Recommendation Systems

based on slides by Alex Smola
Yahoo! Research and ANU

Significant content courtesy of Yehuda Koren

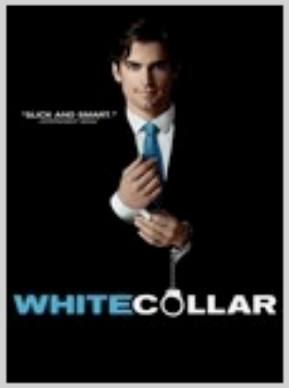
Outline

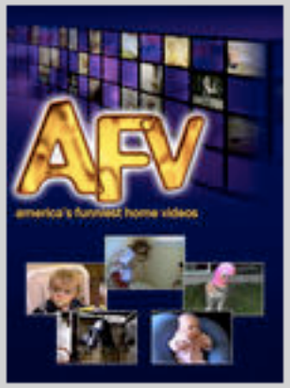
- Neighborhood methods
  - User / movie similarity
  - Iteration on graph
- Matrix Factorization
  - Singular value decomposition
    - Convex reformulation
- Improvements

Why

Thousands of movies and TV episodes including these:

New Arrivals in TV





TV Drama

TV Comedy

Children & Family

Netflix

Three Kings R

★★★★☆ (398 customer reviews)

George Clooney, Mark Wahlberg, Ice Cube conspire to steal a huge cache of gold hidden near their desert base.

Starring: George Clooney, Mark Wahlberg
Directed by: David O. Russell
Runtime: 1 hour 56 minutes
Release year: 1999
Studio: Warner Bros.

Also available in **HD** with **Amazon Instant Video on Your TV**

Prime
Your Amazon Prime membership now includes unlimited, commercial-free, instant streaming of thousands of movies and TV shows at no additional cost.

amazon instant video

Prime instant videos

Watch now unlimited streaming $0.00

48 hour rental

1-Click® $2.99

Buy movie

1-Click® $9.99

Learn more about renting and buying

Add to Wish List

[Send us Feedback]

Customers Who Bought This Item Also Bought

Page 1 of 19

Tower Heist Amazon Instant Video ~ Eddie Murphy
★★★☆☆ (42)
$3.99

Syriana Amazon Instant Video ~ George Clooney
★★★☆☆ (355)
$2.99

Five Minutes of Heaven Amazon Instant Video ~ Liam Neeson
★★★★☆ (70)
$2.99

Foolproof Amazon Instant Video ~ Ryan Reynolds
★★★★☆ (81)
$2.99

The Recruit Amazon Instant Video ~ Al Pacino
★★★½☆ (161)
$1.99

Alexander's Amazon.com › **Recommended for You**
(If you're not Alexander Smola, click here.)

Just For Today

Browse Recommended

Recommendations

Amazon Instant Video
Appstore for Android
Baby
Beauty
Books
Books on Kindle
Camera & Photo
Clothing & Accessories
Computers & Accessories
Electronics
Grocery & Gourmet Food
Health & Personal Care
Home Improvement
Industrial & Scientific
Jewelry
Kitchen & Dining
MP3 Downloads
Magazine Subscriptions
Movies & TV
Music
Musical Instruments
Office & School Supplies
Patio, Lawn & Garden
Shoes
Software
Sports & Outdoors
Toys & Games
Video Games
Watches

These recommendations are based on items you own and more.

view: **All** | New Releases | Coming Soon

More results

1.

Convex Optimization
by Stephen Boyd (March 8, 2004)
Average Customer Review: ★★★★½ (13)
In Stock

List Price: ~~$84.00~~
Price: $68.13
44 used & new from $61.32

Add to Cart | Add to Wish List

I own it | Not interested | ☆☆☆☆☆ Rate this item
Recommended because you purchased **Nonlinear Programming** and more (Fix this)

2.

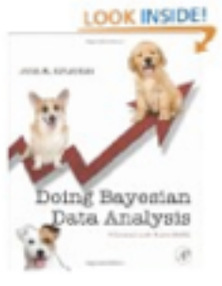
Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)
by Nir Friedman (July 31, 2009)
Average Customer Review: ★★★★☆ (11)
In Stock

List Price: ~~$95.00~~
Price: $93.55
47 used & new from $91.33

Add to Cart | Add to Wish List

I own it | Not interested | ☆☆☆☆☆ Rate this item
Recommended because you purchased **Nonlinear Programming** and more (Fix this)

3.



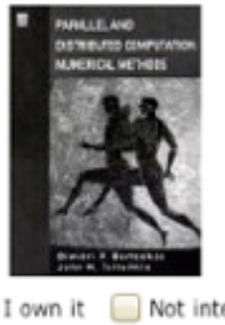
Doing Bayesian Data Analysis: A Tutorial with R and BUGS
by John K. Kruschke (November 10, 2010)
Average Customer Review: ★★★★★ (15)
In Stock

List Price: ~~$89.95~~
Price: $77.98
44 used & new from $68.40

Add to Cart | Add to Wish List

I own it | Not interested | ☆☆☆☆☆ Rate this item
Recommended because you purchased **Bayesian Nonparametrics** and more (Fix this)

4.



Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)
by Dimitri P. Bertsekas (January 1, 1997)
Average Customer Review: ★★★★★ (1)
In Stock

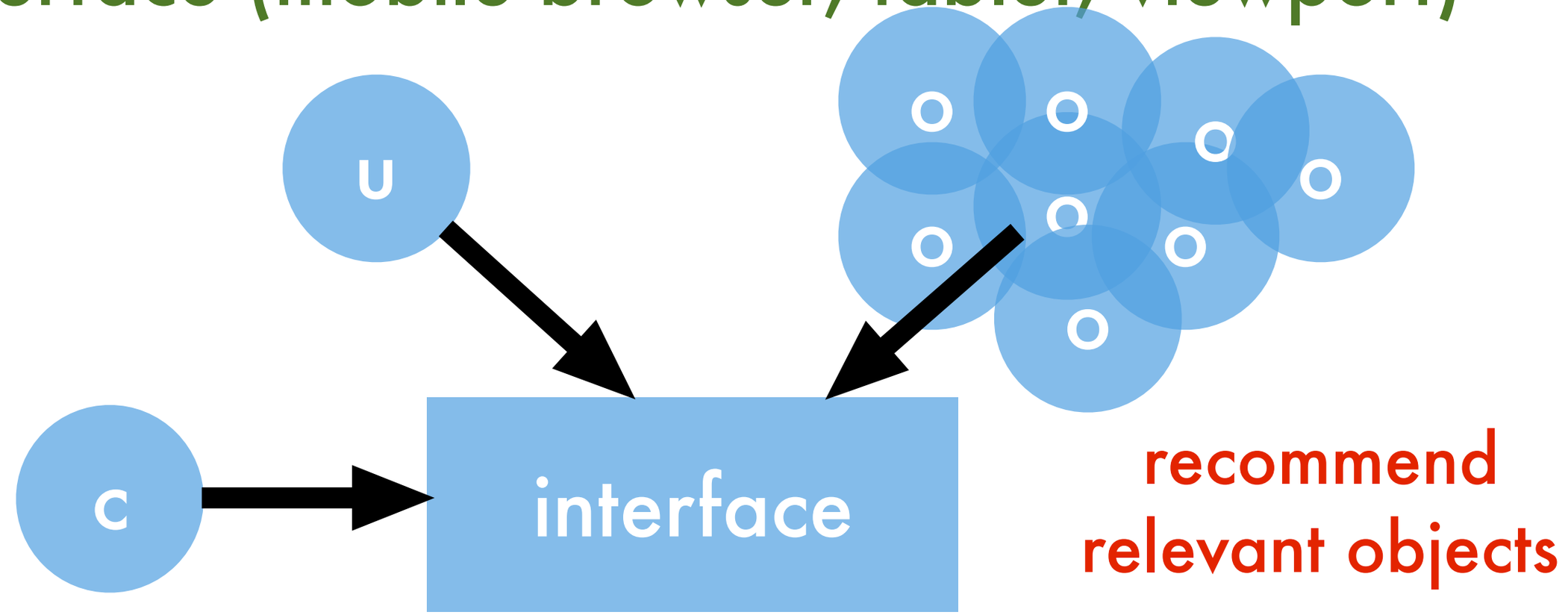
Price: $49.50
15 used & new from $45.49

Add to Cart | Add to Wish List

I own it | Not interested | ☆☆☆☆☆ Rate this item
Recommended because you purchased **Nonlinear Programming** (Fix this)

A more formal view

- User (requests content)
- Objects (that can be displayed)
- Context (device, location, time)
- Interface (mobile browser, tablet, viewport)



Examples

- Movie recommendation (Netflix)
- Related product recommendation (Amazon)
- Web page ranking (Google)
- Social recommendation (Facebook)
- News content recommendation (Yahoo)
- Priority inbox & spam filtering (Google)
- Online dating (OK Cupid)
- Computational Advertising (Yahoo)

Running Example
Netflix Movie Recommendation

Training data

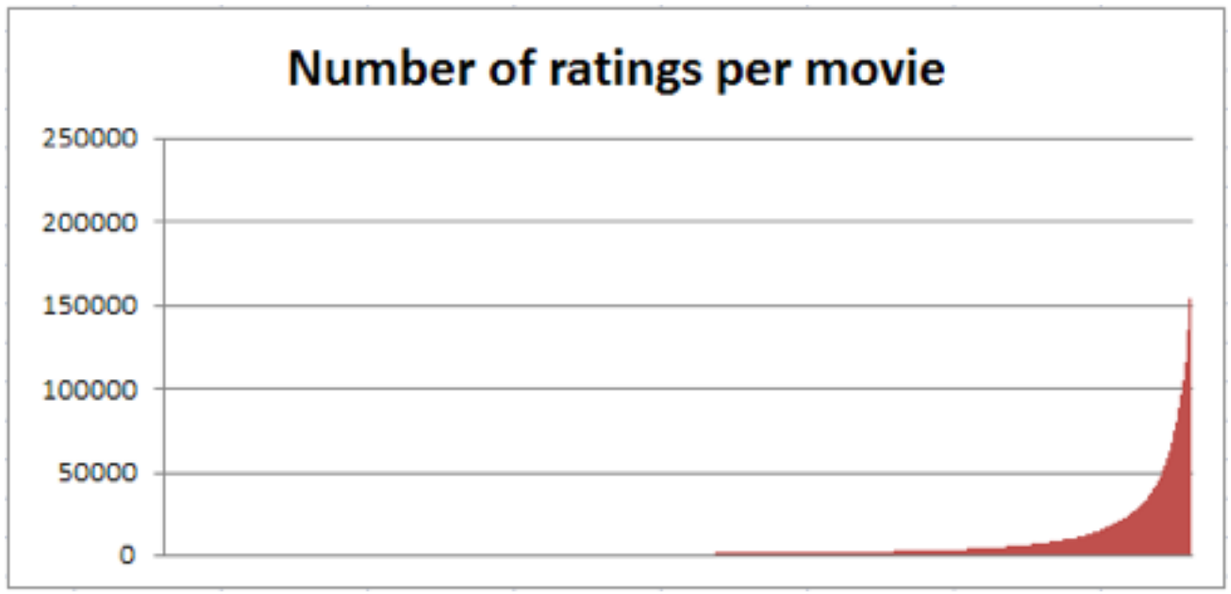
user	movie	date	score
1	21	5/7/02	1
1	213	8/2/04	5
2	345	3/6/01	4
2	123	5/1/05	4
2	768	7/15/02	3
3	76	1/22/01	5
4	45	8/3/00	4
5	568	9/10/05	1
5	342	3/5/03	2
5	234	12/28/00	2
6	76	8/11/02	5
6	56	6/15/03	4

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	?

Challenges

- Scalability
  - Millions of objects
  - 100s of millions of users
- Cold start
  - Changing user base
  - Changing inventory (movies, stories, goods)
  - Attributes
- Imbalanced dataset
  User activity / item reviews are power law distributed



http://www.igvita.com/2006/10/29/dissecting-the-netflix-dataset/

Netflix competition yardstick

- Least mean squares prediction error
  - Easy to define

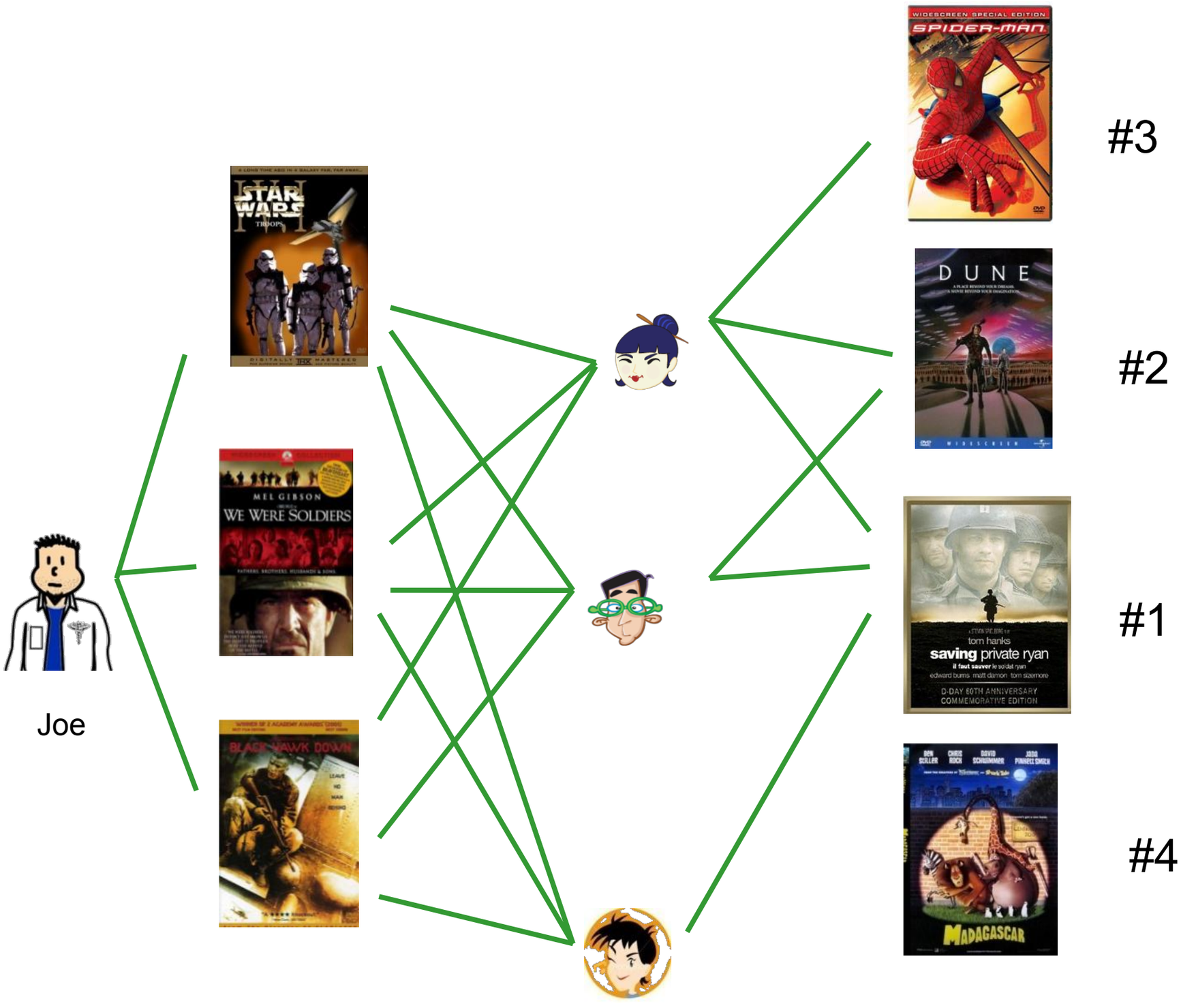
$$\mathrm{rmse}(S) = \sqrt{|S|^{-1} \sum_{(i,u)\in S} (\hat{r}_{ui} - r_{ui})^2}$$

- Wrong measure for composing sessions!

- Consistent (in large sample size limit this will converge to minimizer)

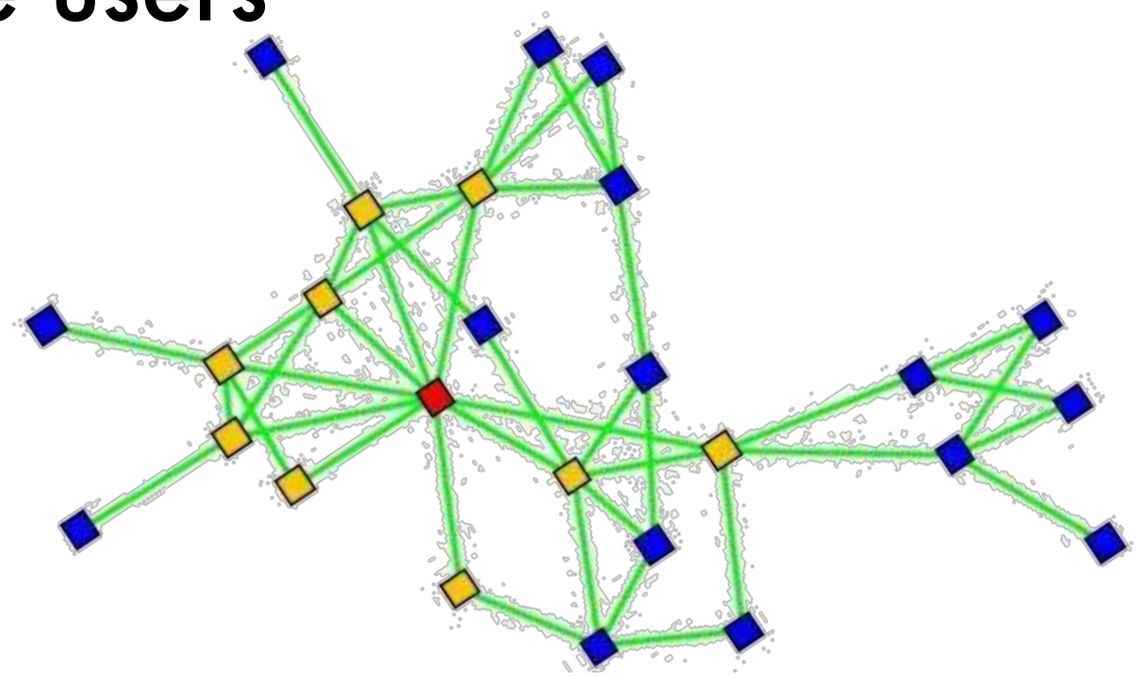
1 Neighborhood Methods

Basic Idea

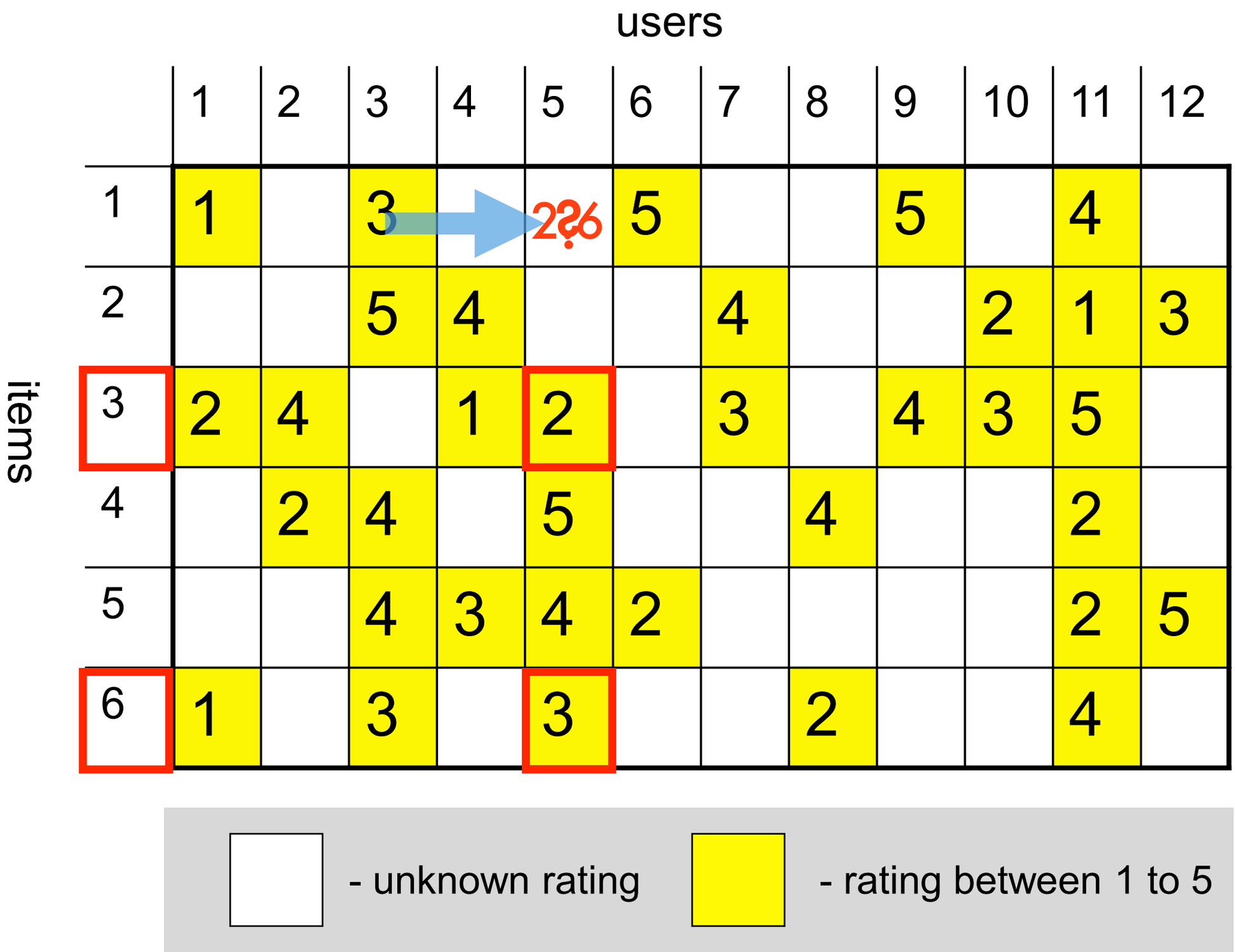

STAR WARS
TROOPS
WE WERE SOLDIERS
MEL GIBSON
BLACK HAWK DOWN
LEAVE NO MAN BEHIND
SPIDER-MAN
WIDESCREEN SPECIAL EDITION
DUNE
tom hanks
saving private ryan
il faut sauver le soldat ryan
edward burns matt damon tom sizemore
D-DAY 60TH ANNIVERSARY
COMMEMORATIVE EDITION
MADAGASCAR
Joe
#3
#2
#1
#4

Basic Idea

- (user,user) similarity to recommend items
  - good if item base is smaller than user base
  - good if item base changes rapidly
  - traverse bipartite similarity graph
- (item,item) similarity to recommend new items that were also liked by the same users
  - good if the user base is small is small
- Oldest known CF method



Neighborhood based CF



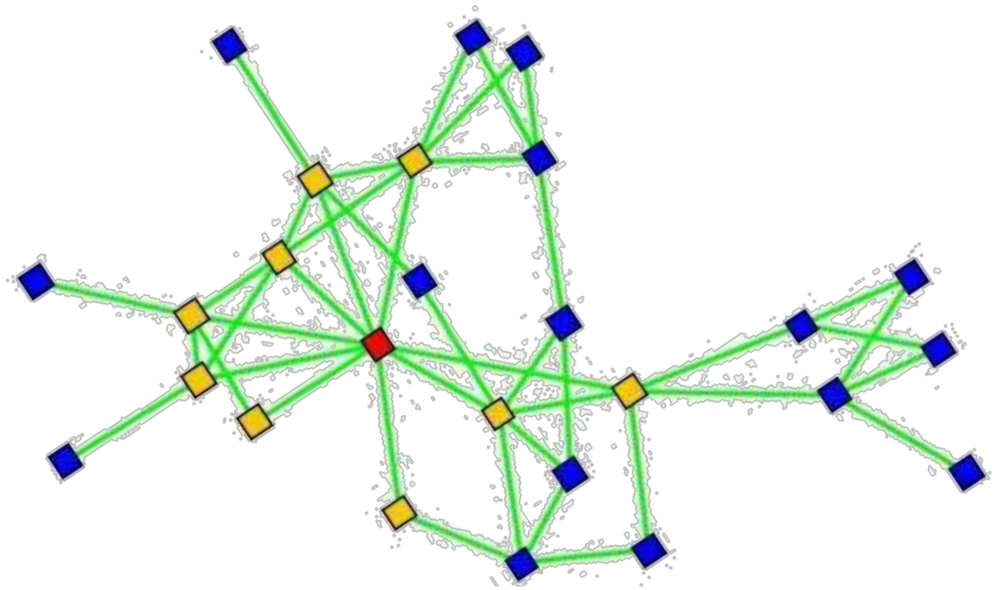
similarity

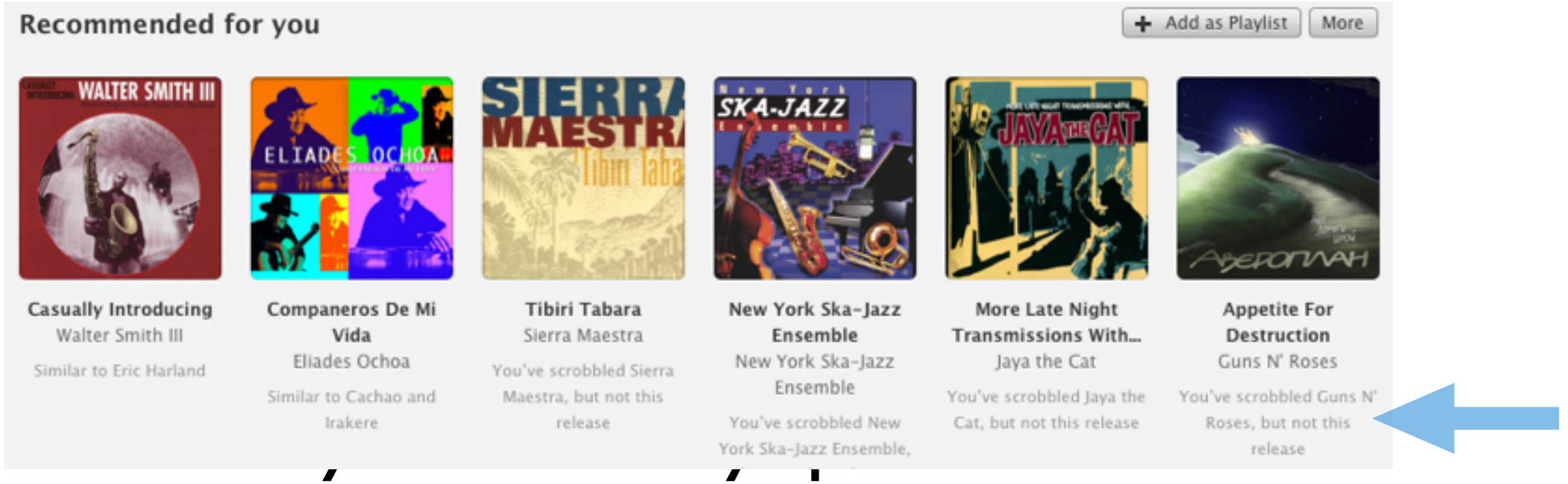
$s_{13} = 0.2$

$s_{16} = 0.3$

Properties

- Intuitive
- No (substantial) training
- Handles new users / items
- Easy to explain to user





Normalization / Bias

- Problem
  - Some items are significantly higher rated
  - Some users rate substantially lower
  - Ratings change over time
- Bias correction is crucial for nearest neighborhood recommender algorithm
  - Offset per user
  - Offset per movie
  - Time effects
  - Global bias

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

globa

user

item

Bell & Koren ICDM 2007

http://public.research.att.com/~volinsky/netflix/BellKorICDM07.pdf

Baseline estimation

- Mean rating is 3.7
- Troll Hunter is 0.7 above mean
- User rates 0.2 below mean
- Baseline is 4.2 stars
- Least mean squares problem

$$\underset{b}{\text{minimize}} \sum_{(u,i)} (r_{ui} - \mu - b_u - b_i)^2 + \lambda \left[\sum_u b_u^2 + \sum_i b_i^2 \right]$$

- Jointly convex. Alternatively remove mean & iterate

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

Parzen Windows style CF

- Similarity measure s_{ij} between items
- Find set $s_k(i,u)$ of k-nearest neighbors to i that were rated by user u
- Weighted average over the set

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in s_k(i,u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in s_k(i,u)} s_{ij}} \text{ where } b_{ui} = \mu + b_u + b_i$$

- How to compute s_{ij}?

(item,item) similarity measures

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

- Pearson correlation coefficient
  - nonuniform support $s_{ij} = \frac{\mathrm{Cov}[r_{ui}, r_{uj}]}{\mathrm{Std}[r_{ui}]\mathrm{Std}[r_{uj}]}$
  - compute only over shared support
  - shrinkage towards 0 to address problem of small support (typically few items in

(item,item) similarity

- Empirical Pearson correlation coefficient

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

- Smoothing towards 0 for small support

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

- Here, |U(i,j)| users that rated both i and j
- Shrink towards baseline for small neighborhood

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in s_k(i,u)} s_{ij}(r_{uj} - b_{uj})}{\lambda + \sum_{j \in s_k(i,u)} s_{ij}}$$

Similarity for binary data

- Pearson correlation meaningless
  - Views
  - Purchase behavior
  - Clicks
- Jaccard similarity (intersection vs. joint)
- Observed/expected ratio
  Improve by counting per user (many users better than heavy users)

m_i users acting on i

m_{ij} users acting on both i and j

m total number of users

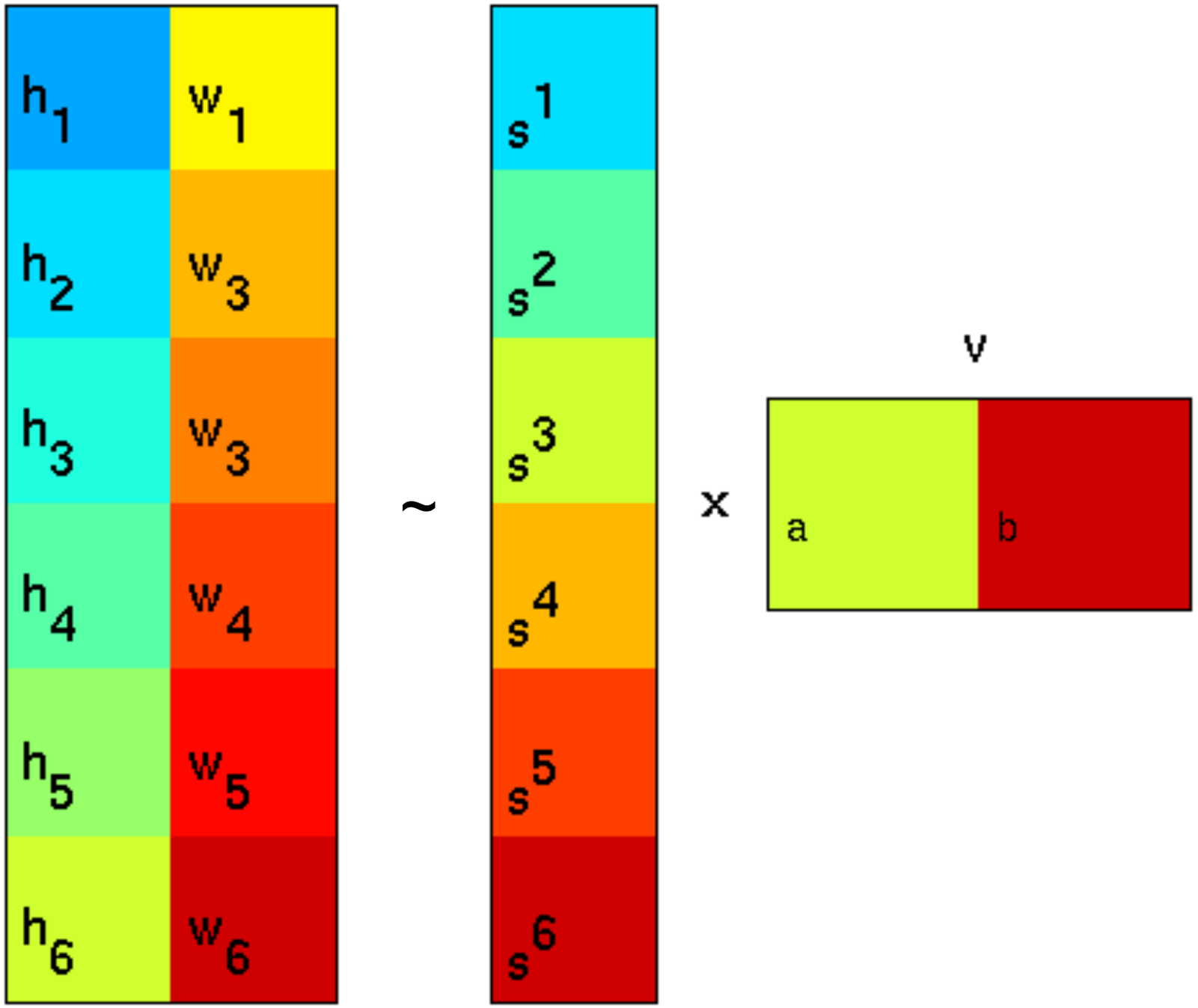
$$s_{ij} = \frac{m_{ij}}{\alpha + m_i + m_j - m_{ij}}$$

$$s_{ij} = \frac{\text{observed}}{\text{expected}} \approx \frac{m_{ij}}{\alpha + m_i m_j / m}$$

2 Matrix Factorization

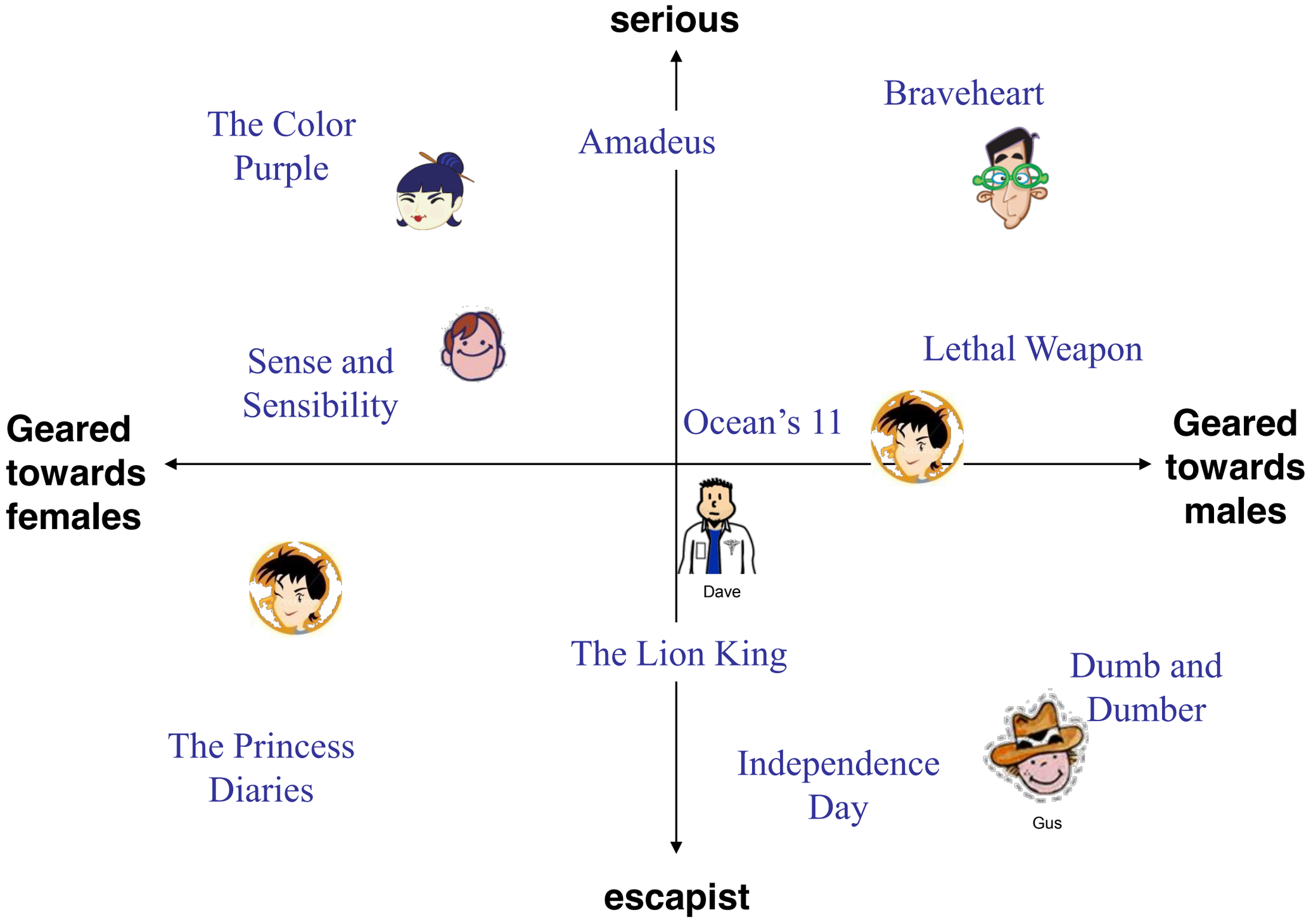
Basics

Basic Idea



$$M \approx U \cdot V$$

Latent variable view



Basic matrix factorization

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

A rank-3 SVD approximation

Estimate unknown ratings as inner products of latent 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 latent 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 latent 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	

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

Properties

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	

~

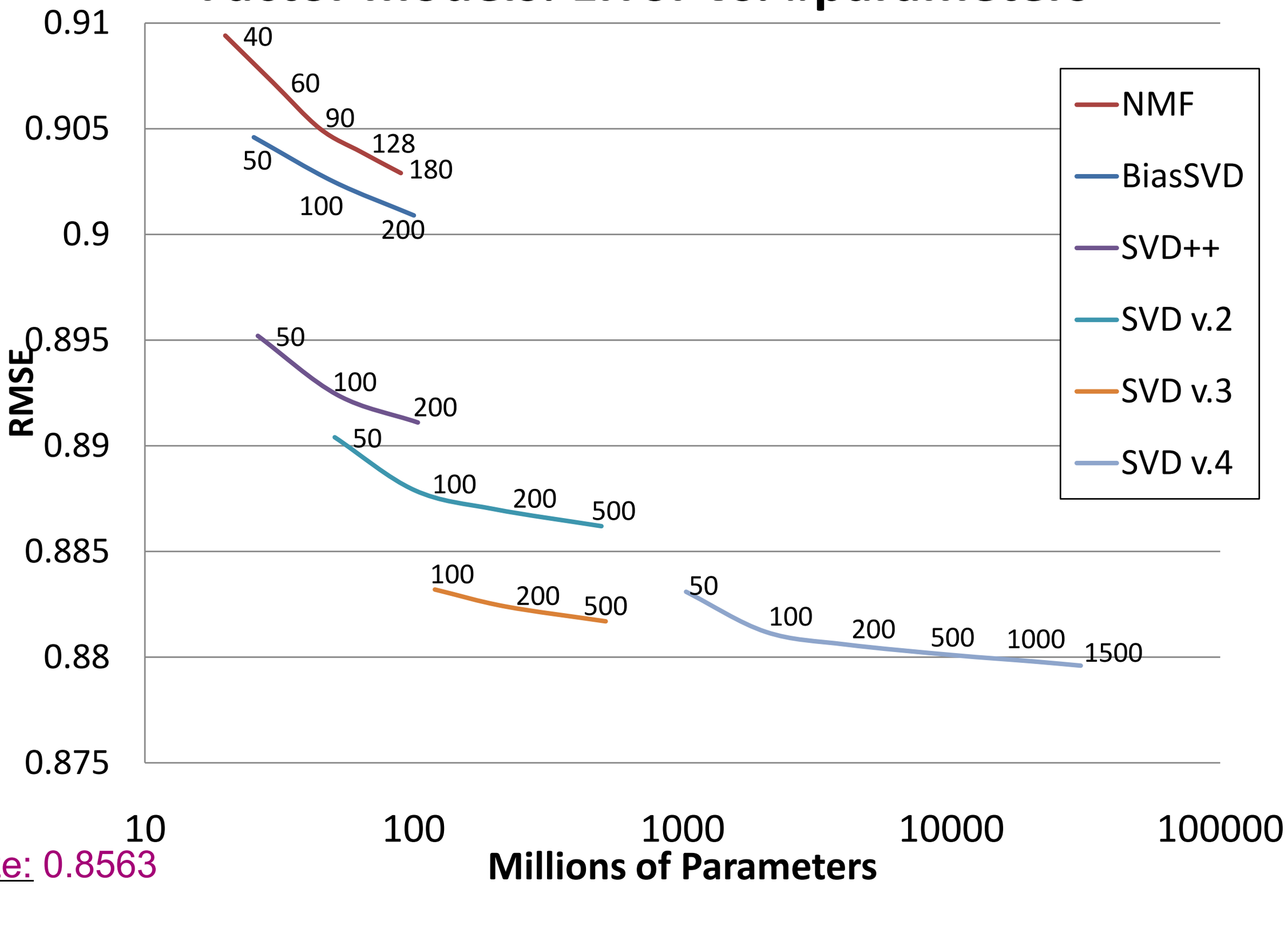
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

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

- SVD is undefined for missing entries
  - stochastic gradient descent (faster)
  - alternating optimization
- Overfitting without regularization
  particularly if fewer reviews than dimensions
- Very popular on Netflix

Netflix: 0.9514

Factor models: Error vs. #parameters



Prize: 0.8563

Risk Minimization View

- Objective Function

$$\underset{p,q}{\text{minimize}} \sum_{(u,i)\in S} (r_{ui} - \langle p_u, q_i \rangle)^2 + \lambda \left[\|p\|^2_{\text{Frob}} + \|q\|^2_{\text{Frob}} \right]$$

- Alternating least squares

$$p_u \leftarrow \left[\lambda \mathbf{1} + \sum_{i|(u,i)\in S} q_i q_i^\top \right]^{-1} \sum_i q_i r_{ui}$$

$$q_i \leftarrow \left[\lambda \mathbf{1} + \sum_{u|(u,i)\in S} p_u p_u^\top \right]^{-1} \sum_i p_u r_{ui}$$

good for MapReduce

Risk Minimization View

- Objective Function

$$\underset{p,q}{\text{minimize}} \sum_{(u,i)\in S} (r_{ui} - \langle p_u, q_i \rangle)^2 + \lambda \left[\|p\|^2_{\text{Frob}} + \|q\|^2_{\text{Frob}} \right]$$

- Stochastic gradient descent

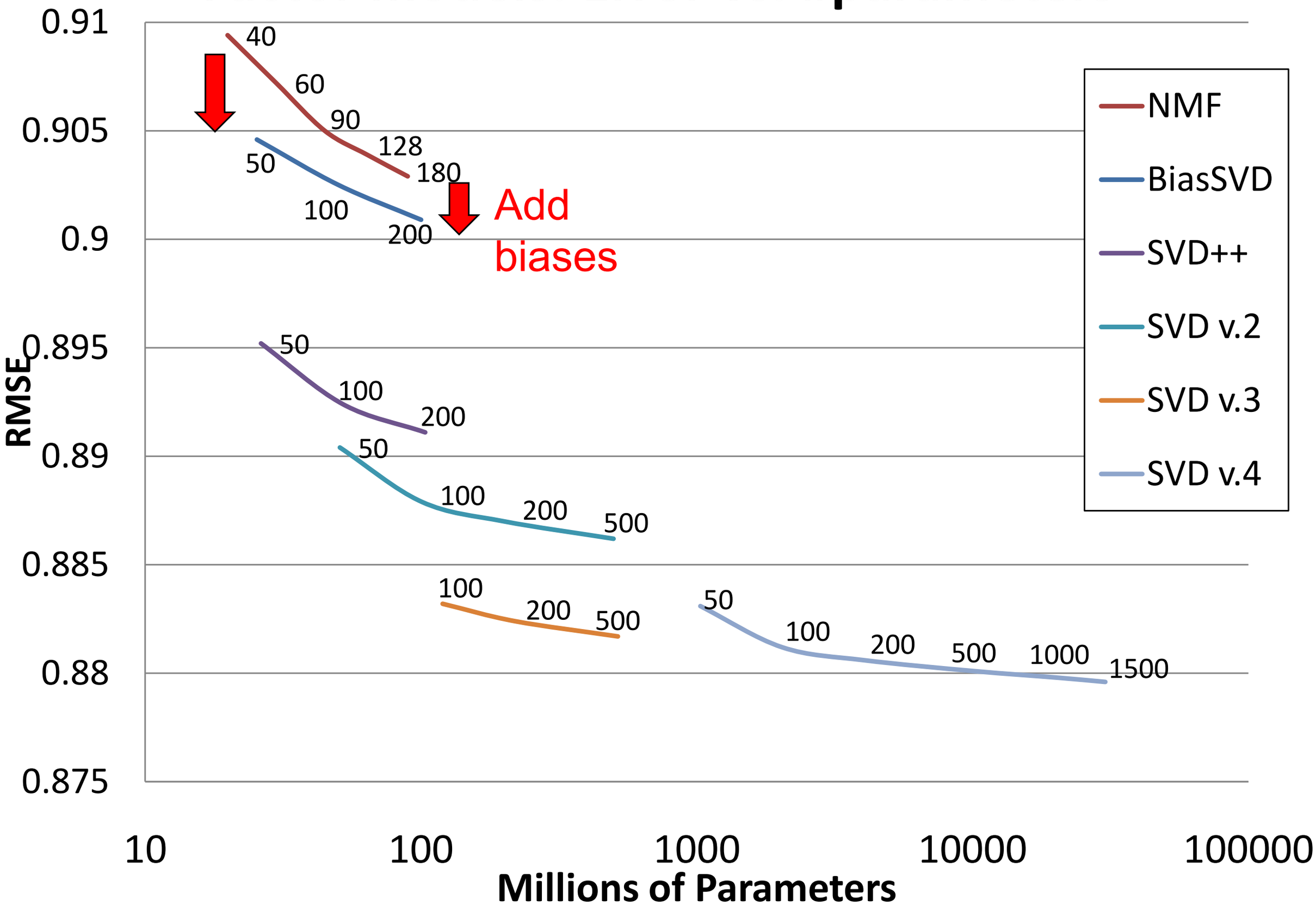
$$p_u \leftarrow (1 - \lambda\eta_t)p_u - \eta_t q_i (r_{ui} - \langle p_u, q_i \rangle)$$
$$q_i \leftarrow (1 - \lambda\eta_t)q_i - \eta_t p_u (r_{ui} - \langle p_u, q_i \rangle)$$

much faster

  - No need for locking
  - Multicore updates asynchronously (Recht, Re, Wright, 2012 - Hogwild)

Improvements

Factor models: Error vs. #parameters



Bias

- Objective Function

$$\underset{p,q}{\text{minimize}} \sum_{(u,i)\in S} (r_{ui} - (\mu + b_u + b_i + \langle p_u, q_i \rangle))^2 +$$

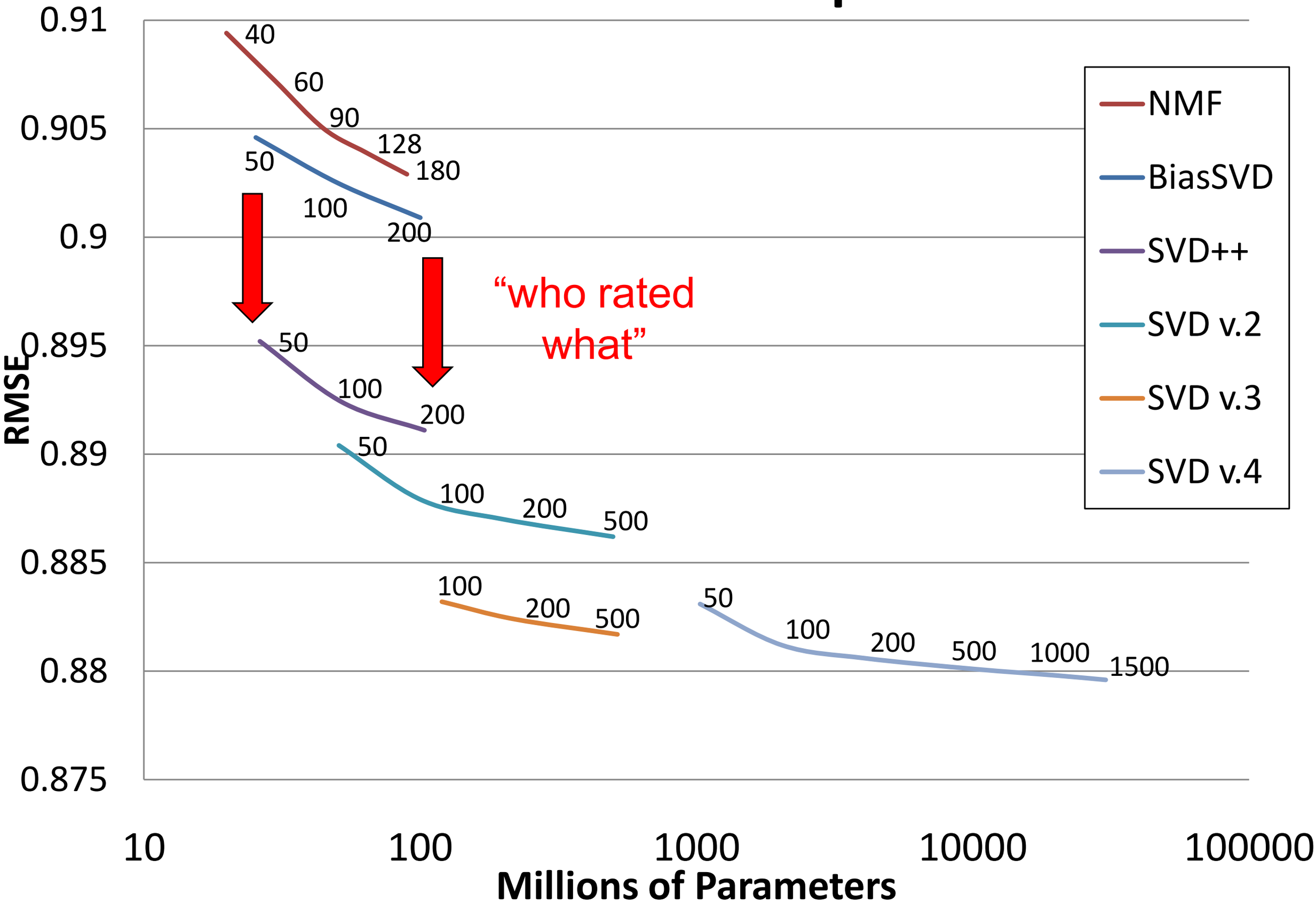
$$\lambda \left[\|p\|^2_{\text{Frob}} + \|q\|^2_{\text{Frob}} + \|b_{\text{users}}\|^2 + \|b_{\text{items}}\|^2 \right]$$

- Stochastic gradient descent

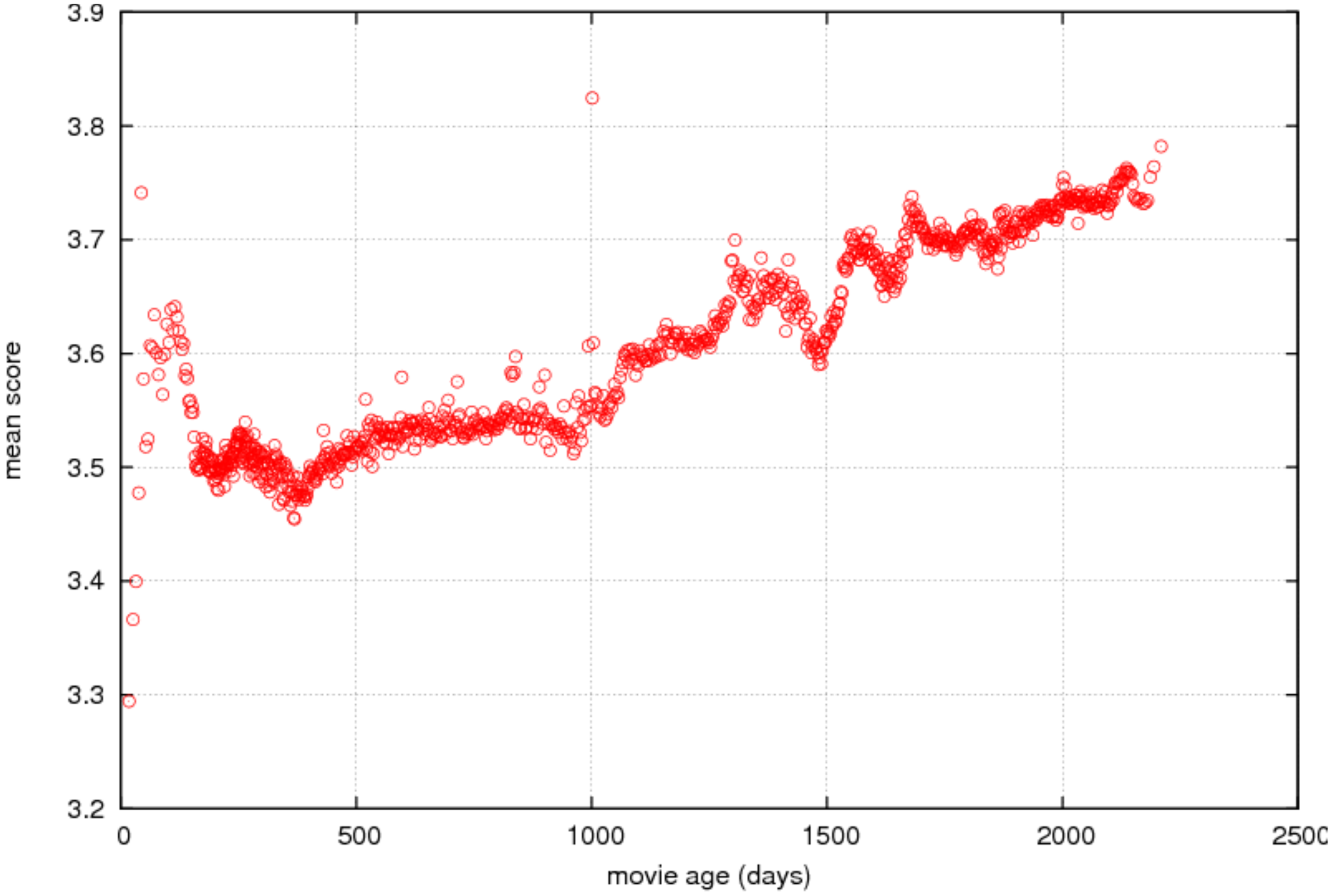
$$p_u \leftarrow (1 - \lambda\eta_t)p_u - \eta_t q_i \rho_{ui}$$
$$q_i \leftarrow (1 - \lambda\eta_t)q_i - \eta_t p_u \rho_{ui}$$
$$b_u \leftarrow (1 - \lambda\eta_t)b_u - \eta_t \rho_{ui}$$
$$b_i \leftarrow (1 - \lambda\eta_t)b_i - \eta_t \rho_{ui}$$
$$\mu \leftarrow (1 - \lambda\eta_t)\mu - \eta_t \rho_{ui}$$

$$\text{where } \rho_{ui} = (r_{ui} - (\mu + b_i + b_u + \langle p_u, q_i \rangle))$$

Factor models: Error vs. #parameters



Are movies getting better with time?



Sources of temporal change

- Items
  - Seasonal effects
    (Christmas, Valentine's day, Holiday movies)
  - Public perception of movies (Oscar etc.)
- Users
  - Changed labeling of reviews
  - Anchoring (relative to previous movie)
  - Change of rater in household
  - Selection bias for time of viewing

Modeling temporal change

- Time-dependent bias
- Time-dependent user preferences

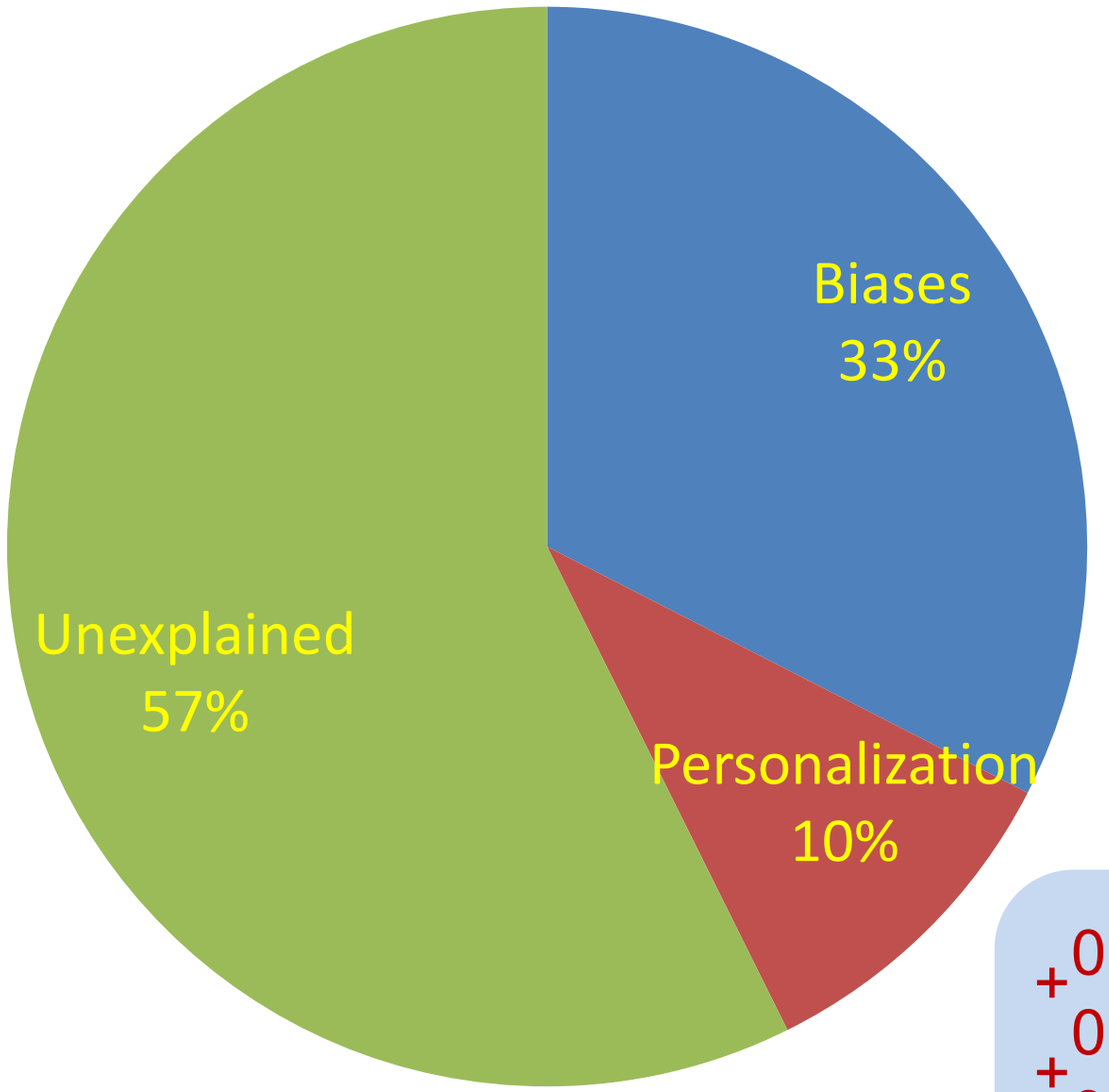
$$r_{ui}(t) = \mu + b_u(t) + b_i(t) + \langle q_i, p_u(t) \rangle$$

- Parameterize functions b and p
  - Slow changes for items
  - Fast sudden changes for users
  - Good parametrization is key

Koren et al., KDD 2009 (CF with temporal dynamics)

Bias matters

Sources of Variance in Netflix data



+0.732 (unexplained)
+0.415 (biases)
+0.129 (personalization)

1.276 (total variance)

Summary

- Neighborhood methods
  - User / movie similarity
  - Iteration on graph
- Matrix Factorization
  - Singular value decomposition
    - Convex reformulation
- Improvements

Further reading

- Neighborhood factorization http://research.yahoo.com/files/paper.pdf
- Matrix Factorization for recommender systems https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf
- Yehuda Koren's papers http://research.yahoo.com/Yehuda_Koren