Introduction to Computational Advertising

MS&E 239

Stanford University

Autumn 2011

Instructors: Dr. Andrei Broder and Dr. Vanja Josifovski

Yahoo! Research

General course info

- Course Website: http://www.stanford.edu/class/msande239/
- Instructors
  - **Dr. Andrei Broder**, Yahoo! Research, broder@yahoo-inc.com
  - **Dr. Vanja Josifovski**, Yahoo! Research, vanjaj@yahoo-inc.com
- TA: **Krishnamurthy Iyer**
  - Office hours: Tuesdays 6:00pm-7:30pm, Huang
- Course email lists
  - Staff: msande239-aut1112-staff
  - All: msande239-aut1112-students
  - Please use the staff list to communicate with the staff
- Lectures: 10am ~ 12:30pm Fridays in HP
- Office Hours:
  - After class and by appointment
  - Andrei and Vanja will be on campus for 2 times each to meet and discuss with students. Feel free to come and chat about even issues that go beyond the class.

Course Overview (subject to change)

1. 09/30 Overview and Introduction
2. 10/07 Marketplace and Economics
3. 10/14 Textual Advertising 1: Sponsored Search
4. 10/21 Textual Advertising 2: Contextual Advertising
5. 10/28 Display Advertising 1
6. 11/04 Display Advertising 2
7. 11/11 Targeting
8. 11/18 Recommender Systems
9. 12/02 Mobile, Video and other Emerging Formats
10. 12/09 Project Presentations

Lecture 8:
Recommender Systems for Display Advertising Targeting

Disclaimers

- This talk presents the opinions of the authors. It does not necessarily reflect the views of Yahoo! inc or any other entity.
- Algorithms, techniques, features, etc mentioned here might or might not be in use by Yahoo! or any other company.
- This lecture is largely based on slides by **Yehuda Koren**. His help very much appreciated! Other contributors: Amr Ahmed and Alex Smola

Lecture 8 plan

- Problem definition
- Neighborhood based approaches
- Matrix factorization approaches
- Generative models for factorization: Latent Dirichlet Allocation
- External information in campaign optimization (time permitting)

Checkpoint - targeting

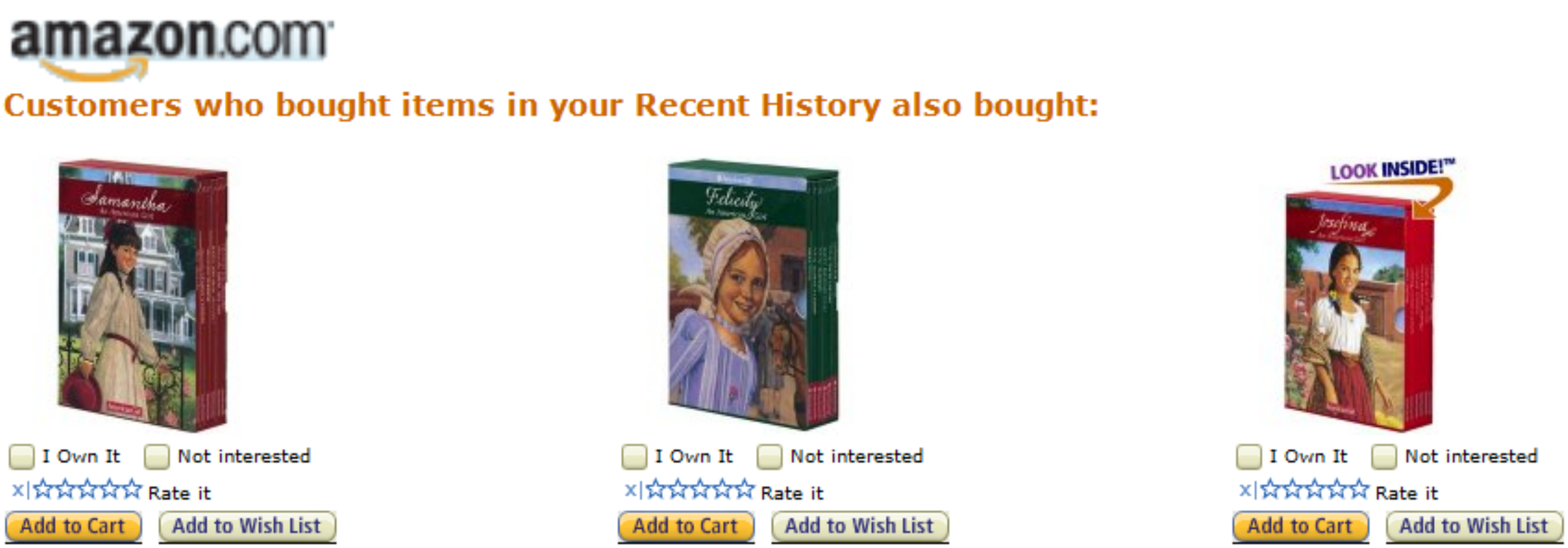
- Targeting is a key step in differentiation of impressions and extracting value!
- Traditional targeting: demo, geo, BT
  - How to get the data from the user?
  - Infer the data from historical activity
- One of the key step in targeting is user profile generation
  - Generative models to assign probability of a sequence of events
  - Weighting based on time, event type and content
  - Predict the counts of events in certain categories
  - Clustering and other unsupervised techniques useful – more to come in the next lecture

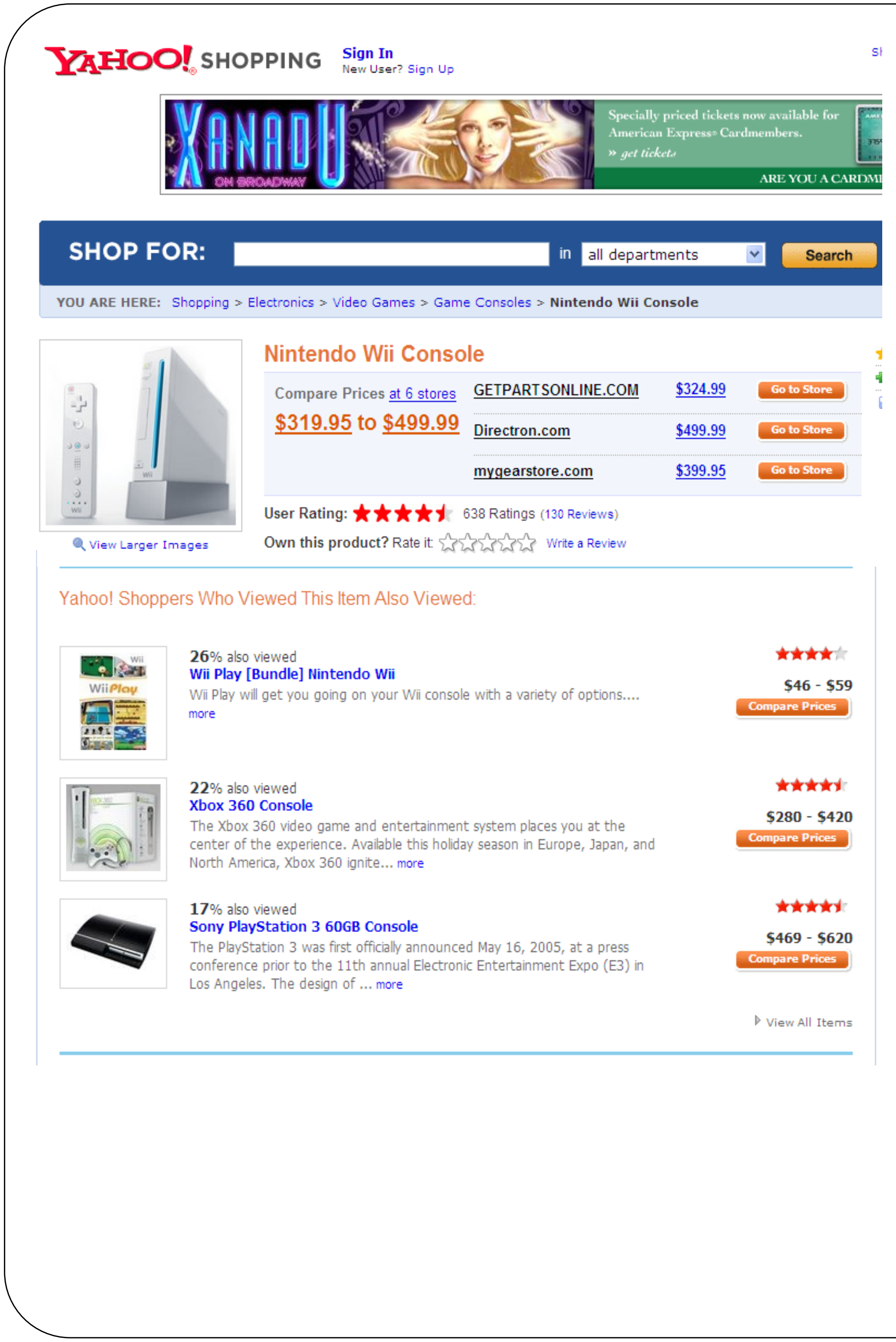
Recommender systems

- What is actually recommender system technology?
- 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 opaque
- 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
- Shown to be effective in many domains, including advertising
- Slides mostly on movie ratings, we will discuss the similarities/differences wit the ad domain

Collaborative filtering

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




YAHOO! SHOPPING
Sign In
New User? Sign Up
XANADU ON BROADWAY
Specially priced tickets now available for American Express Cardmembers.
» get tickets
ARE YOU A CARDM
SHOP FOR:
in all departments
Search
YOU ARE HERE: Shopping > Electronics > Video Games > Game Consoles > Nintendo Wii Console
Nintendo Wii Console
Compare Prices at 6 stores
$319.95 to $499.99
GETPARTSONLINE.COM $324.99 Go to Store
Directron.com $499.99 Go to Store
mygearstore.com $399.95 Go to Store
User Rating: 638 Ratings (130 Reviews)
Own this product? Rate it: Write a Review
View Larger Images
Yahoo! Shoppers Who Viewed This Item Also Viewed:
26% also viewed
Wii Play [Bundle] Nintendo Wii
Wii Play will get you going on your Wii console with a variety of options....
more
$46 - $59
Compare Prices
22% also viewed
Xbox 360 Console
The Xbox 360 video game and entertainment system places you at the center of the experience. Available this holiday season in Europe, Japan, and North America, Xbox 360 ignite... more
$280 - $420
Compare Prices
17% also viewed
Sony PlayStation 3 60GB Console
The PlayStation 3 was first officially announced May 16, 2005, at a press conference prior to the 11th annual Electronic Entertainment Expo (E3) in Los Angeles. The design of ... more
$469 - $620
Compare Prices
View All Items


NETFLIX
Browse
Movies You'll
Community
Queue
DVD Sale $5.99+
Watch Now
Movies, actors, directors
Search
Home
Genres
New Releases
Previews
Netflix Top 100
Critics' Picks
Award Winners
High Crimes (2002)
At A Glance
Friends
Member Reviews
Critics
More Like This
San Francisco attorney Claire Kubik (Ashley Judd) teams up with a former military attorney (Morgan Freeman) to defend her husband, Tom (James Caviezel), in military court. The military has declared Tom a deserter, charging him with participating in a mass killing in El Salvador. Can Claire get him free? What's more, as the disturbing top-secret details of the crime are revealed, will she want to?
Watch the Preview
Add
Average of raters like you: 4.0 stars
Average of 1,325,394 ratings: 3.7 stars
WHY IS THIS RECOMMENDED
Don't Say a Word
How to Lose a Guy in 10 Days
Sweet Home Alabama
Man on Fire
The Italian Job
Ghost
Double Jeopardy
Taking Lives
DETAILS
PG-13 For violence, sexual content and language
PARENTS: for 13+ (more)
Length:


Netflix Prize
Home
Rules
Leaderboard
Register
Update
Submit
Download
NETFLIX
Movies For You
You really liked it...
Welcome!
The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences. Improve it enough and you win one (or more) Prizes. Winning the Netflix Prize improves our ability to connect people to the movies they love.
Read the Rules to see what is required to win the Prizes. If you are interested in joining the quest, you should register a team.
You should also read the frequently-asked questions about the Prize. And check out how various teams are doing on the Leaderboard.
Good luck and thanks for helping!

Movie rating data

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	?

Alternate view of the data: matrix

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	

Major 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
    - Some datasets contain millions of users/items

Conventions

- r_{ui} - rating by user u to item i
- $\hat{r}_{ui}$ - predicted rating by user u to item i
- Error function:

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

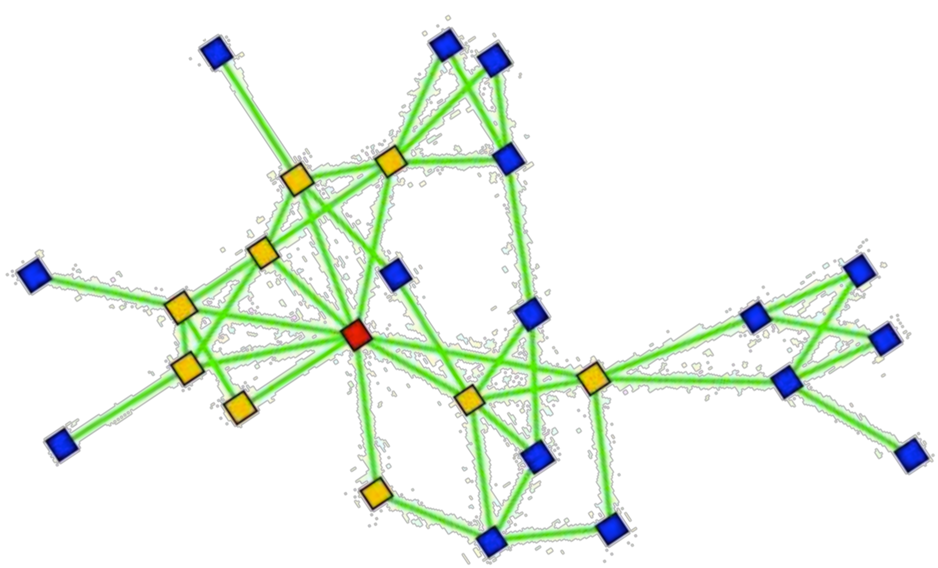
How does this map to the ad world

- Ad matrix a lot sparser
- As with movies, no info does not mean negative response
  - We could determine negative responses by analysis of user history
- 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 rec sys methods with predictive modeling for best performance

Neighborhood methods

Neighborhood-based CF

- Earliest and most popular collaborative filtering method
- Derive unknown ratings from those of "similar" items (item-item variant)
- A parallel 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 (white) - rating between 1 to 5 (yellow)

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	

- 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	

Compute similarity weights:

$s_{13}=0.2$, $s_{16}=0.3$

Neighborhood-based CF

users

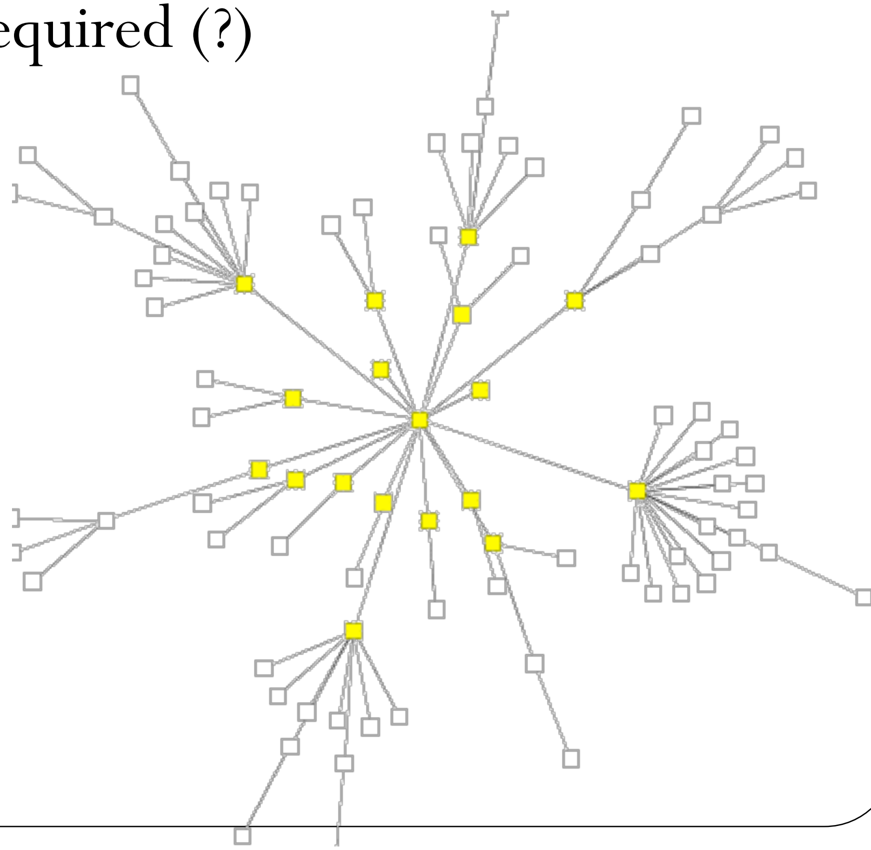
items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		2.6	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Predict by taking a weighted average:

$(0.2*2+0.3*3)/(0.2+0.3)=2.6$

Properties of neighborhood-based CF

- Intuitive
- Easy to explain reasoning behind a recommendation
- Handles new ratings/users seamlessly
- No substantial preprocessing is required (?)
- Accurate (enough?)



Data normalization

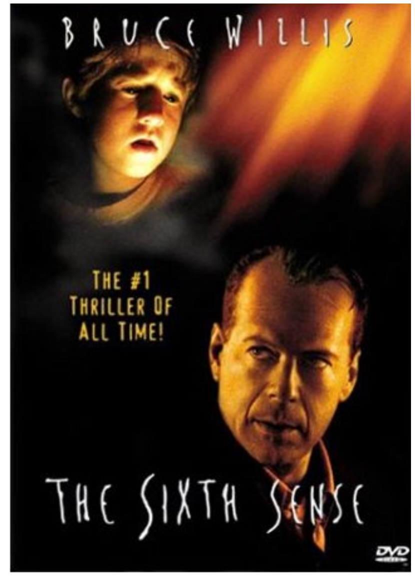
- Need to identify relations and mix ratings across items/ users
- However:
- User and item-specific variability masks fundamental relationships
- Examples:
  - Some items are systematically rated higher
  - Some items were rated by users that tend to rate low
  - Ratings change along time
- Normalization is critical to the success of a k-NN approach

Data normalization

- Remove data characteristics that are unlikely to be explained by k-NN
- Common practice is to use centering:
  Remove user- and item-means
- A more comprehensive approach eliminates additional interfering variability such as time effects
  See "global effects" @ "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights", ICDM'07
- Here, we normalize by removing the baseline predictors:

$$b_{ui} = \underbrace{\mu}_{\text{Global mean}} + \underbrace{b_u}_{\text{User bias}} + \underbrace{b_i}_{\text{Item bias}}$$

Baseline predictors (biases)



- Mean rating: 3.7 stars (μ)
- *The Sixth Sense* is 0.5 stars above avg (b_i)
- *Joe* rates 0.2 stars below avg (b_u)

➔Baseline estimation:
Joe will rate *The Sixth Sense* 4 stars ($\mu + b_i + b_u$)

Estimation of biases

- Try to explain each r_{ui} in the train set as $\mu + b_u + b_i$
- Solve the **regularized** least squares problem:

$$\min_{b_*} \sum_{(u,i)\in K} \underbrace{(r_{ui} - \mu - b_u - b_i)^2}_{\text{Error for a training case}} + \underbrace{\lambda_1 \left(\sum_u b_u^{\,2} + \sum_i b_i^{\,2}\right)}_{\text{Regularization}}$$

An alternative:

- First, estimate item biases by averaging over users that rated the item:

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

- Then, estimate user biases by averaging residuals over items rated by the user:

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

Residual: ratings of similar items by the same user

1. Define a similarity measure between items: s_{ij}
2. Use s_{ij} to select neighbors – $s^k(i;u)$:
   k items most similar to i, that were rated by u
3. Estimate unknown rating, r_{ui}, as the weighted average rating that u gave to the neighbors:

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

- How to compute item-item similarity?

Estimating item-item similarities

- Common practice – rely on Pearson correlation coeff
- Challenge – non-uniform user support of item ratings, each item rated by a distinct set of users

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

- Compute correlation over shared support

Estimating item-item similarities

- Empirical Pearson correlation coefficient on shared support of items i and j:

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

U(i,j) contains the users who rated both items i and j

- Estimates with smaller supports are less reliable
- Use shrunk correlation coeff as a similarity measure:

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

- λ penalizes small supports:

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

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

Improvements to common practice

- Use transformed similarities as interpolation coeff's
- E.g., by squaring we emphasize stronger relations:

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

- See A. Toscher, M. Jahrer and R. Legenstein, "Improved Neighborhood-Based Algorithms for Large-Scale Recommender Systems" for sigmodial transformations
- Shrink towards baseline when not enough neighborhood info is available:

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

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

Similarities for binary data

- Often data is not ratings but binary. Example: ad clicks and conversions
- This requires other natural similarity measures

- Notation:
  m_i - #users acting on i
  m_{ij} - #users acting on both i and j
  m - overall #users

(1) Jaccard similarity:

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

- Shrink estimates:

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

Similarities for binary data #2

(2) Observed/Expected ratio:

- Under random sampling, expected i-j |intersection| :

$$m_i \cdot m_j / m$$

(expectation of a hypergeometric distribution)

$$\Rightarrow \quad \frac{\text{observed}}{\text{expected}} = \frac{m_{ij}}{(m_i \cdot m_j / m)}$$

- As usual, need to shrink:

$$s_{ij} = \frac{m_{ij}}{\alpha + (m_i \cdot m_j / m)}$$

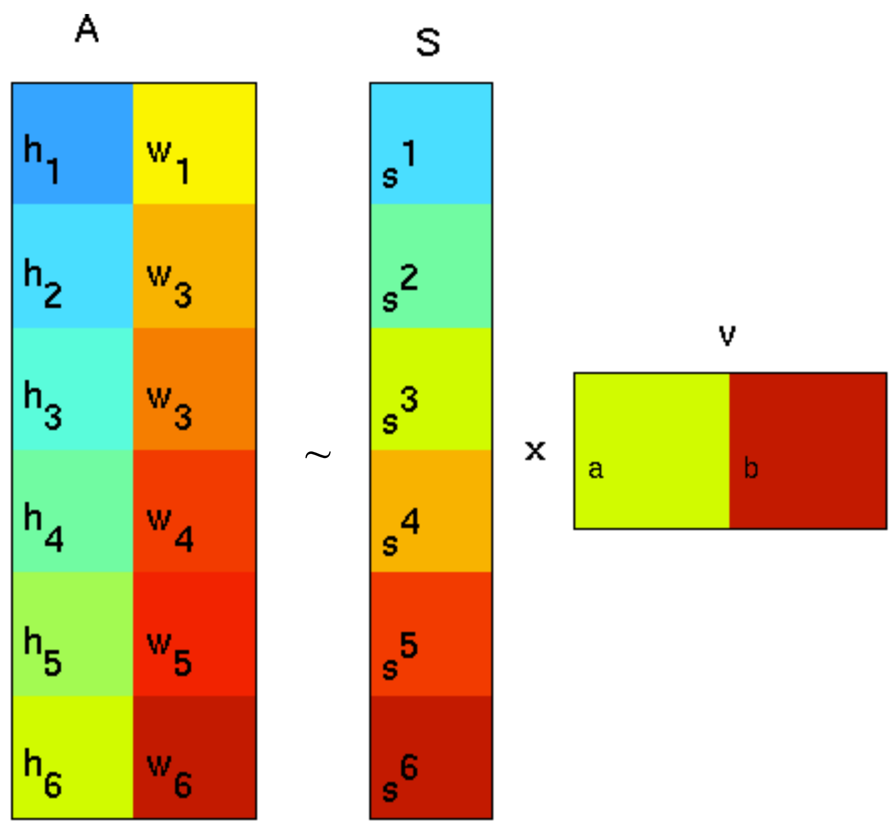
A user-user approach

- Dual to the so-far described item-item approach (with similar derivation)
- Predict a rating from ratings of similar users on the same item
- Building stones are user-user similarities s_{uv}:

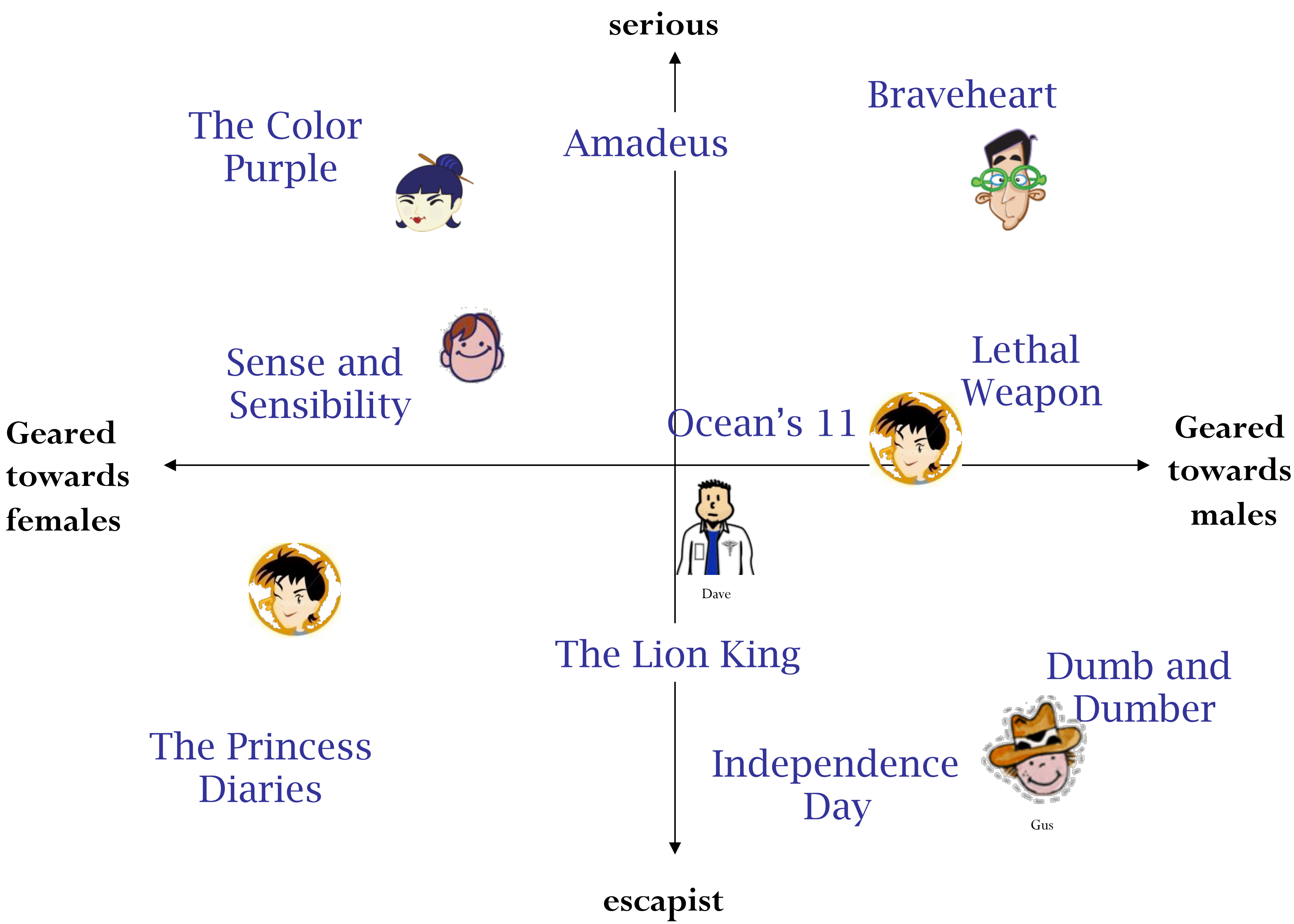
$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in S^k(u;i)} s_{uv}(r_{vi} - b_{vi})}{\lambda + \sum_{v \in S^k(u;i)} s_{uv}}$$

- 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 rec's
- In some cases user-user becomes more sensible:
  - When users are the more stable anchor of the system (e.g. items are web articles that quickly expire)
  - When #users < #items

Part II:
Matrix factorization techniques



Latent factor models



Basic matrix factorization 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

$\bullet$

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

A rank-3 SVD approximation

Estimate unknown ratings as inner-products of factors:

users

items

1		3			5			5		4	
		5		?		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

A rank-3 SVD approximation

Estimate unknown ratings as inner-products of factors:

users

items

1		3			5			5		4	
		5		?		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

A rank-3 SVD approximation

Estimate unknown ratings as inner-products of factors:

users

items

1		3			5			5		4	
		5	4	2.4		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

A rank-3 SVD approximation

Matrix factorization model

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	

~

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

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

Why can't use standard SVD $R = M \Sigma V$?

- SVD isn't defined when entries are missing
- Regularization is necessary:
  Estimate as much signal as possible where there are sufficient data, without over fitting where data are scarce

A regularized model

- Limit the values the factors can take

  Unlimited values will produce an overfit

  Reduce the optimization space
- User factors:

  Model a user u as a vector $p_u \sim N_k(\mu, \Sigma)$
- Item factors:

  Model an item i as a vector $q_i \sim N_k(\gamma, \Lambda)$
- Ratings:

  Measure "agreement" between u and i: $r_{ui} \sim N(p_u^T q_i, \varepsilon^2)$
- Simplifying assumptions:

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

Matrix factorization as a cost function

$$\mathrm{Min}_{p_*, q_*} \sum_{\text{known } r_{ui}} \left(r_{ui} - p_u^T q_i\right)^2 + \underbrace{\lambda\left(\|p_u\|^2 + \|q_i\|^2\right)}_{\text{regularization}}$$

p_u - user-factor of u

q_i - item-factor of i

r_{ui} - rating by u for i

- Optimize by either stochastic gradient-descent or alternating least squares

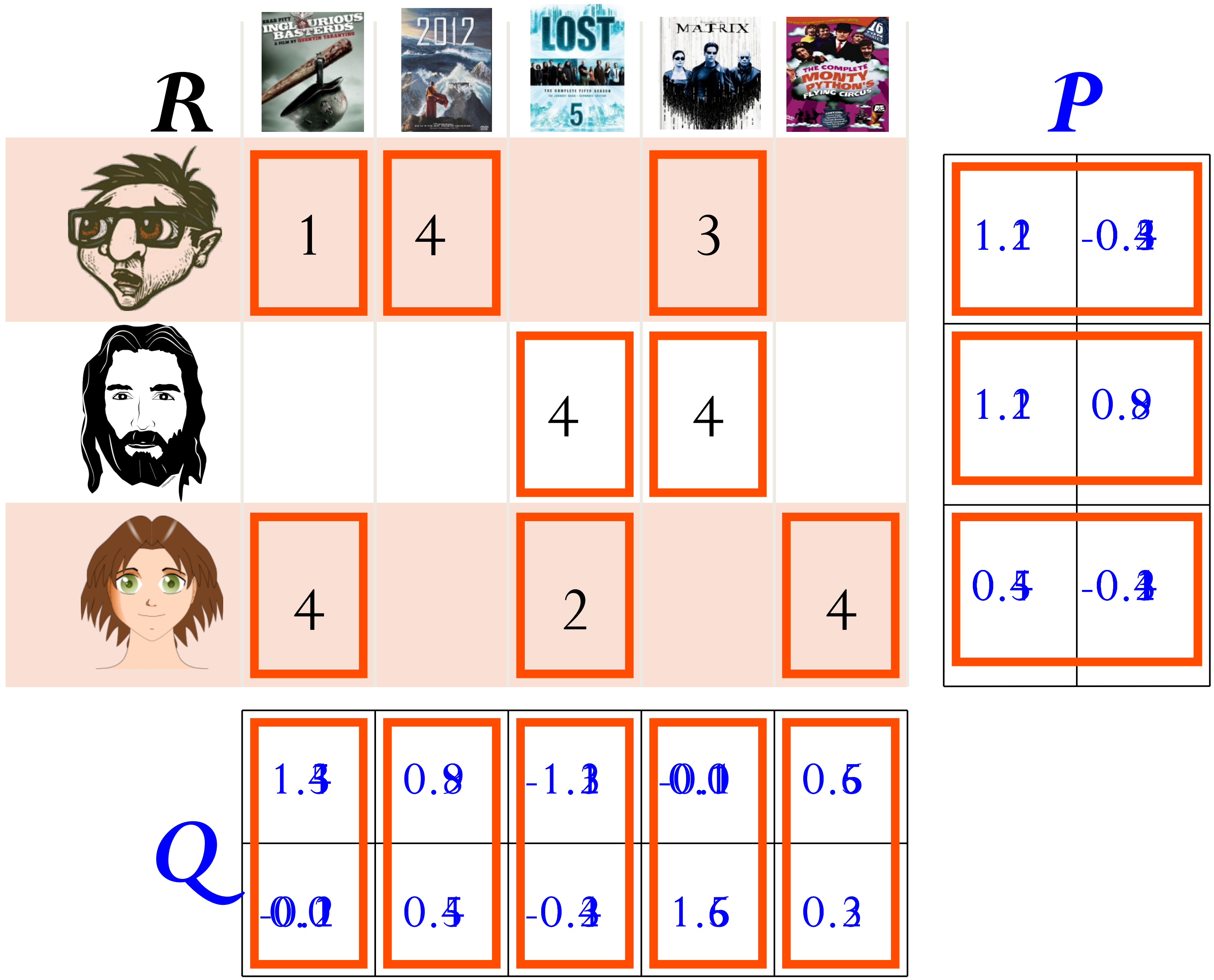
Stochastic gradient descent optimization

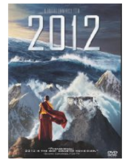
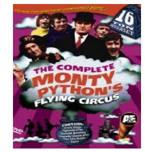
Perform till convergence:

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

- Two constants to tune: γ (step size) and λ (regularization)
- Find values that minimize error on **validation** set

See, e.g., ***Simon Funk, "Netflix Update: Try This at Home",***
http://sifter.org/~simon/journal/20061211.html


R
P
Q
2012
LOST
MATRIX
1
4
3
4
4
4
2
4

R					
	1	4	3.3	3	2.4
	-0.5	3.5	4	4	1.5
	4	4.9	2	1.1	4

P

1.4	1.1
0.9	1.9
2.5	-0.3

Q

1.5	2.1	1.0	0.7	1.6
-1.0	0.8	1.6	1.8	0.0

Matrix factorization with biases

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

Baseline predictors:
μ – global average
b_u – bias of u
b_i – bias of i

➔Minimization problem:

$$\mathrm{Min}_{p_*, p_*, b_*} \sum_{\text{known } r_{ui}} \left(r_{ui} - (\mu + b_u + b_i + p_u^T q_i) \right)^2 + \underbrace{\lambda \left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 \right)}_{\text{regularization}}$$

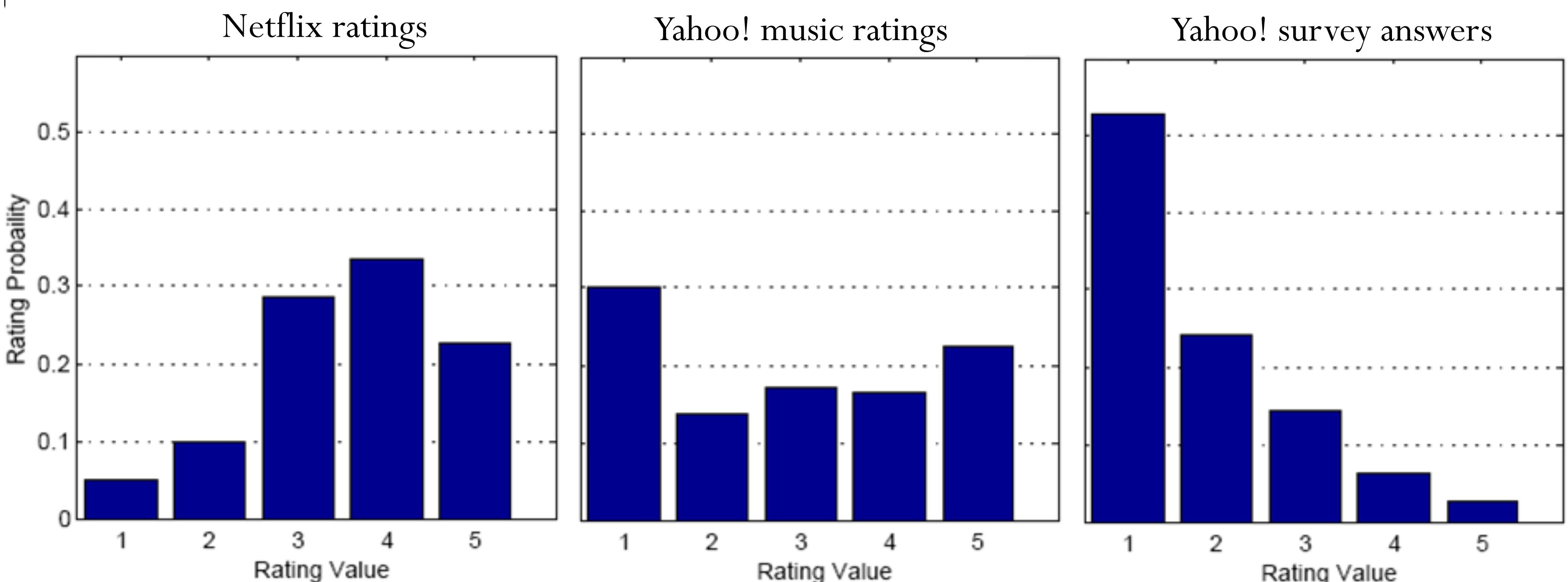
See, e.g., **A. Paterek, "Improving regularized singular value decomposition for collaborative filtering",** Proc. KDD Cup and Workshop 2007

Ratings values vs rating 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?

Ratings are not given at random!

Distribution of ratings



B. Marlin et al., "Collaborative Filtering and the Missing at Random Assumption" UAI 2007

Which items users rate?

- A powerful source of information:
  Characterize **users** by **<u>which</u>** items they rated, rather than **<u>how</u>** they rated
- ➔ A dense binary representation of the data:

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	

$$R = \left\{ r_{ui} \right\}_{u,i}$$

users

items

1	0	1	0	0	1	0	0	1	0	1	0
0	0	1	1	0	0	1	0	0	1	1	1
1	1	0	1	1	0	1	0	1	1	1	0
0	1	1	0	1	0	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
1	0	1	0	1	0	0	1	0	0	1	0

$$B = \left\{ b_{ui} \right\}_{u,i}$$

Factoring the binary view

- Describe both R and B using a factor model:

$$r_{ui} = \mu + b_u + b_i + p_u^T q_i$$
$$b_{ui} = p_u^T x_i$$

- User factors are shared across models
- Each item i is associated with two factor vectors: q_i and x_i

Factoring the binary view

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

$$X = (x_1, x_2, \cdots, x_n)$$

$$B_u = (b_{u1}, b_{u2}, \cdots, b_{un})$$

$$p_u = (XX^T)^{-1} XB_u$$

$$p_u \propto XB_u = \sum_j b_{uj} x_j$$

User factor is indirectly defined by item factors:
sum of item factors for items rated by u

Integrating ratings and binary views

So far:

- Each user u is associated with a factor vector p_u
- Each item i is associated with two factor vectors: q_i and x_i
- The pure rating model:

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

- The binary view of the user factors:

$$p_u \propto \sum_j b_{uj} x_j = \sum_{j \text{ rated by } u} x_j$$

Integrating ratings and binary views

So far:

- Each user u is associated with a factor vector p_u
- Each item i is associated with two factor vectors: q_i and x_i
- The pure rating model:

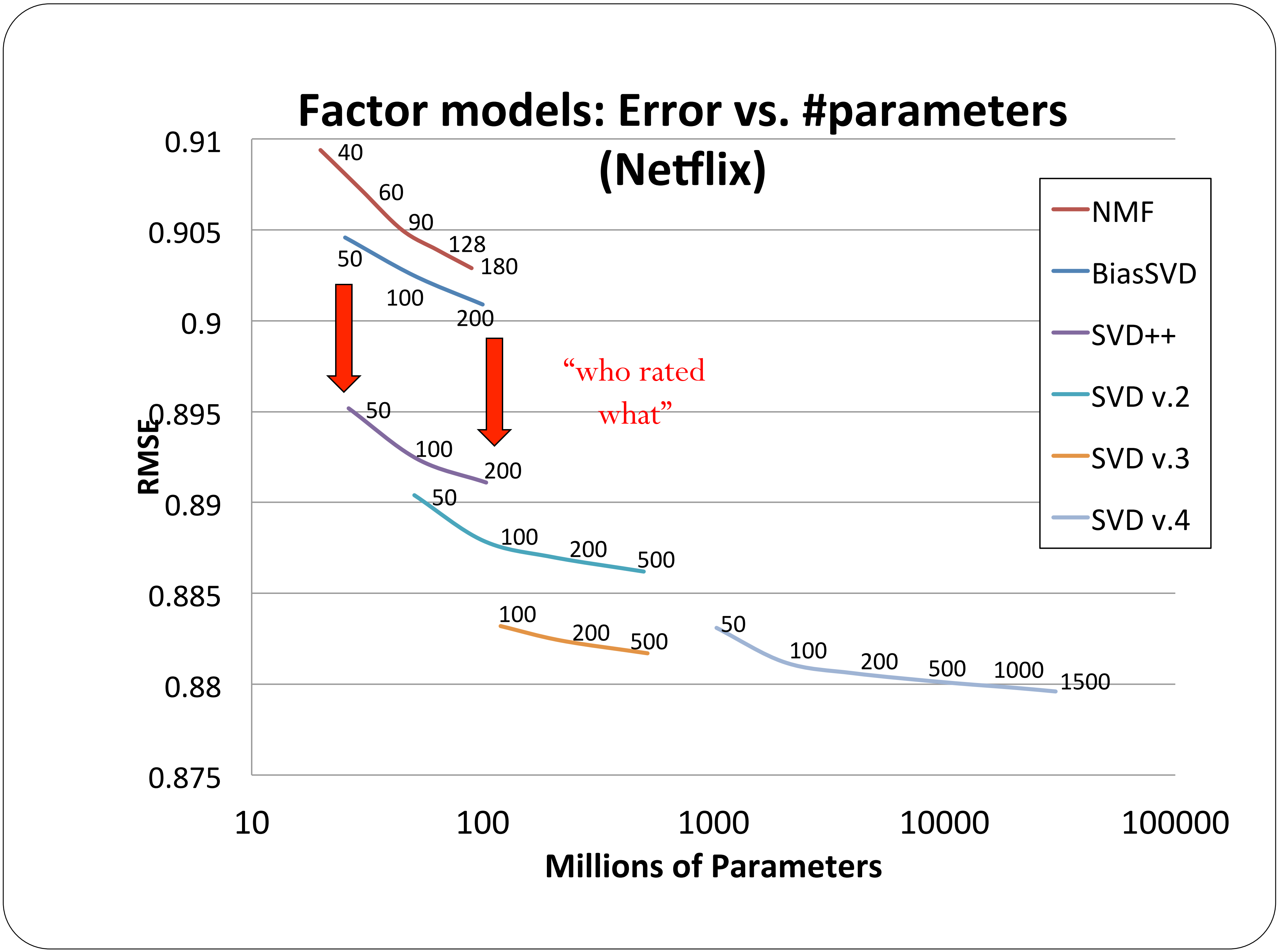
$$r_{ui} = \mu + b_u + b_i + q_i^T p_u$$

- The binary view of the user factors:

$$p_u \propto \sum_j b_{uj} x_j = \sum_{j \text{ rated by } u} x_j$$

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

"Factorization Meets the Neighborhood…", KDD'08
R. Salakhutdinov and A. Mnih, **"Probabilistic Matrix Factorization"**, NIPS'07

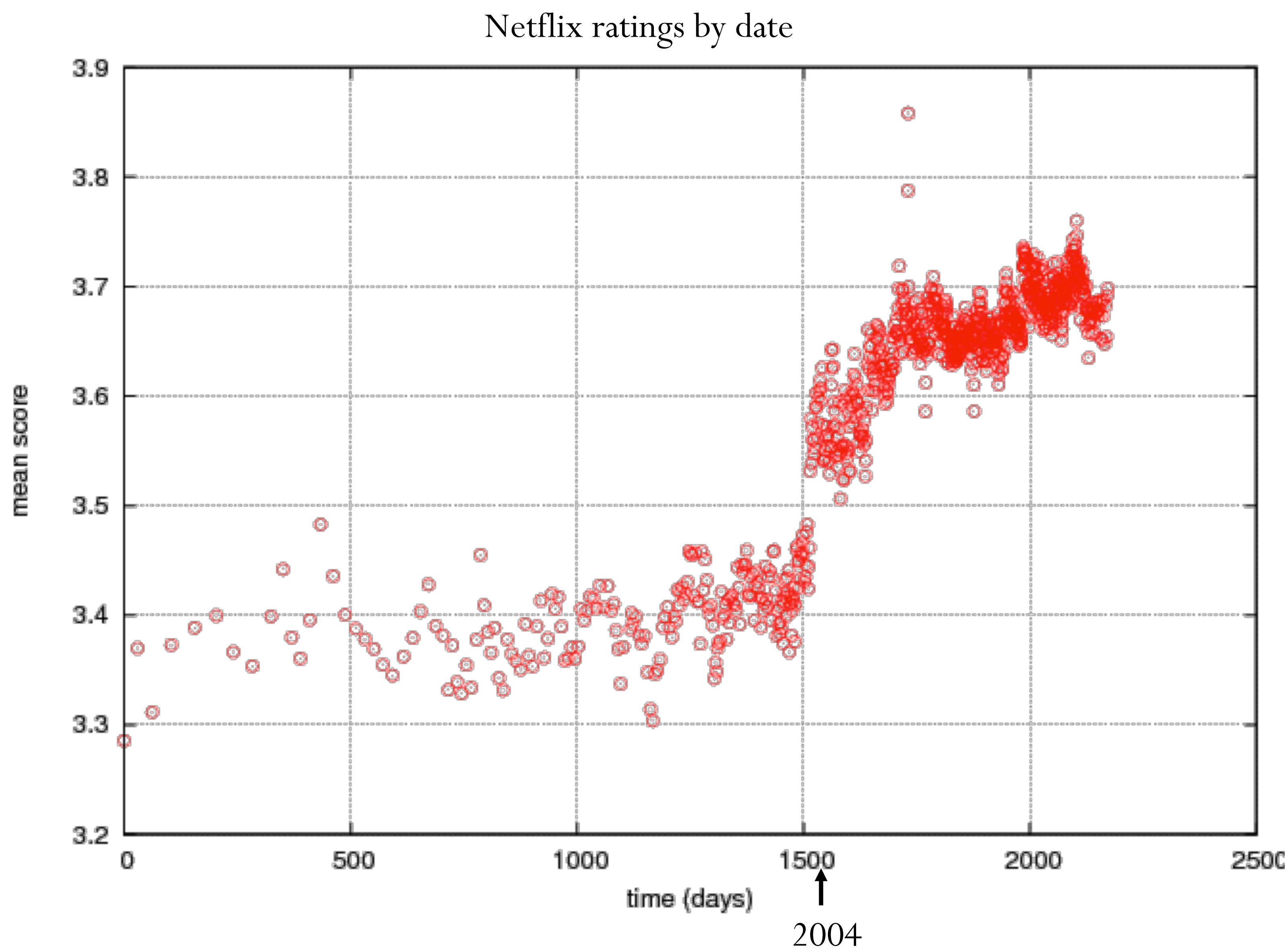


Factor models: Error vs. #parameters (Netflix)
RMSE
0.91
0.905
0.9
0.895
0.89
0.885
0.88
0.875
10
100
1000
10000
100000
Millions of Parameters
NMF
BiasSVD
SVD++
SVD v.2
SVD v.3
SVD v.4
40
60
90
128
180
50
100
200
50
100
200
50
100
200
500
100
200
500
50
100
200
500
1000
1500
“who rated what”

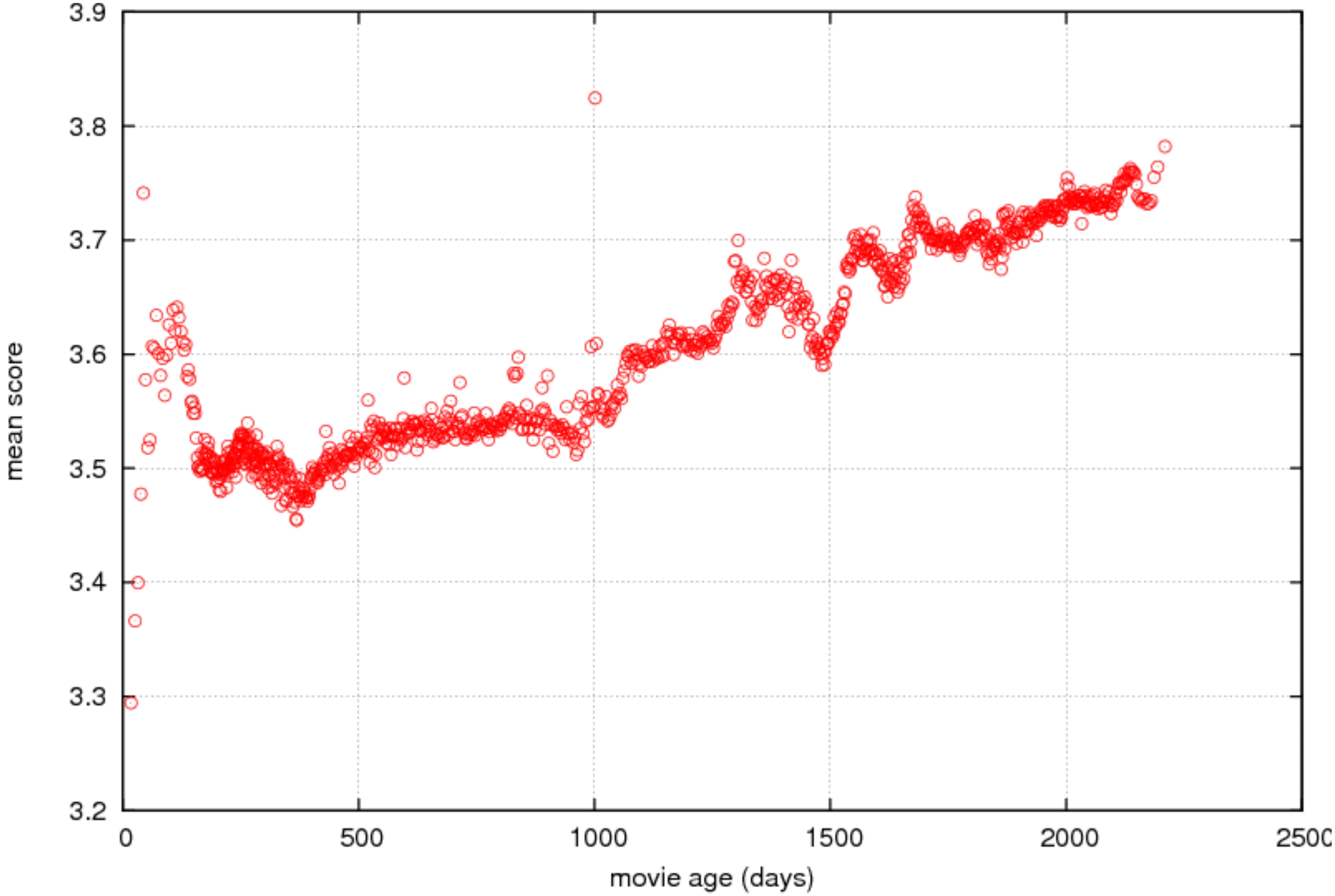
Temporal dynamics

Panta rhei

Something Happened in Early 2004…



Are movies getting better with time?



Multiple sources of temporal dynamics

- Item-side effects:
  - Product perception and popularity are constantly changing
  - Seasonal patterns influence items' popularity
- User-side effects:
  - Customers redefine their taste
  - Transient, short-term bias; anchoring
  - Drifting rating scale
  - Change of rater within household

Temporal dynamics - challenges

- 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

→ Common "concept drift" methodologies won't hold.
E.g., **underweighting older instances is unappealing**

Addressing temporal dynamics

- Factor model conveniently allows separately treating different aspects
- We observe changes in:
  1. Rating scale of individual users
  2. Popularity of individual items
  3. User preferences

Baseline predictors

User factors

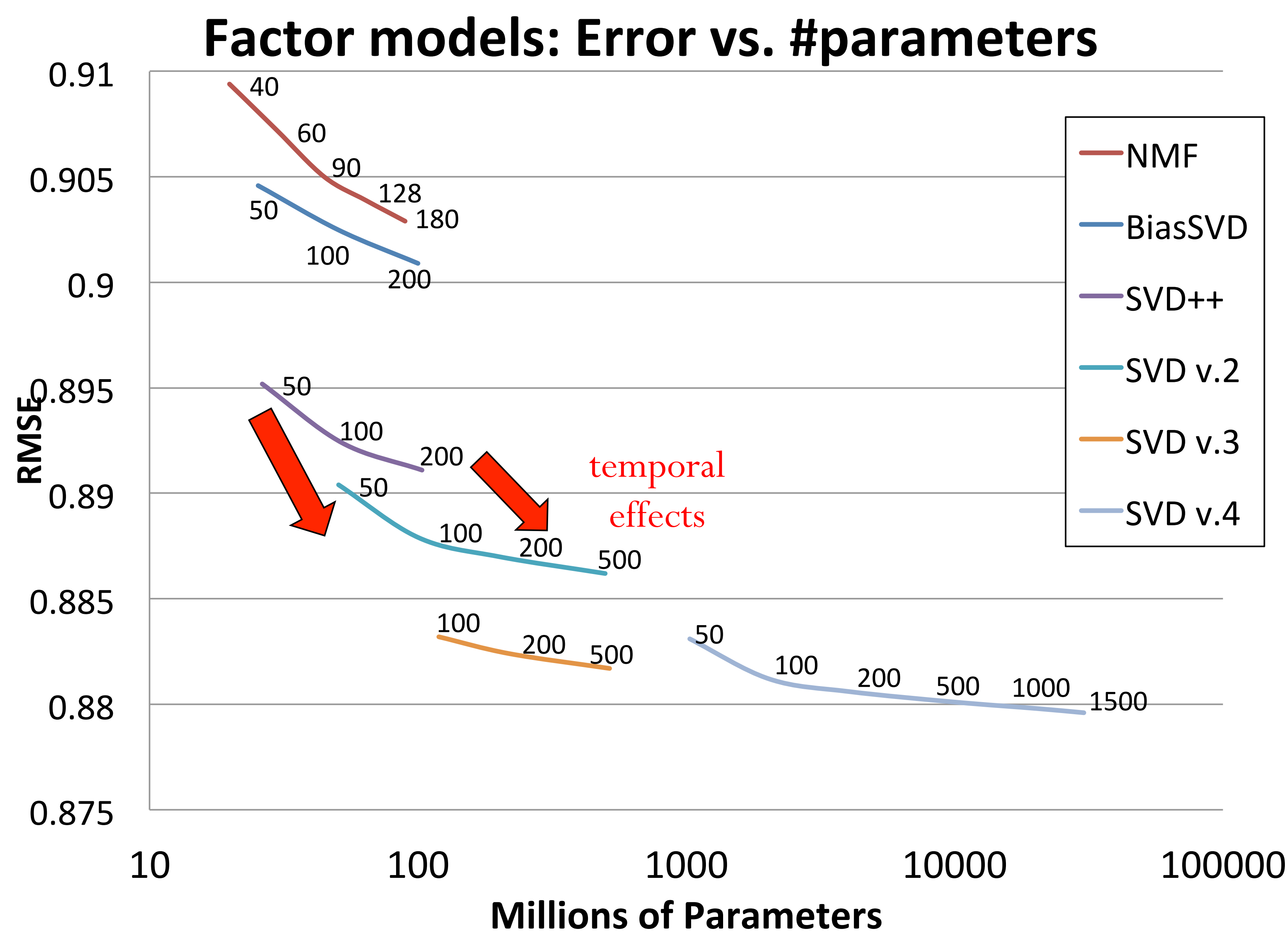
$$r_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^T p_u(t)$$

Parameterizing the model

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

- Use functional forms: $b_u(t)=f(u,t)$, $b_i(t)=g(i,t)$, $p_u(t)=h(u,t)$
- Need to find adequate f(), g(), h()
- General guidelines:
  - Items show slower temporal changes
  - Users exhibit frequent and sudden changes
  - Factors –$p_u(t)$– are expensive to model
  - Gain flexibility by heavily parameterizing the functions

"Collaborative Filtering with Temporal Dynamics", KDD'09



Factor models: Error vs. #parameters
0.91
0.905
0.9
0.895
0.89
0.885
0.88
0.875
RMSE
10
100
1000
10000
100000
Millions of Parameters
NMF
BiasSVD
SVD++
SVD v.2
SVD v.3
SVD v.4
40
60
90
128
180
50
100
200
50
100
200
50
100
200
500
100
200
500
50
100
200
500
1000
1500
temporal
effects