

Data Mining Techniques

CS 6220 - Section 3 - Fall 2016

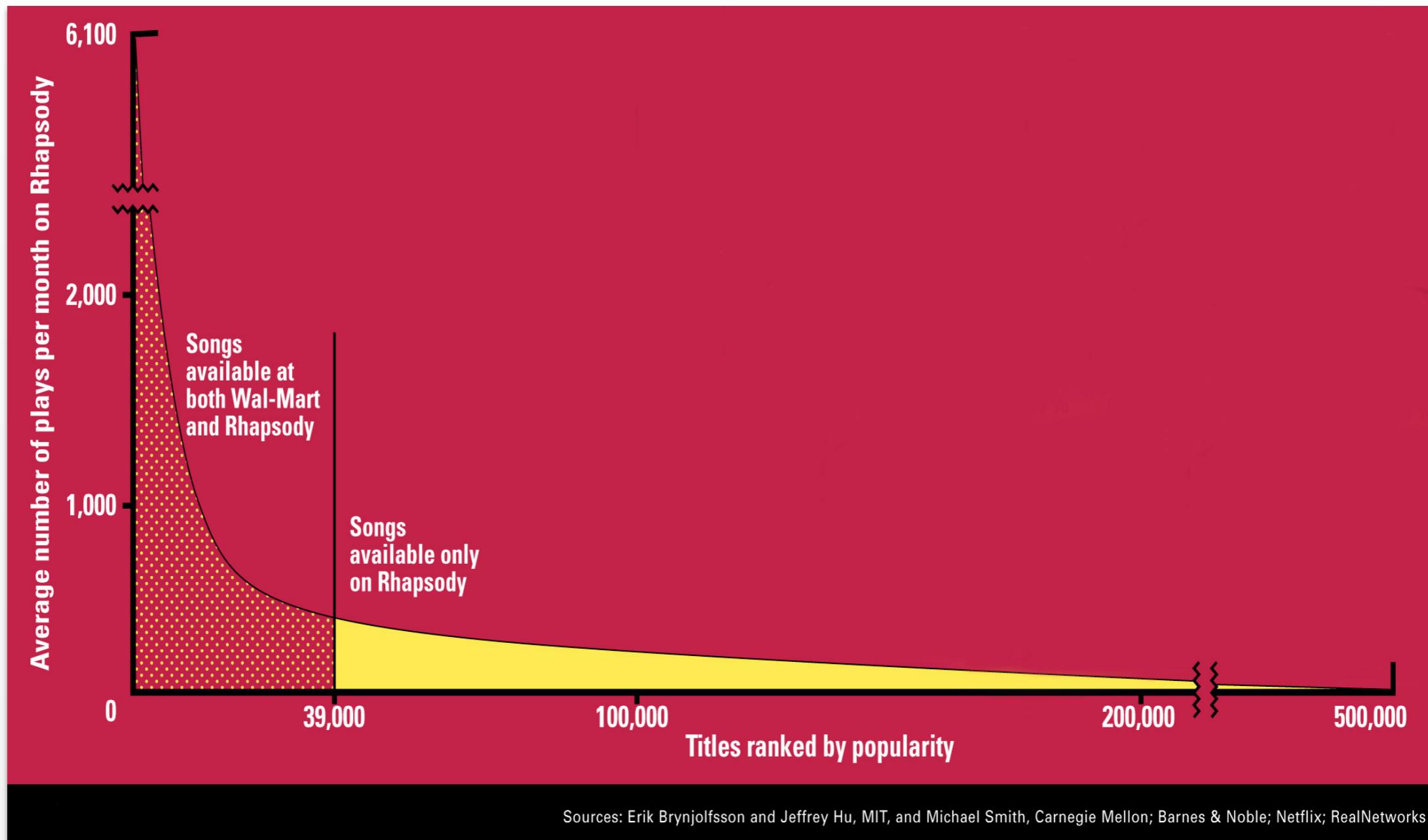
Lecture 14

Jan-Willem van de Meent
(credit: Andrew Ng, Alex Smola,
Yehuda Koren, Stanford CS246)



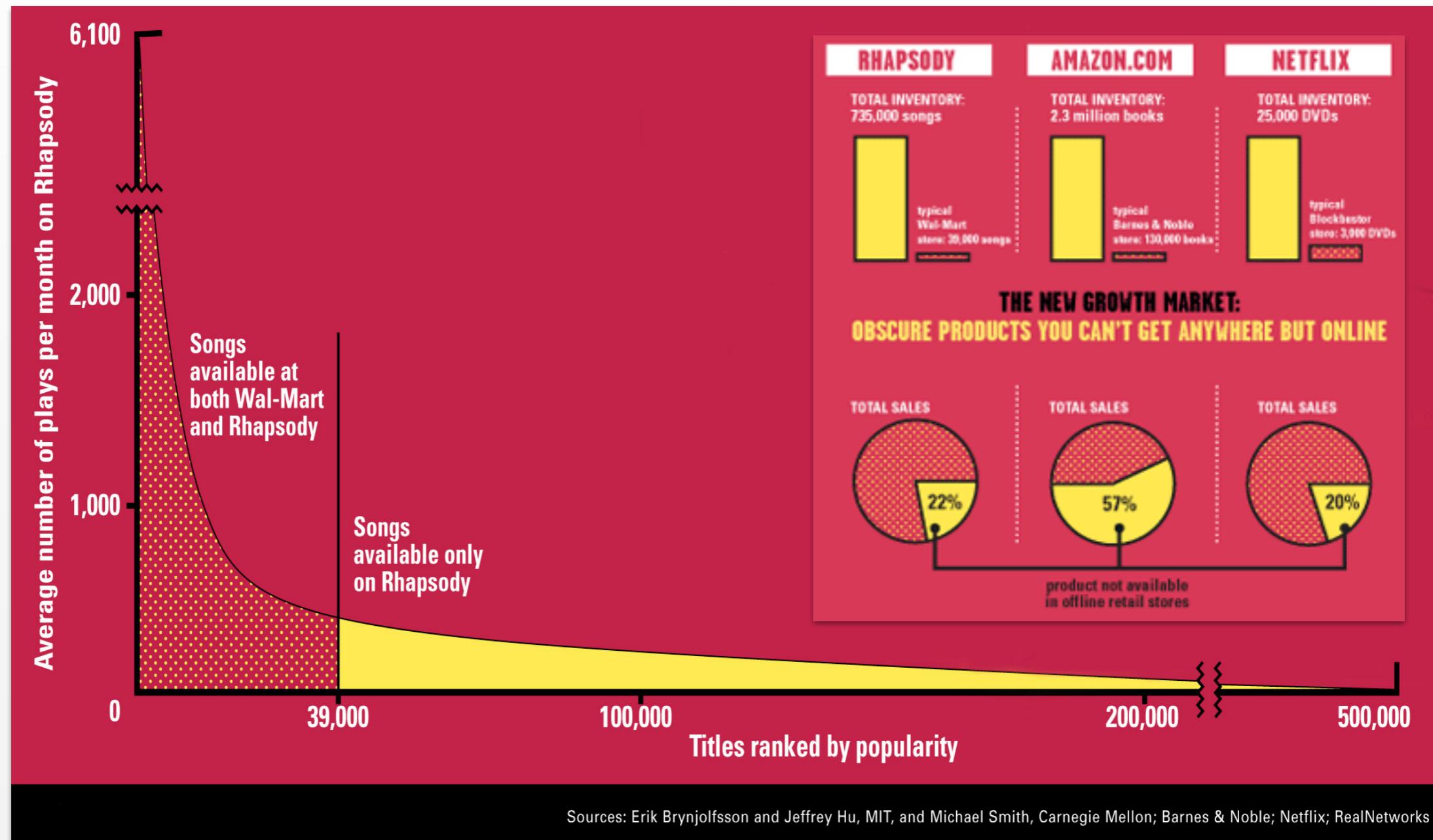
Recommender Systems

The Long Tail



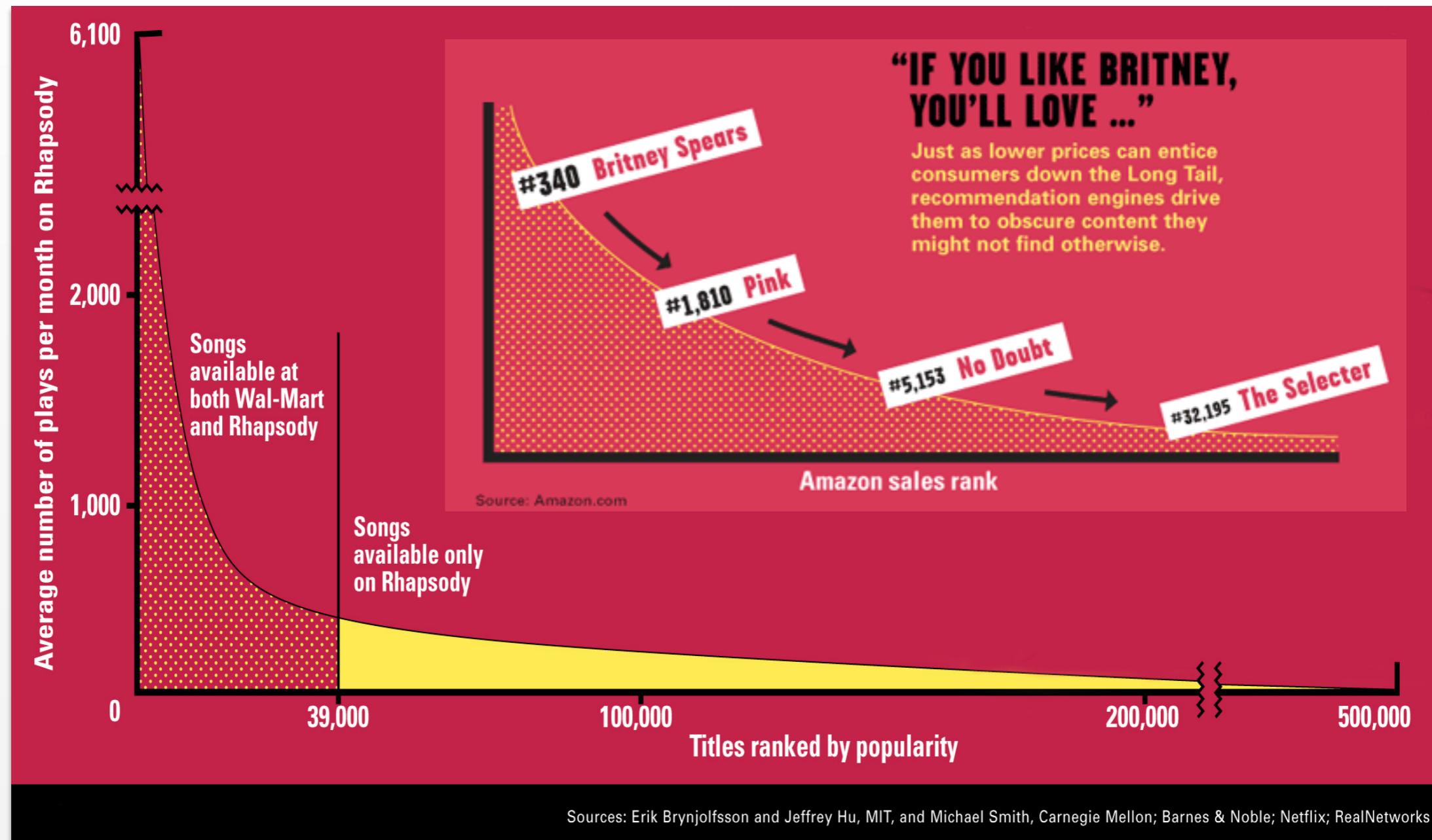
(from: <https://www.wired.com/2004/10/tail/>)

The Long Tail



(from: <https://www.wired.com/2004/10/tail/>)

The Long Tail



(from: <https://www.wired.com/2004/10/tail/>)

Problem Setting

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

Problem Setting

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

Problem Setting

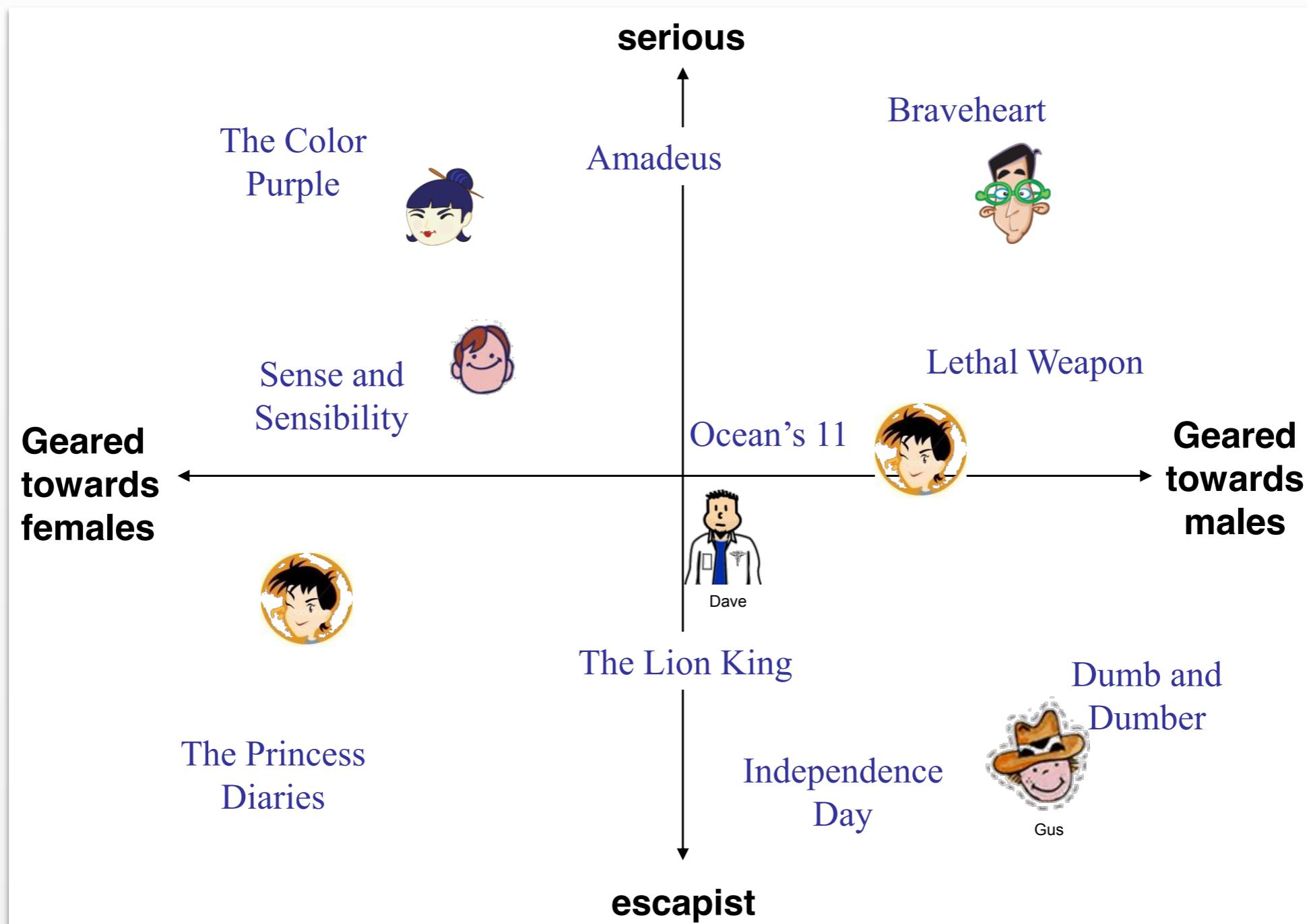
Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

Problem Setting

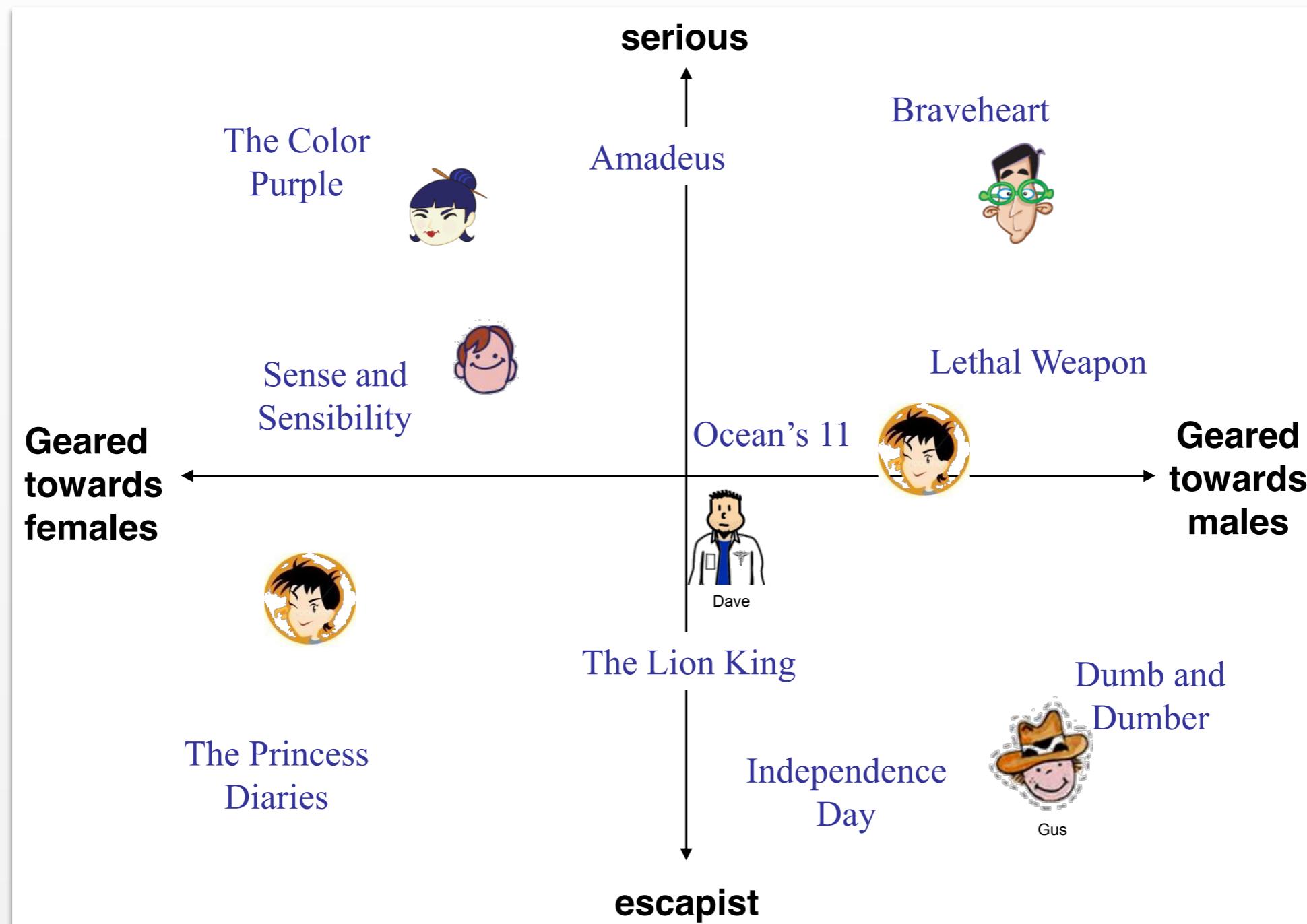
Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

- Task: Predict user preferences for unseen items

Content-based Filtering

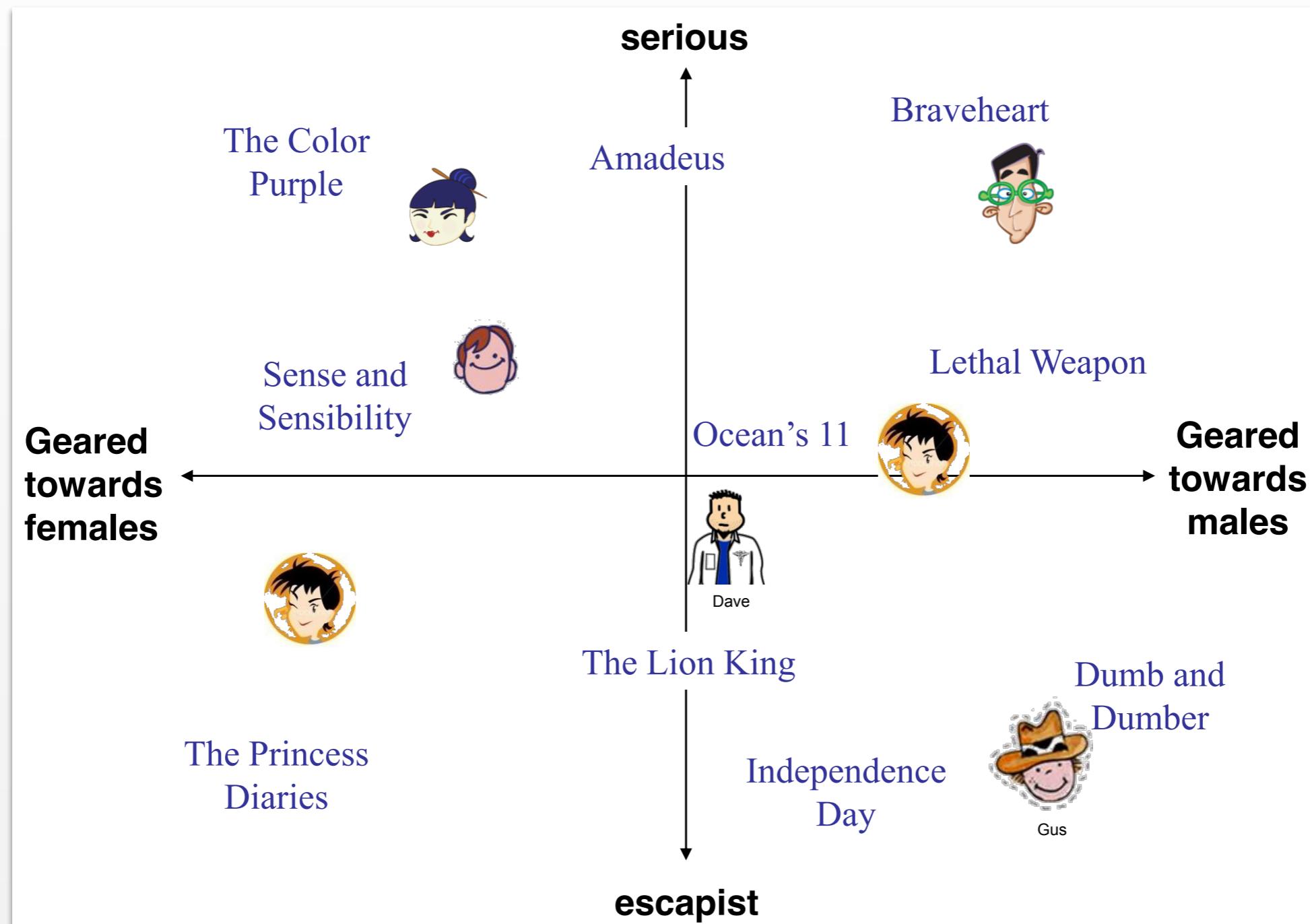


Content-based Filtering



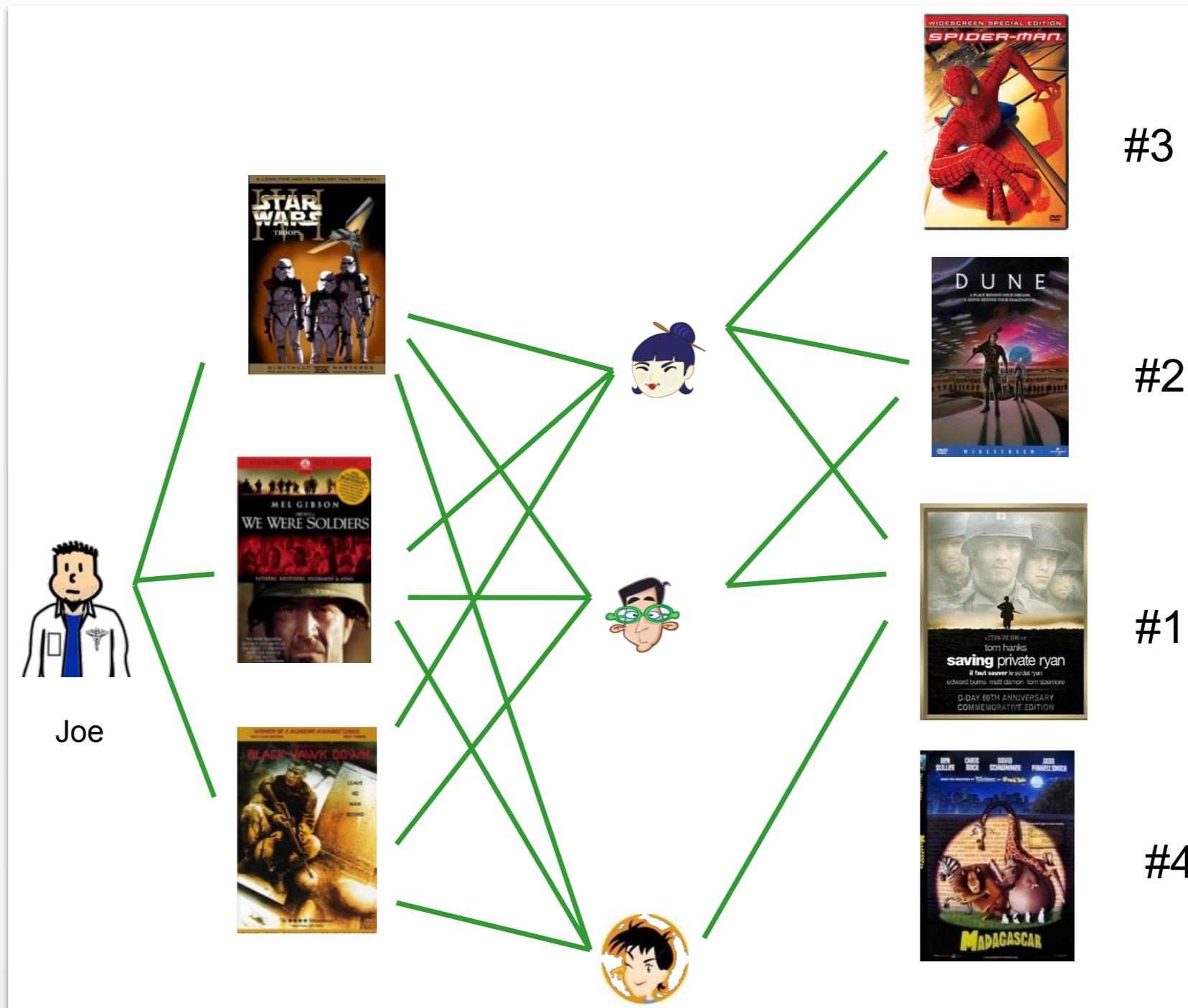
Idea: Predict rating using **item** features on a **per-user** basis

Content-based Filtering



Idea: Predict rating using **user** features on a ***per-item*** basis

Collaborative Filtering



Idea: Predict rating based on similarity to other users

Problem Setting

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

- Task: Predict user preferences for unseen items
- *Content-based filtering*: Model user/item features
- *Collaborative filtering*: Implicit similarity of users items

Recommender Systems

- 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 (Everyone)

Challenges

- *Scalability*
 - Millions of objects
 - 100s of millions of users
- *Cold start*
 - Changing user base
 - Changing inventory
- *Imbalanced dataset*
 - User activity / item reviews power law distributed
 - Ratings are not missing at random

Running Example: Netflix Data

Training data				Test data			
user	movie	date	score	user	movie	date	score
1	21	5/7/02	1	1	62	1/6/05	?
1	213	8/2/04	5	1	96	9/13/04	?
2	345	3/6/01	4	2	7	8/18/05	?
2	123	5/1/05	4	2	3	11/22/05	?
2	768	7/15/02	3	3	47	6/13/02	?
3	76	1/22/01	5	3	15	8/12/01	?
4	45	8/3/00	4	4	41	9/1/00	?
5	568	9/10/05	1	4	28	8/27/05	?
5	342	3/5/03	2	5	93	4/4/05	?
5	234	12/28/00	2	5	74	7/16/03	?
6	76	8/11/02	5	6	69	2/14/04	?
6	56	6/15/03	4	6	83	10/3/03	?

- Released as part of \$1M competition by Netflix in 2006
- Prize awarded to BellKor in 2009

Running Yardstick: RMSE

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

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

Running Yardstick: RMSE

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

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

(doesn't tell you how to actually do recommendation)

Ratings aren't everything

Netflix then

The screenshot shows the original Netflix website design. At the top, there's a red header bar with the word "NETFLIX" in white. Below it, a navigation bar includes "Watch Instantly", "Browse DVDs", "Your Queue", and "Suggestions For You". A user profile "Amanda" is shown on the right. The main content area features two sections: "Critically-acclaimed Visually-striking Dramas" and "Emotional Biographical Underdog Movies". Each section displays movie posters with titles like "AMÉLIE", "127 Hours", "The Departed", "The Pride of the Yankees", "Cinderella Man", and "INVINCIBLE". Each poster has an "Add" button and a rating scale below it. On the left side, there are sidebar links for "Genres", "New Releases", "Netflix Top 100", "Critics' Picks", and "Award Winners".

Netflix now

The screenshot shows the updated Netflix website. The top navigation bar includes "MY LIST", "Browse", "Kids", and "DVD". A search bar and a user profile "Jan Willem" are on the right. The main content area features a "Because you liked This" section with a play button over a thumbnail of "The Imitation Game". Below it is a "Trending Now" section with thumbnails for "MASCOTS", "THE CAGE", "THE RANCH", and "Haters Back Off!". The bottom section is labeled "NETFLIX ORIGINALS" and shows posters for "FIGHTER", "The Pride of the Yankees", "Cinderella Man", "INVINCIBLE", "MARK WAHLBERG", and a gold Oscar statuette.

Content-based Filtering

Item-based Features

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

Item-based Features

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Item-based Features

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Per-user Regression

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Learn a set of regression coefficients for each user

$$\mathbf{w}_u = \underset{\mathbf{w}}{\operatorname{argmin}} |\mathbf{r}_u - \mathbf{X}\mathbf{w}|^2$$

Bias

Bias

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Bias

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	5	4	4	0.3	0.2

Bias

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	5	4	4	0.3	0.2

Problem: Some movies are universally loved / hated

Bias

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	3	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	3	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	3	4	4	0.3	0.2

Problem: Some movies are universally loved / hated
some users are more picky than others

Bias

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	5	4	4	0.3	0.2

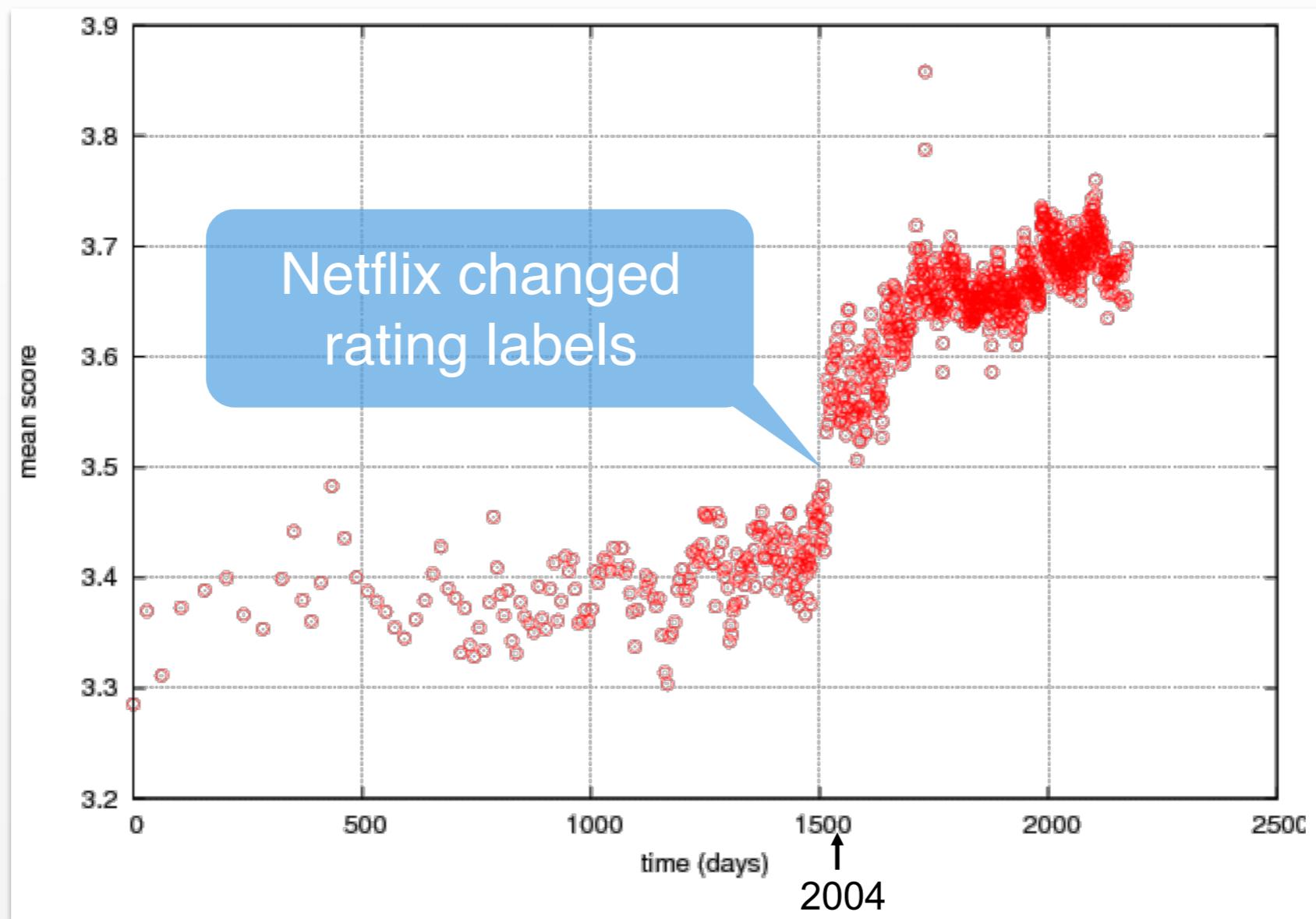
Problem: Some movies are universally loved / hated
some users are more picky than others

Solution: Introduce a per-movie and per-user bias

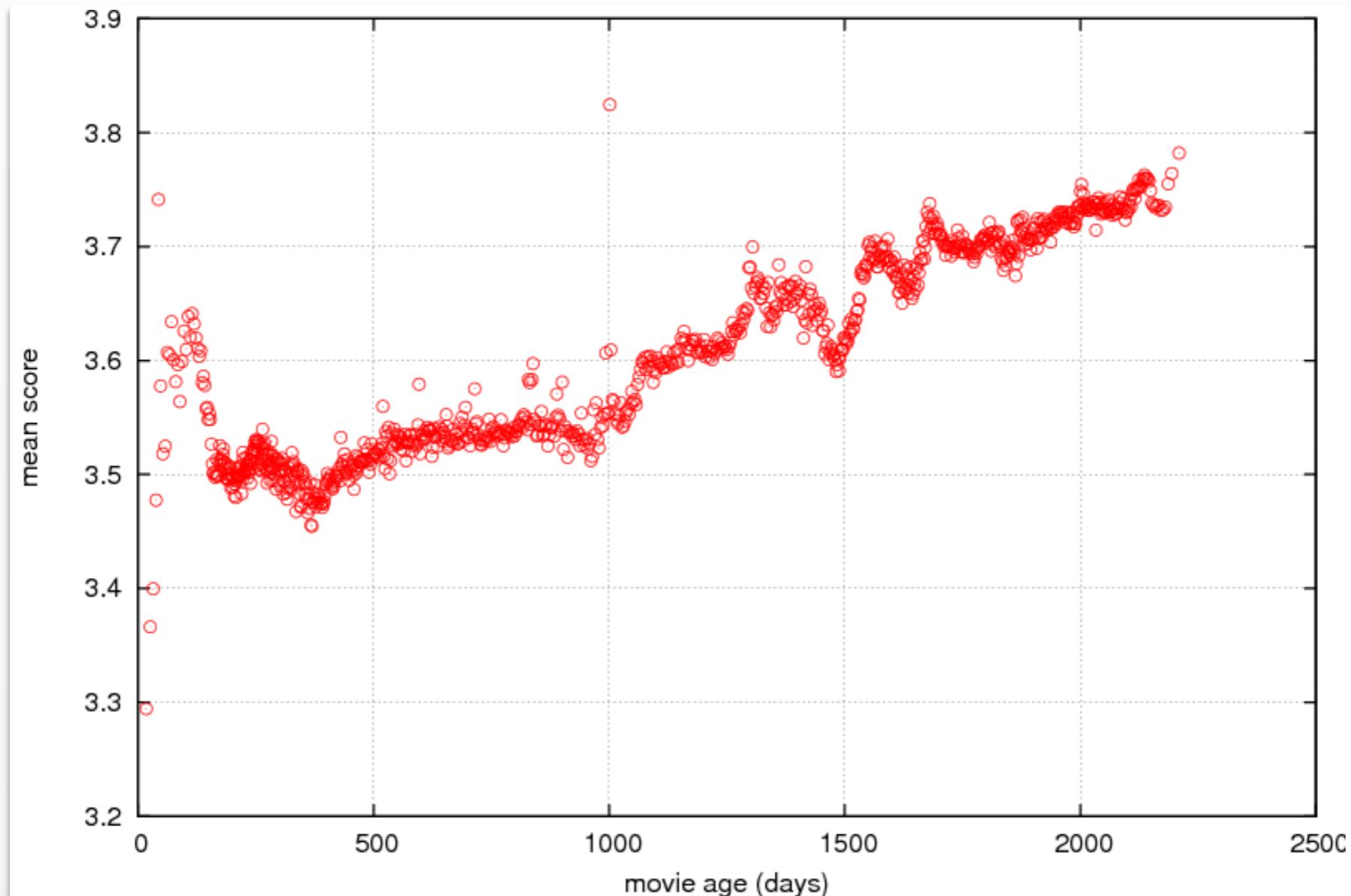
$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{x}_i^\top \mathbf{w}_u$$

Temporal Effects

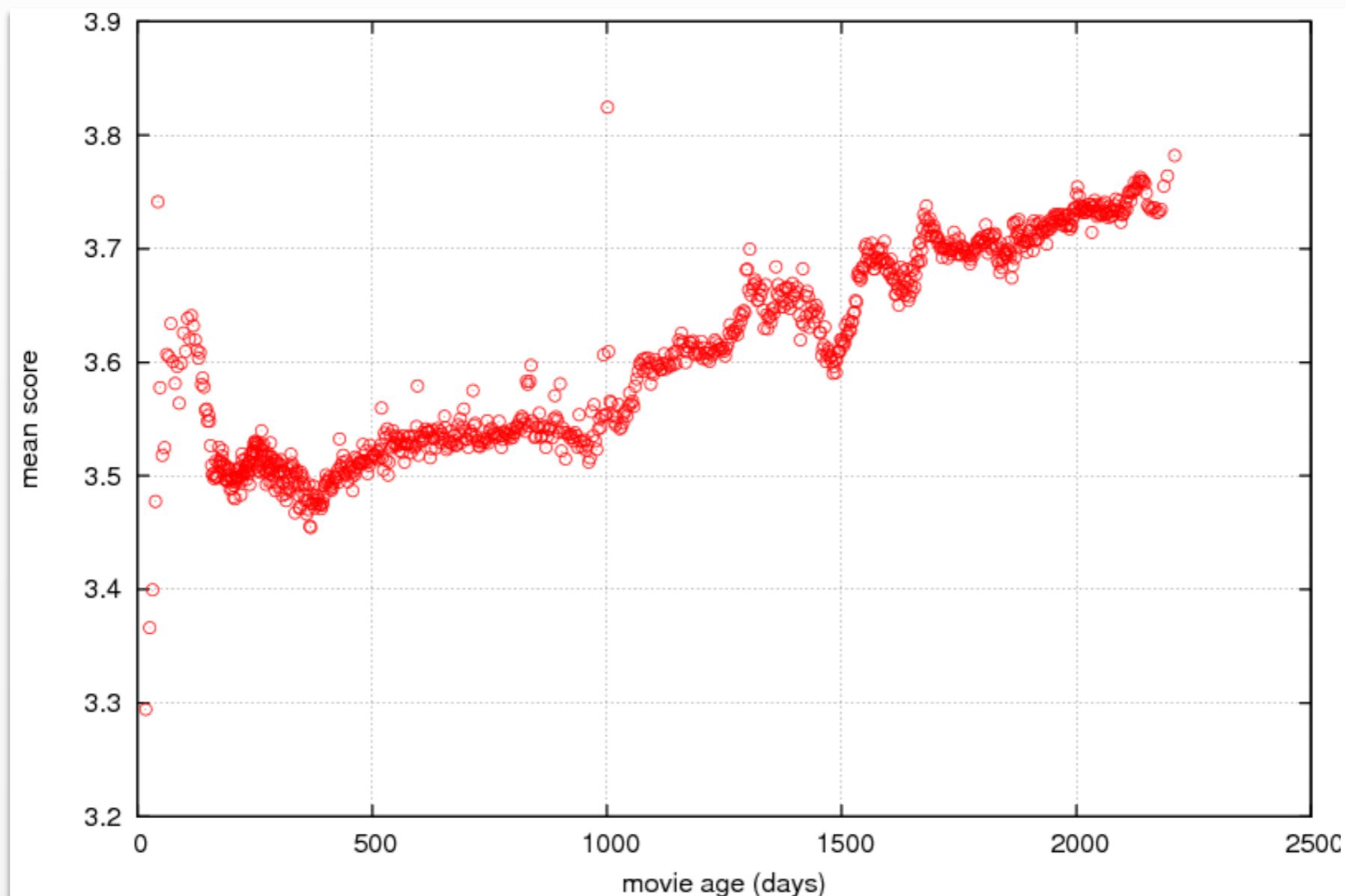
Changes in user behavior



Movies get better with time?



Temporal Effects

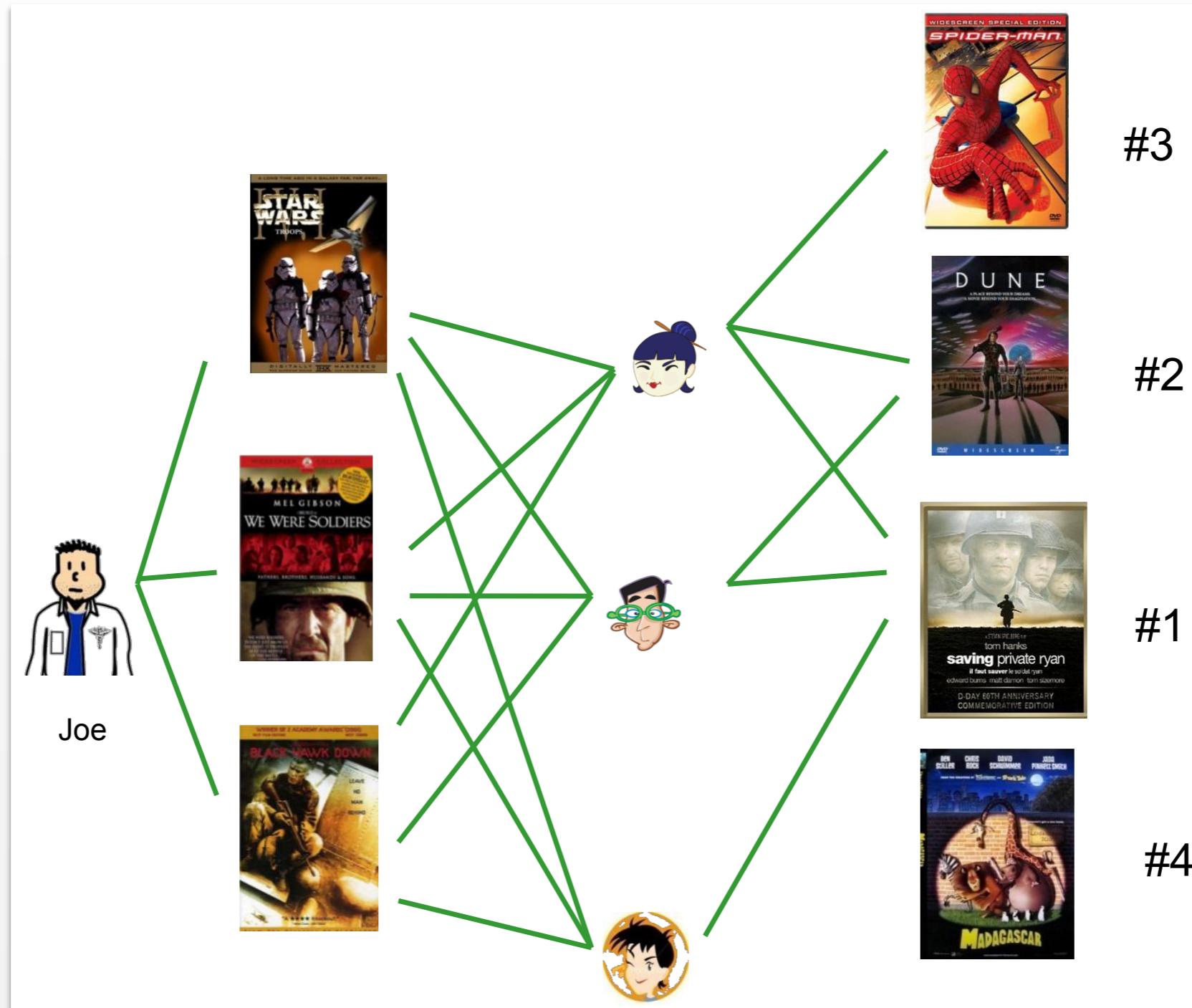


Solution: Model temporal effects in bias not weights

$$\hat{r}_{ui} = \mu(t) + b_u(t) + b_i(t) + \mathbf{x}_i^\top \mathbf{w}_u$$

Neighborhood Methods

Neighborhood Based Methods



Users and items form a bipartite graph (edges are ratings)

Neighborhood Based Methods

(user, user) similarity

- predict rating based on average from k-nearest users
- good if item base is smaller than user base
- good if item base changes rapidly

(item,item) similarity

- predict rating based on average from k-nearest items
- good if the user base is small
- good if user base changes rapidly

Parzen-Window Style CF

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

- Define a similarity s_{ij} between items
- Find set $s_k(i,u)$ of k -nearest neighbors to i that were rated by user u
- Predict rating using weighted average over set
- How should we define s_{ij} ?

Pearson Correlation Coefficient

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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$$s_{ij} = \frac{\text{Cov}[r_{ui}, r_{uj}]}{\text{Std}[r_{ui}] \text{Std}[r_{uj}]}$$

(item,item) similarity

Empirical estimate of Pearson correlation coefficient

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

Regularize towards 0 for small support

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

Regularize 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 labels

Pearson correlation not meaningful for binary labels
(e.g. Views, Purchases, Clicks)

Jaccard similarity

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

Observed / Expected ratio

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

m_i users acting on i

m_{ij} users acting on both i and j

m total number of users

Matrix Factorization Methods

Matrix Factorization

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	5	4	4	0.3	0.2

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{x}_i^\top \mathbf{w}_u$$

Matrix Factorization

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
Moonrise Kingdom	4	5	4	4	0.3	0.2

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{x}_i^\top \mathbf{w}_u$$

Idea: pose as (biased) matrix factorization problem

$$\hat{\mathbf{R}} = \mathbf{B} + \mathbf{X}\mathbf{W}^\top$$

Matrix Factorization

users

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	

~

users

~

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

A rank-3 SVD approximation

Prediction

users

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

~

users

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

items

~

.

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

Prediction

users

1		3		5		5		4	
		5		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

~

~

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

SVD with missing values

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

Pose as regression problem

$$\operatorname{argmin}_{X, W} \sum_{(u,i) \in S} (r_{ui} - w_u^\top x_i)^2 + \lambda (||X||_F^2 + ||W||_F^2)$$

Regularize using Frobenius norm

$$||A||_F^2 = \sum_{ij} |A_{ij}|^2$$

Alternating Least Squares

R

X

W^\top

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

$$w_u \leftarrow \left[\lambda I + \sum_{i:(u,i) \in S} x_i x_i^\top \right]^{-1} \sum_{i:(u,i) \in S} x_i r_{ui} \quad (\text{regress } w_u \text{ given } X)$$

Alternating Least Squares

R

X

W^T

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	

\sim

.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

$$w_u \leftarrow \left[\lambda I + \sum_{i:(u,i) \in S} x_i x_i^\top \right]^{-1} \sum_{i:(u,i) \in S} x_i r_{ui} \quad (\text{regress } w_u \text{ given } X)$$

L2: closed form solution

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

Remember
ridge regression?

Alternating Least Squares

R

X

W^\top

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

$$w_u \leftarrow \left[\lambda I + \sum_{i:(u,i) \in S} x_i x_i^\top \right]^{-1} \sum_{i:(u,i) \in S} x_i r_{ui} \quad (\text{regress } w_u \text{ given } X)$$

$$x_i \leftarrow \left[\lambda I + \sum_{u:(u,i) \in S} w_u w_u^\top \right]^{-1} \sum_{u:(u,i) \in S} w_u r_{ui} \quad (\text{regress } x_i \text{ given } W)$$

Stochastic Gradient Descent

$$R \sim X w^\top$$

A diagram illustrating the components of Stochastic Gradient Descent. On the left is a sparse matrix R with yellow entries representing non-zero values. In the center is a small matrix X . To the right is a large matrix w^\top . A tilde symbol (\sim) is placed between R and X , indicating their relationship.

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

$$w_u \leftarrow (1 - \lambda \eta_t) w_u + \eta_t x_i (r_{ui} - w_u^\top x_i)$$

$$x_i \leftarrow (1 - \lambda \eta_t) x_i + \eta_t w_u (r_{ui} - w_u^\top x_i)$$

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

Netflix Prize

Netflix Prize

Training data

- 100 million ratings, 480,000 users, 17,770 movies
- 6 years of data: 2000-2005

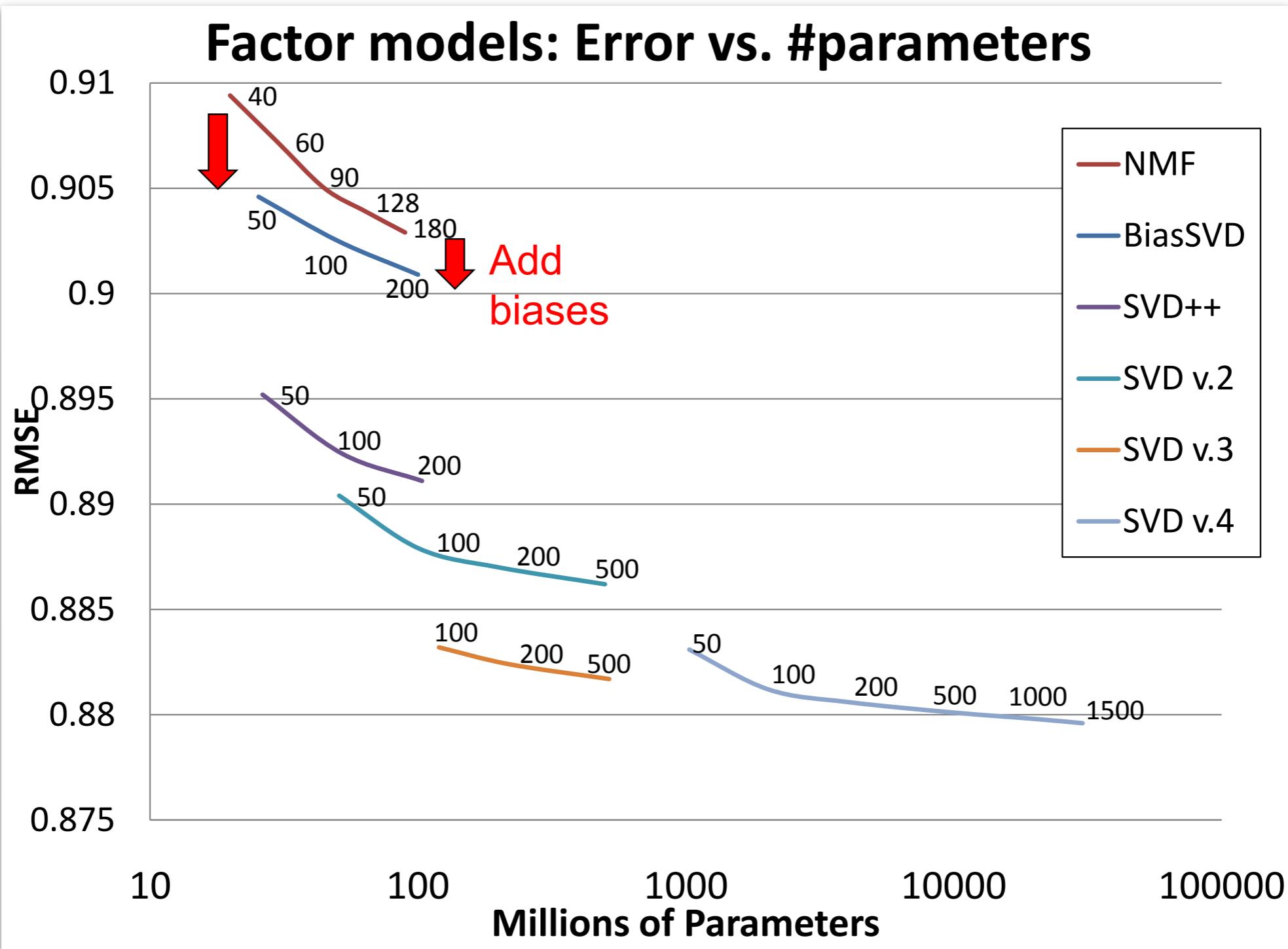
Test data

- Last few ratings of each user (2.8 million)
- Evaluation criterion: Root Mean Square Error (RMSE)

Competition

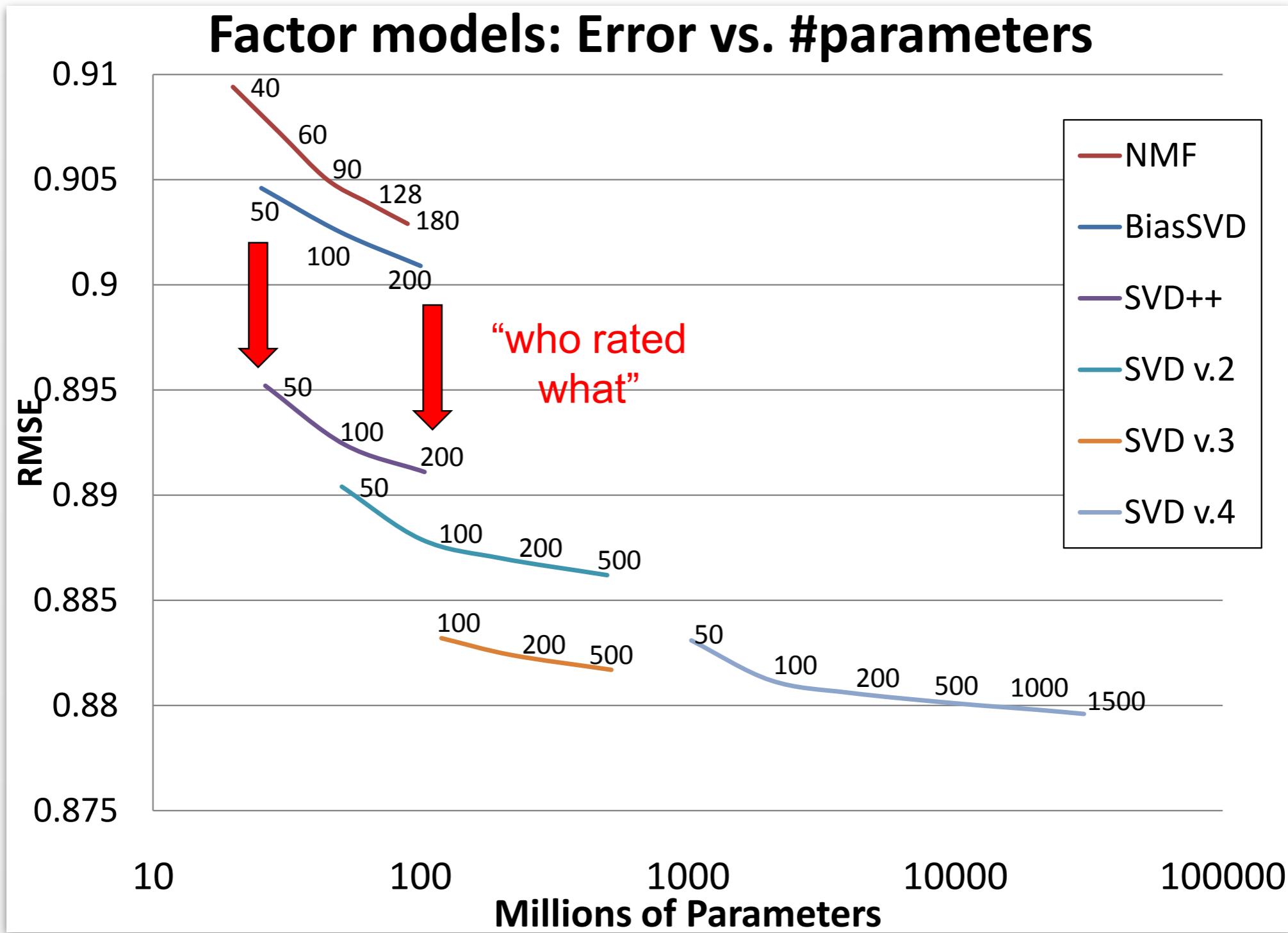
- 2,700+ teams
- Netflix's system RMSE: 0.9514
- \$1 million prize for 10% improvement on Netflix

Improvements



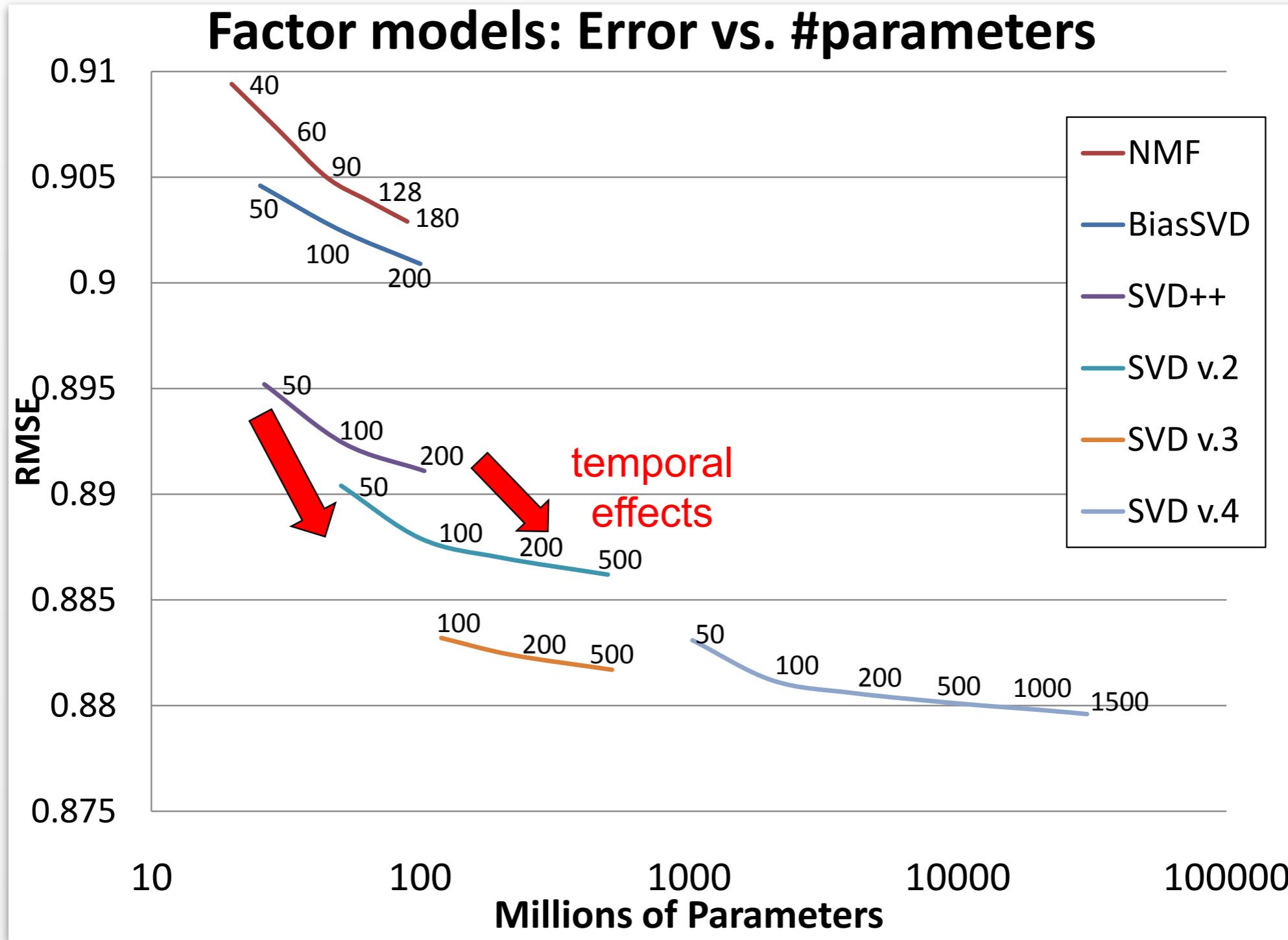
Do SGD, but also learn biases μ , b_u and b_i

Improvements



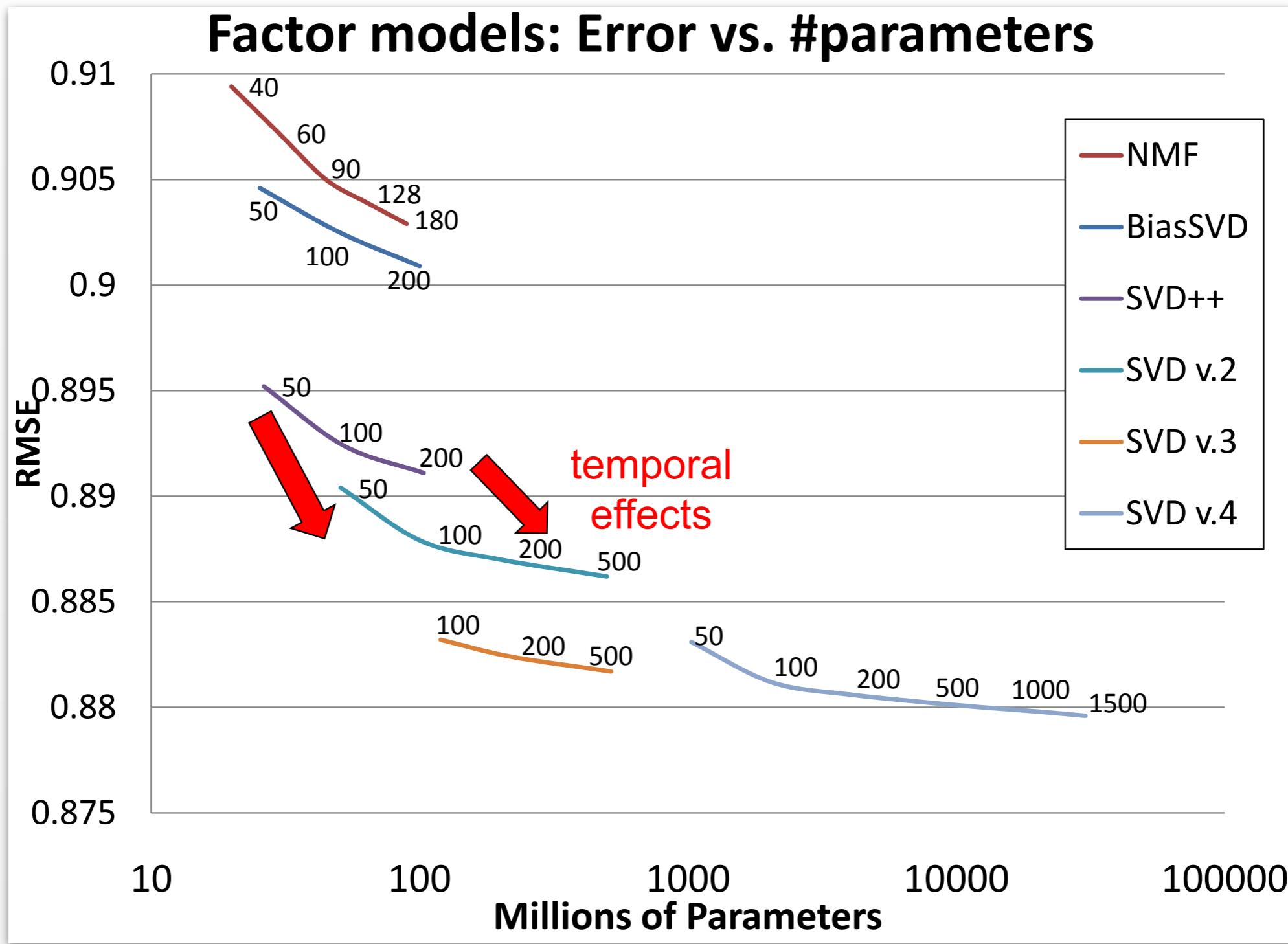
Account for fact that ratings are not missing at random.

Improvements



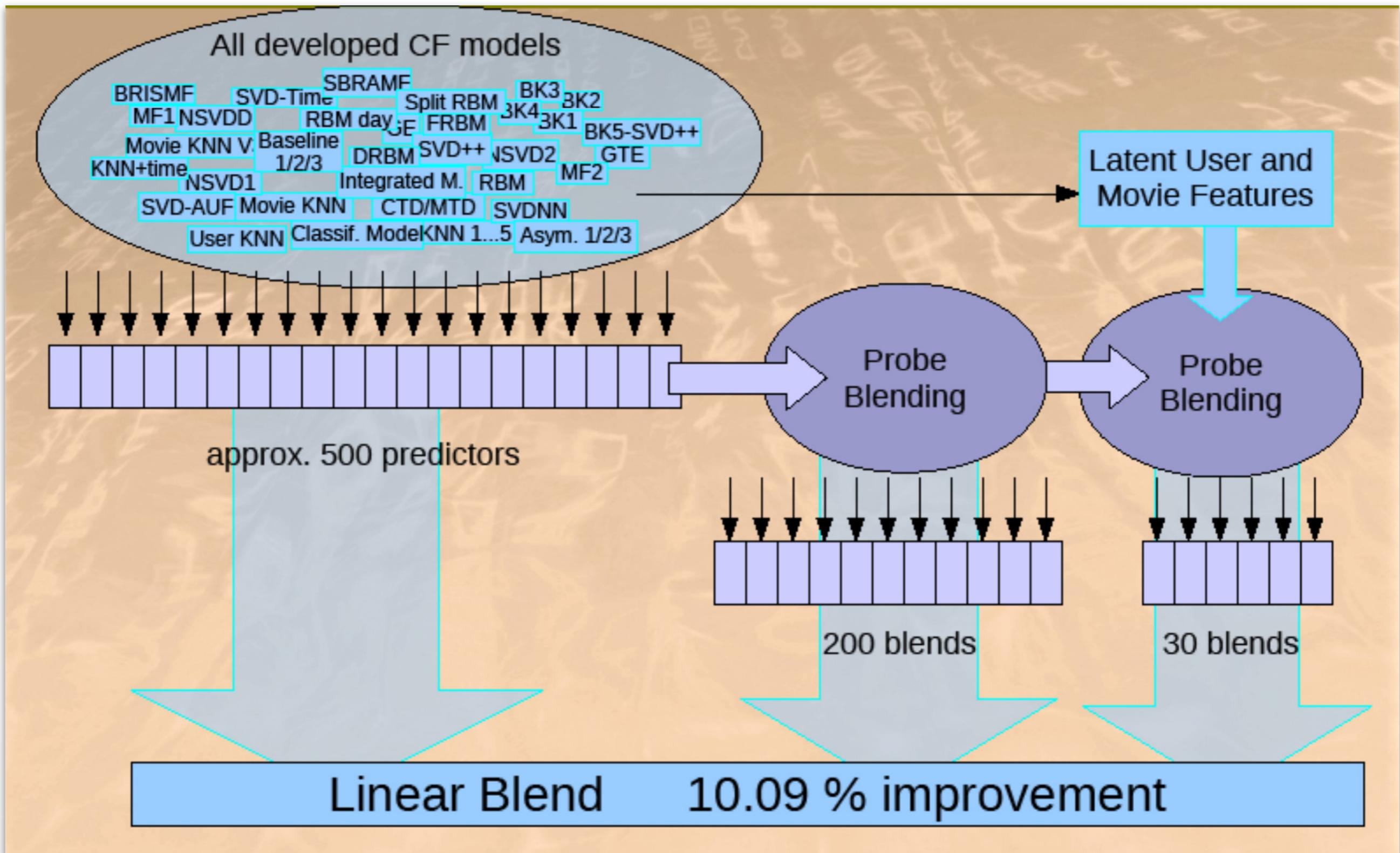
Account for drift in user and item biases

Improvements



Still pretty far from 0.8563 grand prize

Winning Solution from BellKor



Last 30 days

The screenshot shows the Netflix Prize Leaderboard page. At the top, the Netflix logo is visible, followed by the "Netflix Prize" title and a yellow banner with three stars. Below the banner is a navigation menu with links: Home, Rules, Leaderboard, Register, Update, Submit, and Download. The main section is titled "Leaderboard" in large blue letters. To the right of the title is a dropdown menu set to "Display top 20 leaders". The table below lists 13 teams, each with their rank, team name, best score, percentage improvement, and last submit time. A red bar highlights the top entry, and a blue bar highlights the winning team. A note at the bottom states: "June 26th submission triggers 30-day “last call”".

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
7	BellKor	0.8620	9.40	2009-06-24 07:16:02
8	Gravity	0.8634	9.25	2009-04-22 18:31:32
9	Opera Solutions	0.8638	9.21	2009-06-26 23:18:13
10	BruceDengDaoCiYiYou	0.8638	9.21	2009-06-27 00:55:55
11	pengpengzhou	0.8638	9.21	2009-06-27 01:06:43
12	xlvector	0.8639	9.20	2009-06-26 13:49:04
13	xiangliang	0.8639	9.20	2009-06-26 07:47:34

June 26th submission triggers 30-day “last call”

Last 30 days

The screenshot shows the Netflix Prize Leaderboard page. At the top, the Netflix logo is visible, followed by the "Netflix Prize" title and a yellow banner with three stars. Below the banner is a navigation menu with links: Home, Rules, Leaderboard, Register, Update, Submit, and Download. The main section is titled "Leaderboard" in large blue letters. To the right of the title is a dropdown menu set to "Display top 20 leaders". The table below lists 13 teams, each with a rank, team name, best score, percentage improvement, and last submit time. A red bar highlights the top entry, and a blue bar highlights the winning team. A note at the bottom states: "June 26th submission triggers 30-day “last call”".

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
7	BellKor	0.8620	9.40	2009-06-24 07:16:02
8	Gravity	0.8634	9.25	2009-04-22 18:31:32
9	Opera Solutions	0.8638	9.21	2009-06-26 23:18:13
10	BruceDengDaoCiYiYou	0.8638	9.21	2009-06-27 00:55:55
11	pengpengzhou	0.8638	9.21	2009-06-27 01:06:43
12	xlvector	0.8639	9.20	2009-06-26 13:49:04
13	xiangliang	0.8639	9.20	2009-06-26 07:47:34

June 26th submission triggers 30-day “last call”

BellKor fends off competitors by a hair

Netflix Prize **COMPLETED**

Home | Rules | Leaderboard | Update | Download

Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Display top leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8562	9.98	2009-07-26 18:24:48
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50

BellKor fends off competitors by a hair

