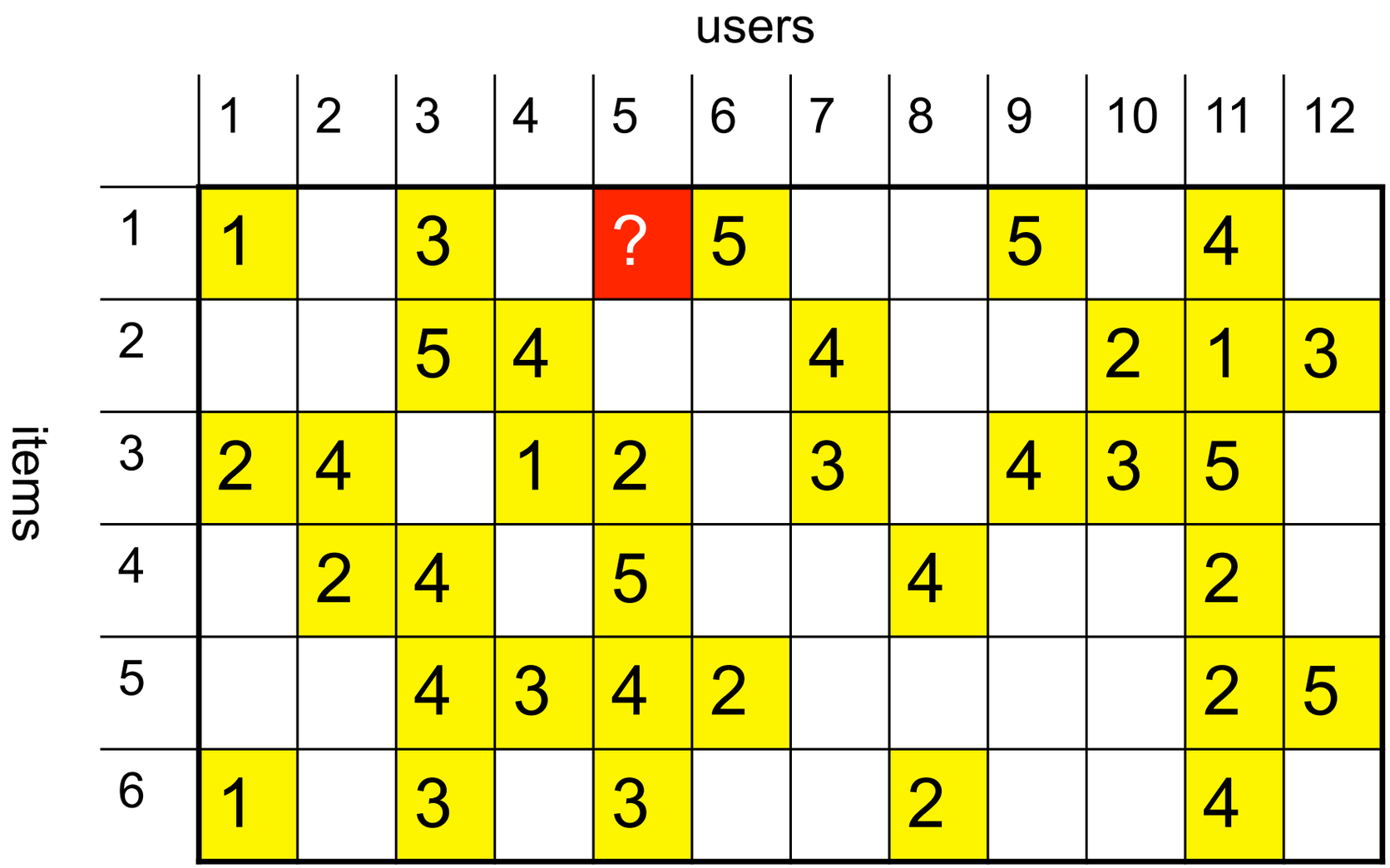
users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- unknown rating

- rating between 1 to 5

neighborhood based CF



- estimate rating of item 1 by user 5

neighborhood based CF

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Neighbor selection:
Identify items similar to 1, rated by user 5

neighborhood based CF

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Neighbor selection:
Identify items similar to 1, rated by user 5

$s_{13} = 0.2,\ s_{16} = 0.3$ **similarity weights**

neighborhood based CF

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Neighbor selection:
Identify items similar to 1, rated by user 5

use weighted average: $\frac{0.2 \times 2 + 0.3 \times 3}{0.2 + 0.3} = 2.6$

some properties

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Neighbor selection:
Identify items similar to 1, rated by user 5

use weighted average: $\frac{0.2 \times 2 + 0.3 \times 3}{0.2 + 0.3} = 2.6$

Intuitive

Easy to explain reasoning behind a recommendation

Handles new ratings/users seamlessly

No substantial preprocessing is required

Accurate (enough?)

Remove data characteristics that are unlikely to be explained by k-NN

Centering is a common practice: Remove user and item means

A more comprehensive approach eliminates time effects

data normalization

Here, we normalize by removing the baseline predictors:

global mean

$$b_{ui} = \mu + b_u + b_i$$

user bias item bias

baseline predictors

Mean rating 3.7 stars (μ)

Episode I is 0.5 stars above average (b_i)

Mary rates 0.2 stars below average (b_u)

baseline:

Mary rates Episode I 4 stars $(\mu + b_i + b_u)$

explain r_{ui}

$$\min_{b_{ui}} \sum_{(u,i)\in K} \left(\underbrace{r_{ui} - \overbrace{\mu - b_u - b_i}^{b_{ui}}}_{\text{Training error}} \right)^2 - \lambda_1 \left(\underbrace{\sum_u b_u^2 + \sum_i b_i^2}_{\text{Regularization}} \right)$$

estimation of biases

We have to estimate: b_u, b_i, μ

$$b_i = \frac{\sum_{u \in R(i)} (r_{ui} - \mu)}{\lambda_2 + |R(i)|}$$

first, estimate item biases by averaging over users that rated the item

an alternative

$$b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu - b_i)}{\lambda_3 + |R(u)|}$$

then, estimate user biases by averaging residuals over items rated by the user

define similarity measure between items s_{ij}

use s_{ij} **to select neighbors** $S^k(i;u)$

k items most similar to **i**, rated by **u**

ratings of similar items by the same user

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij} \times (r_{uj} - b_{uj})}{\sum_{j \in S^k(i;u)} s_{ij}}$$

unknown rating

how to compute item-item similarity?

User ratings for item i:

1	?	?	5	5	3	?	?	?	4	2	?	?	?	?	4	?	5	4	1	?

User ratings for item j:

?	?	4	2	5	?	?	1	2	5	?	?	2	?	?	3	?	?	?	5	4

common practice: rely on Pearson correlation coefficient

estimating item-item similarities

empirical Pearson correlation coefficient on shared support of items i and j

item-item similarity

$$\hat{\rho}_{ij} = \frac{\sum_{u \in U(i,j)} (r_{ui} - b_{ui})(r_{uj} - b_{uj})}{\sqrt{\sum_{u \in U(i,j)} (r_{ui} - b_{ui})^2 \cdot (r_{uj} - b_{uj})^2}}$$

users who rated items i **and** j

dealing with smaller support

estimates with smaller support are unreliable

$$s_{ij} = \frac{|\,U(i,j)\,| - 1}{|\,U(i,j)\,| - 1 + \lambda}\hat{\rho}_{ij}$$

penalizes small support

$$|\,U(i,j)\,| \ll \lambda \implies s_{ij} \to 0$$

$$|\,U(i,j)\,| \gg \lambda \implies s_{ij} \to \hat{\rho}_{ij}$$

improvements # I

transform similarities

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij}^2 \times (r_{uj} - b_{uj})}{\sum_{j \in S^k(i;u)} s_{ij}^2}$$

we emphasize stronger relations

improvements #2

shrink to baseline with sparse data

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij}^2 \times (r_{uj} - b_{uj})}{\lambda + \sum_{j \in S^k(i;u)} s_{ij}^2}$$

$$\sum_{j \in S^k(i;u)} s_{ij}^2 \ll \lambda \implies \hat{r}_{ui} \to b_{ui}$$

users acting on i and j

$$s_{ij} = \frac{m_{ij}}{m_i + m_j - m_{ij}}$$

Jaccard similarity

users acting on i

users acting on j

working with binary data

smoothing

$$s_{ij} = \frac{m_{ij}}{\alpha + m_i + m_j - m_{ij}}$$

data are not ratings, but binary

e.g. ad clicks

requires other natural similarity measures

How many people click on both i and j?

$$\frac{m_i \cdot m_j}{m} = \frac{m_i}{m} \times \frac{m_j}{m} \times m$$

working with binary data

$$\frac{\textbf{observed}}{\textbf{expected}} = \frac{m_{ij}}{\frac{m_i \cdot m_j}{m}}$$

smoothed variant:

$$\frac{\textbf{observed}}{\textbf{expected}} = \frac{m_{ij}}{\alpha + \frac{m_i \cdot m_j}{m}}$$

A user-user approach

dual to the item-item approach

Predict rating from ratings of similar users on the same item

user-user similarities are the basic building blocks

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in S^k(i;u)} s_{uv}(r_{vi} - b_{vi})}{\lambda + \sum_{v \in S^k(i;u)} s_{uv}}$$

when is item-item better?

Item-item is commonly considered advantageous over user-user

- when #items < #users: less item-item relations to store, more stable relations, more reliable estimation

- Item-item meshes better with new users and explaining recommendations

when is user-user preferred?

When users are the more stable anchor of the system (e.g. items are web articles that quickly expire)
When #users < #items

Matrix factorization techniques

key idea: latent factors explain variation

the basic model

users

items

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

≈

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

×

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

approximating the matrix with three factors

estimate ratings as inner products of factors

users

items

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

≈

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

×

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

approximating the matrix with three factors

estimate ratings as inner products of factors

users

items

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

2.4

≈

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

×

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

approximating the matrix with three factors

$$R = M\Sigma V$$

why can't we compute an SVD?

users

items											
	1		3			5			5		4	
			5	4			4			2	1	3
	2	4		1	2		3		4	3	5	
		2	4		5			4			2	
			4	3	4	2					2	5
	1		3		3			2			4	

2.4

≈

items			
	.1	-.4	.2
	-.5	.6	.5
	-.2	.3	.5
	1.1	2.1	.3
	-.7	2.1	-2
	-1	.7	.3

×

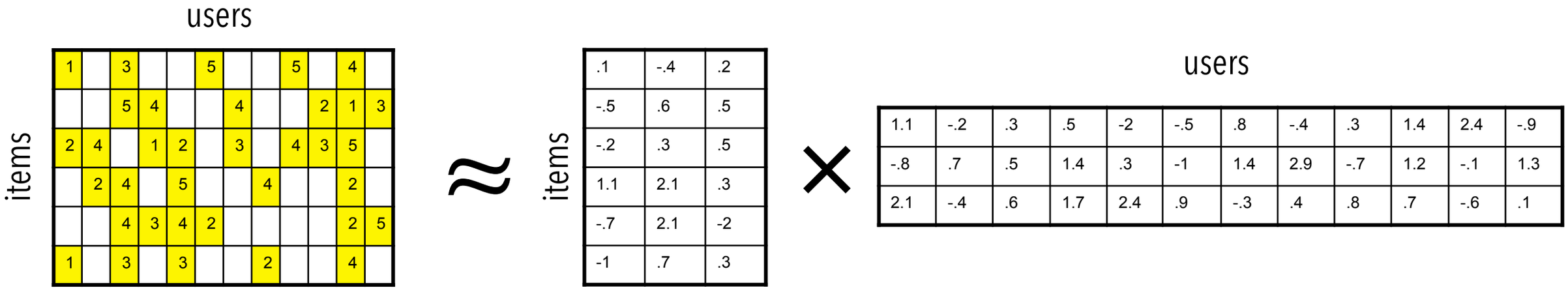
users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

SVD isn't defined when entries are missing

Regularization is necessary:

- Estimate as much signal as possible without overfitting



A regularized model

limit the values that factors can take
unlimited values can produce overfit
reduce the optimization space

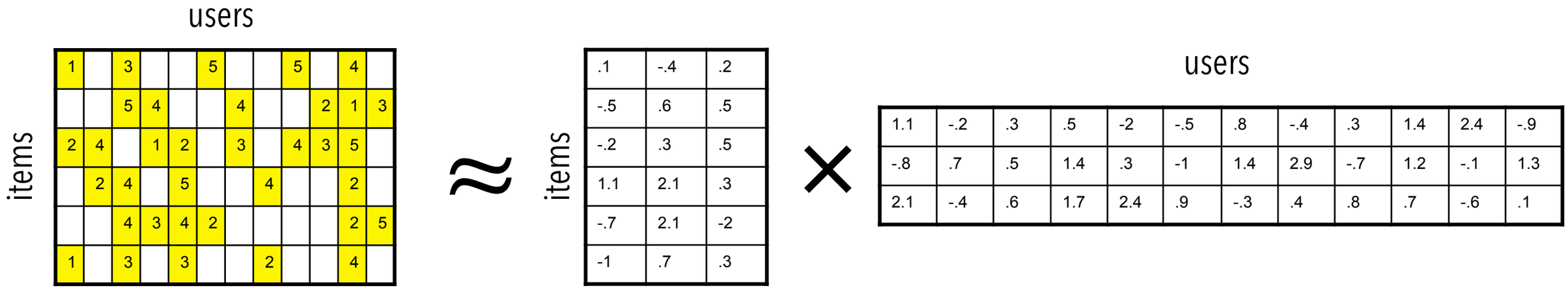
user factors: $p_u \sim N_k(\mu, \Sigma)$

item factors: $q_i \sim N_k(\gamma, \Lambda)$

k-dimensional

user-item agreement: $r_{ui} \sim N(p_u^t q_i, \epsilon^2)$

simplifications: $\mu = \gamma = 0, \Sigma = \Lambda = \lambda I$



cost function

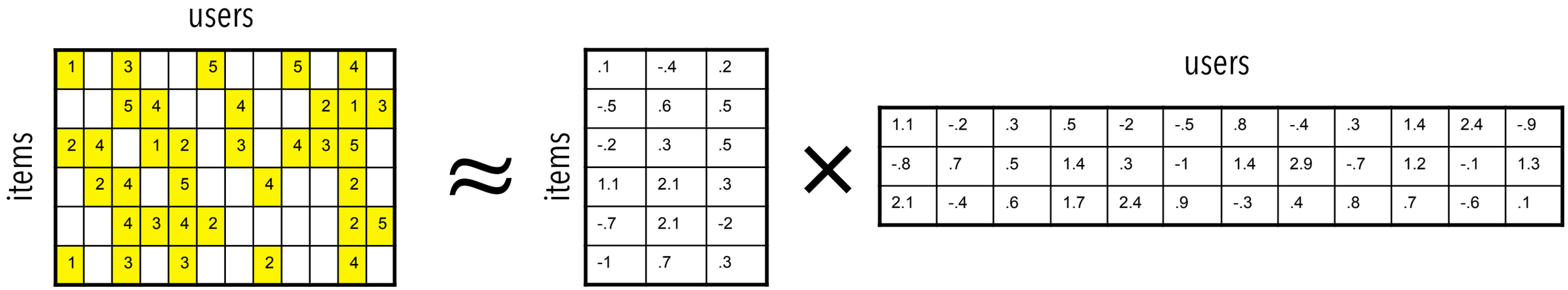
$$\min_{p,q} = \sum_{r_{ui}} \left(r_{ui} - p_u^t q_i\right)^2 + \lambda \left(\| p_u \|^2 + \| q_i \|^2\right)$$

r_{ui} rating

p_u user factor of u

q_i item factor of i

optimize by either stochastic gradient descent or alternating least squares



SGD optimization

for each training example r_{ui}

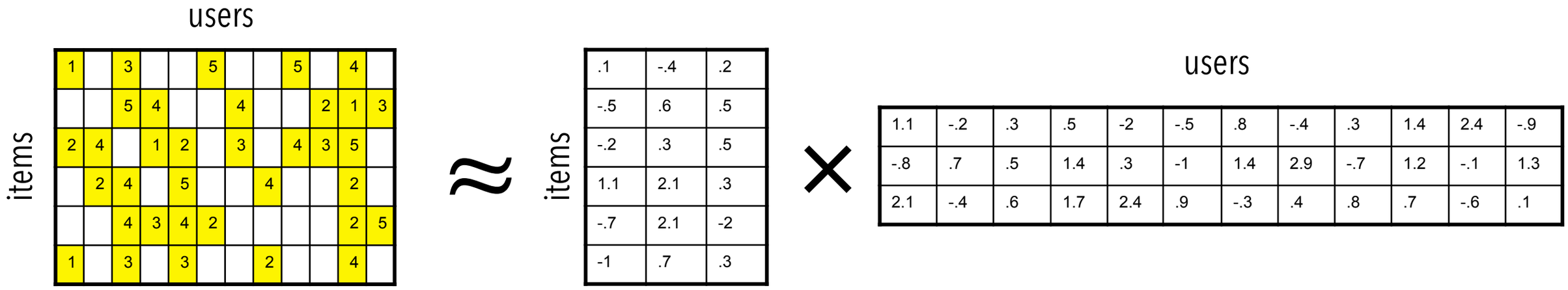
compute prediction error: $e_{ui} = r_{ui} - p_u^t q_i$

update item factor: $q_i \leftarrow q_i + \gamma(p_u e_{ui} - \lambda q_i)$

update user factor: $p_u \leftarrow p_u + \gamma(q_i e_{ui} - \lambda p_u)$

iterate till convergence

tune two constants: γ (step size), λ (regularization)



Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In Proceedings of KDD cup and workshop, volume 2007, pages 5–8.

matrix factorization with bias

bias of i

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^t q_i$$

bias of u

global average

$$\min_{p,q,b} = \sum_{r_{ui}} \left(r_{ui} - \hat{r}_{ui}\right)^2 + \lambda \left(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2 \right)$$

regularization

ratings: values vs. occurrences

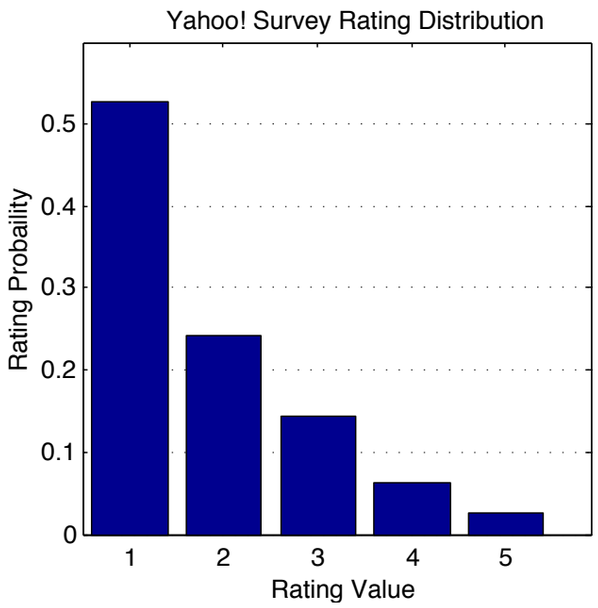
There is information in the fact that user has rated a movie

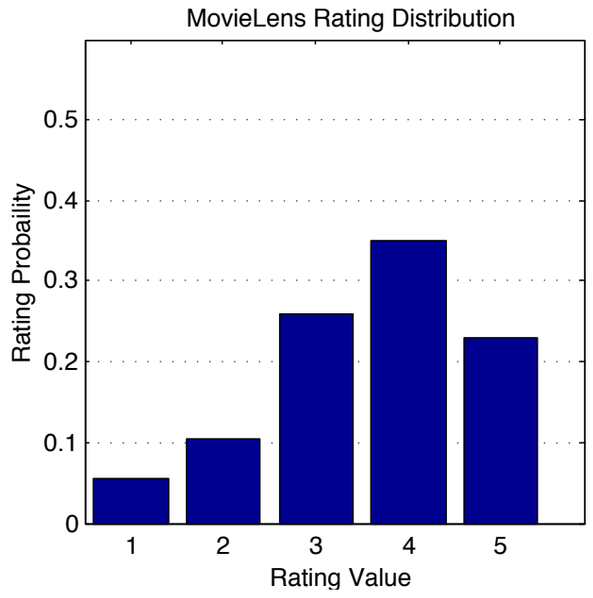
The user chose to see the movie
The user chose to rate the movie
The choice depends on many factors

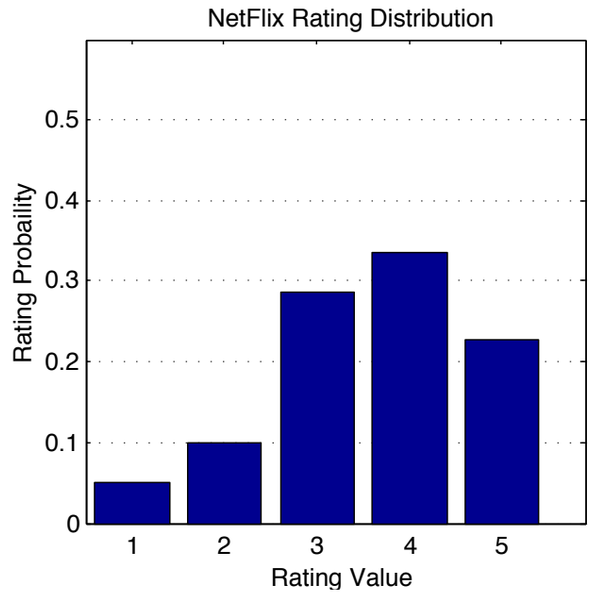
Can we use this information to improve the factorization?

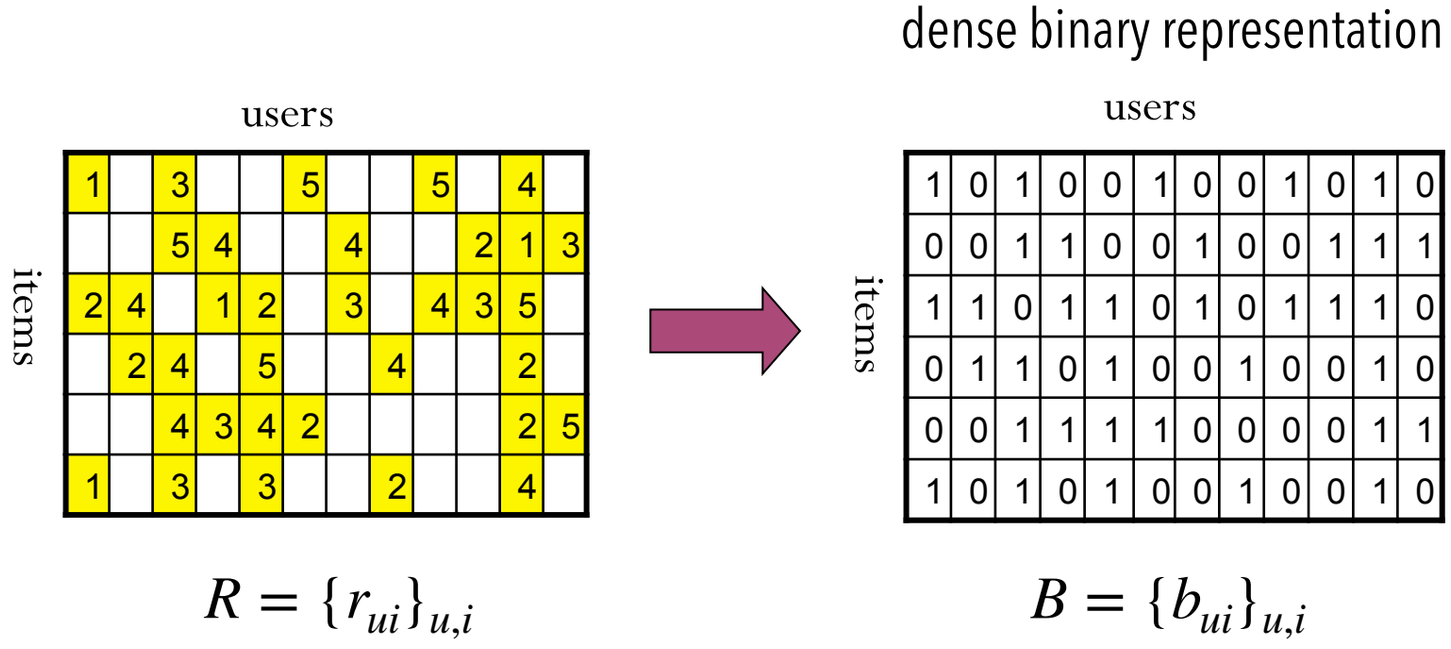
Marlin, B. M., Zemel, R. S., Roweis, S., and Slaney, M. (2007). Collaborative filtering and the missing at random assumption. In Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, UAI'07, pages 267–275, Arlington, Virginia, United States. AUAI Press.

not at random!









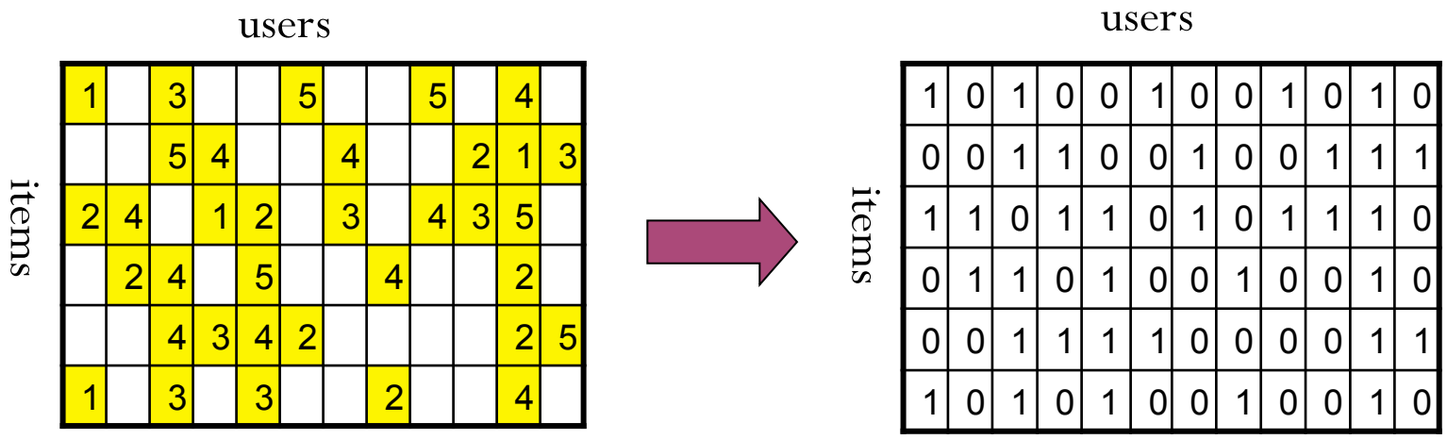
$R = \{r_{ui}\}_{u,i}$ $B = \{b_{ui}\}_{u,i}$

a key piece of information

Characterize users by which items they rated, rather than how they rated

A dense binary representation of the data:

dense binary representation



$R = \{r_{ui}\}_{u,i}$ $B = \{b_{ui}\}_{u,i}$

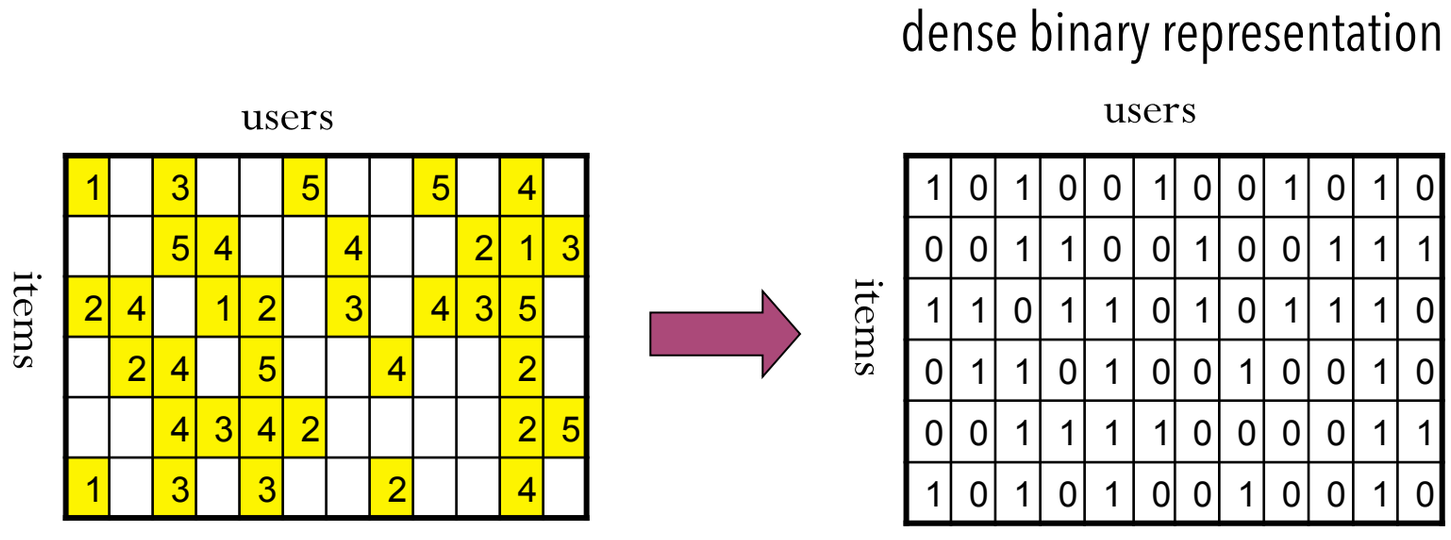
$$r_{ui} = \mu + b_u + b_i + p_u^t q_i$$

$$b_{ui} = p_u^t x_i$$

factoring in the binary view

user factors shared across models

each item i is associated with two factor vectors: q_i, x_i



$$R = \{r_{ui}\}_{u,i} \qquad B = \{b_{ui}\}_{u,i}$$

$$\forall i: \; b_{ui} = x_i^t p_u,$$

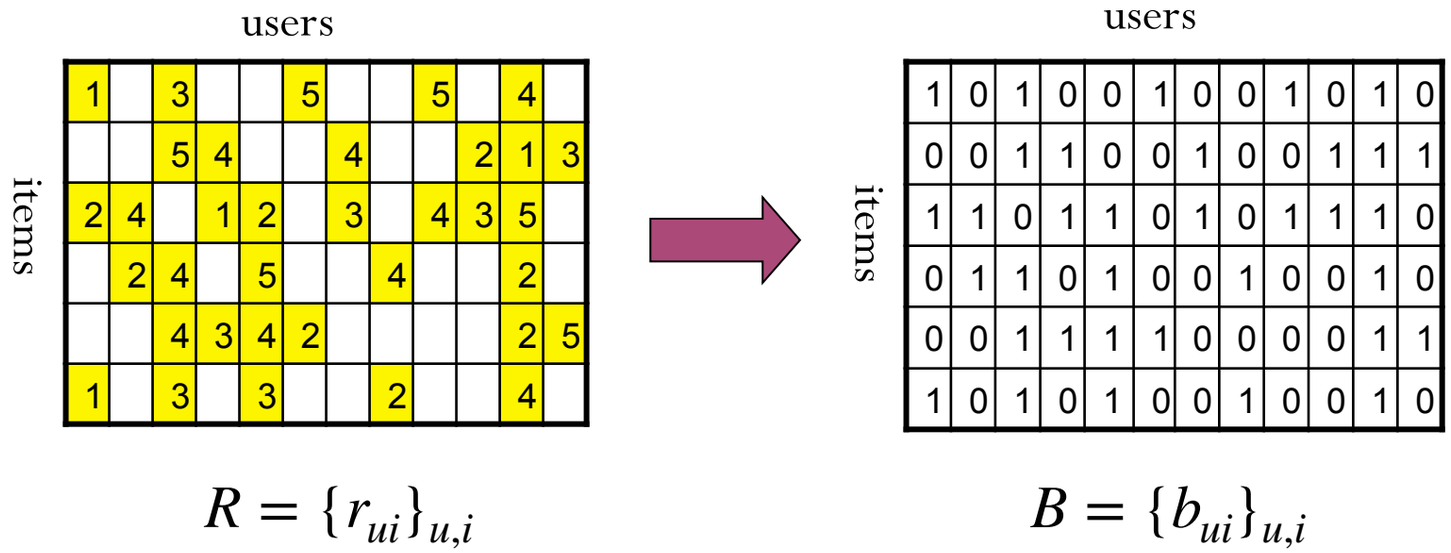
$$p_u = (XX^t)^{-1} XB_u \qquad X = (x_1, x_2, \ldots, x_n)$$

$$p_u \propto XB_u = \sum_i b_{uj} x_j \qquad B_u = (b_{u1}, b_{u2}, \ldots, b_{un})$$

user factors are indirectly defined by item factors: sum of item factors for items rated by u

factoring in the binary view

dense binary representation



$R = \{r_{ui}\}_{u,i}$ $B = \{b_{ui}\}_{u,i}$

ratings model $r_{ui} = \mu + b_u + b_i + q_i^t p_u$

binary view $p_u \propto XB_u = \sum_i b_{uj} x_j = \sum_{\text{j rated by u}} x_j$

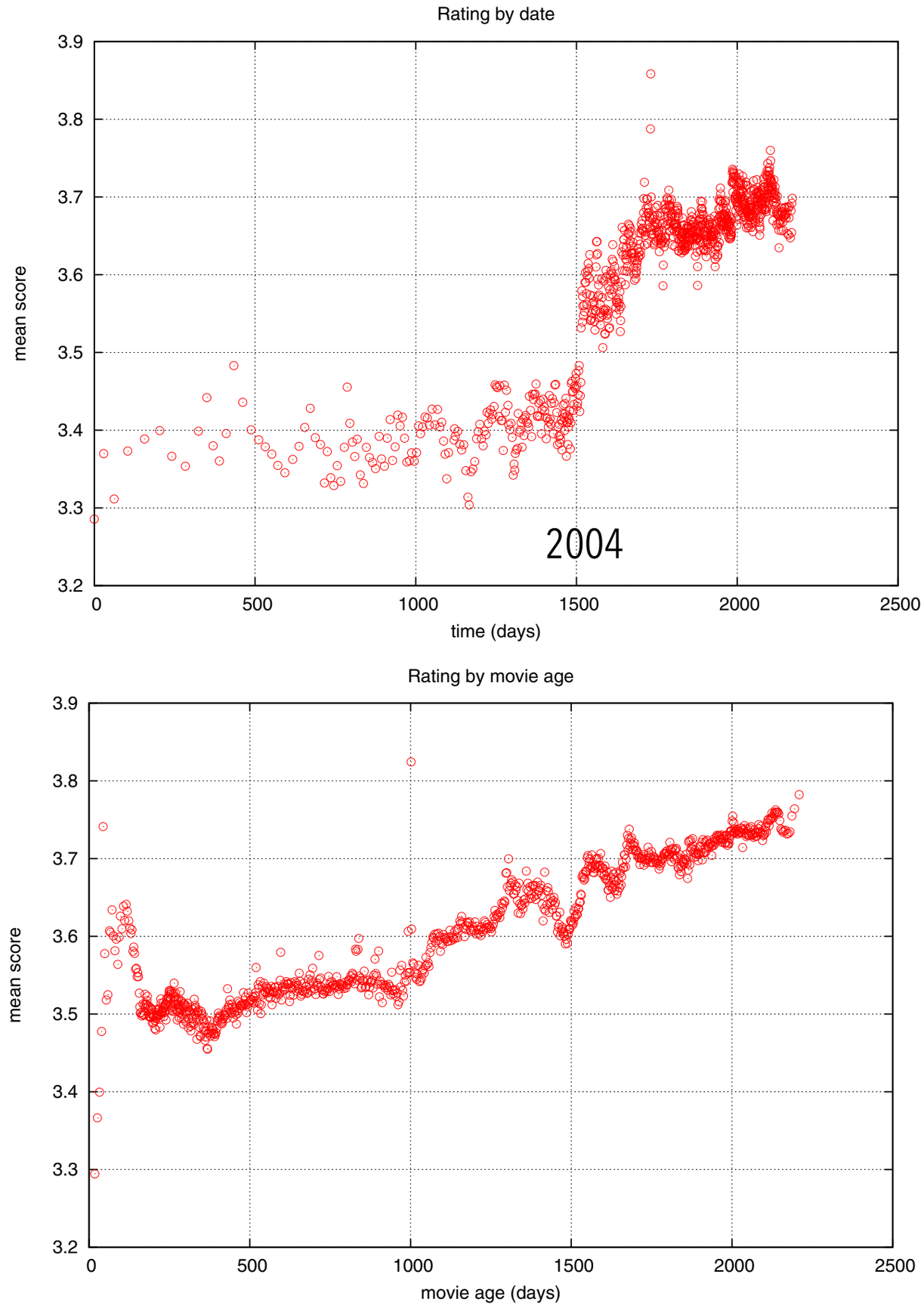
$$r_{ui} = \mu + b_u + b_i + q_i^t \left(p_u + \sum_i b_{uj} x_j \right)$$

integrating the two

variants of "who rated what" produces a 7% improvement over Netflix baseline

temporal dynamics

are movies getting better?



multiple sources

Item-side effects:

- Product perception and popularity are constantly changing

- Seasonal patterns influence items' popularity

multiple sources

User-side effects:

- Customers redefine their taste
- Transient, short-term bias; anchoring
- Drifting rating scale
- Change of rater within household

Multiple effects: Both items and users are changing over time
- Scarce data per target

Inter-related targets: Signal needs to be shared among users– foundation of collaborative filtering
- cannot isolate multiple problems

challenges

Common “concept drift” methodologies won’t hold.
- E.g., underweighting older instances is unappealing

Koren, Y. (2009). Collaborative filtering with temporal dynamics. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pages 447–456, New York, NY, USA. ACM.

$$r_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^t p_u(t)$$

$$b_u(t) = f(u,t), \quad b_i(t) = g(i,t), \quad p_u(t) = h(u,t)$$

need to find adequate functional forms

parameterizing the model

General guidelines:

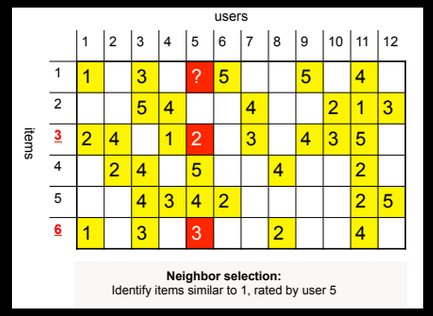
- Items show slower temporal changes
- Users exhibit frequent and sudden changes

$p_u(t)$ expensive to model

Gain flexibility by heavily parameterizing the functions

8% improvement over Netflix baseline

A set of techniques to recommend items based on explicit (rating) or implicit (page visits, ad clicks)



neighborhood CF

use of baselines

global mean

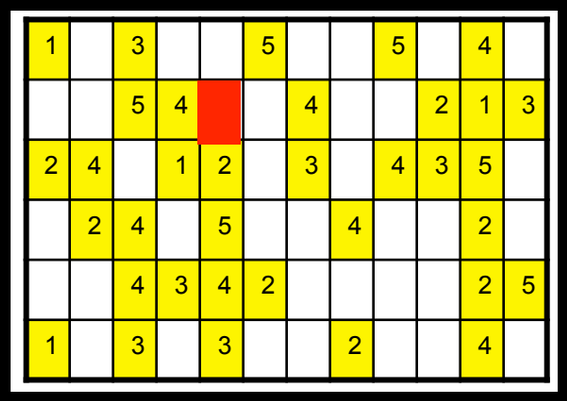
$$b_{ui} = \mu + b_u + b_i$$

user bias item bias

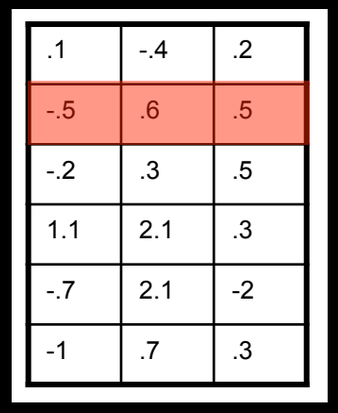
NETFLIX

summary

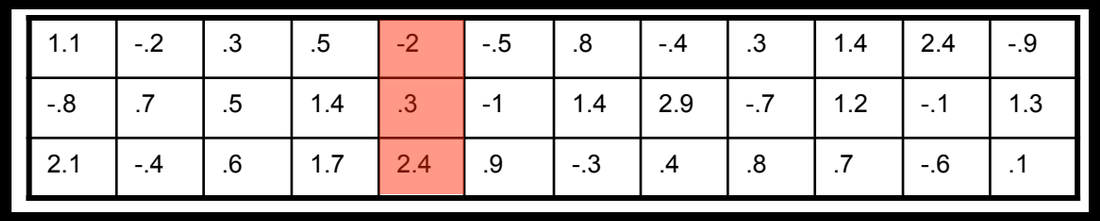
Recommender systems



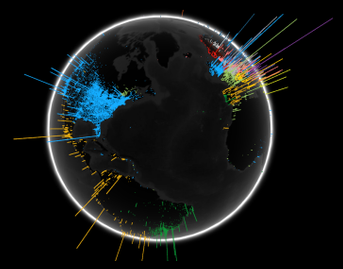
≈



×



Introduction



Web search

Game Theory

Auctions

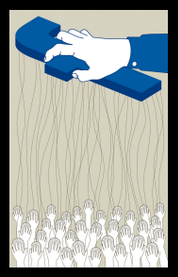
Contextual Ads

Display Ads

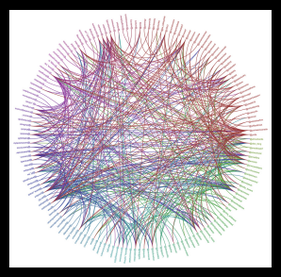
Behavioral targeting

NETFLIX

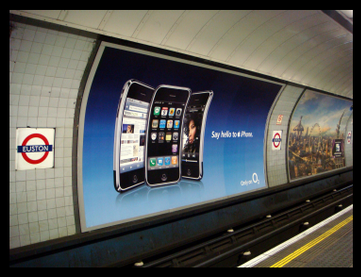
Recommender systems



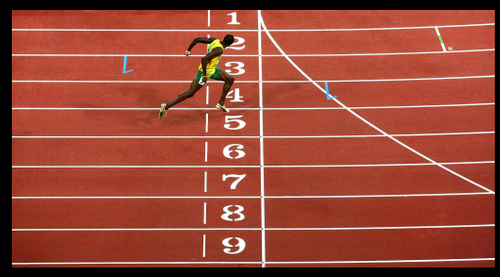
Privacy



Networks



Emerging areas



Final Presentations