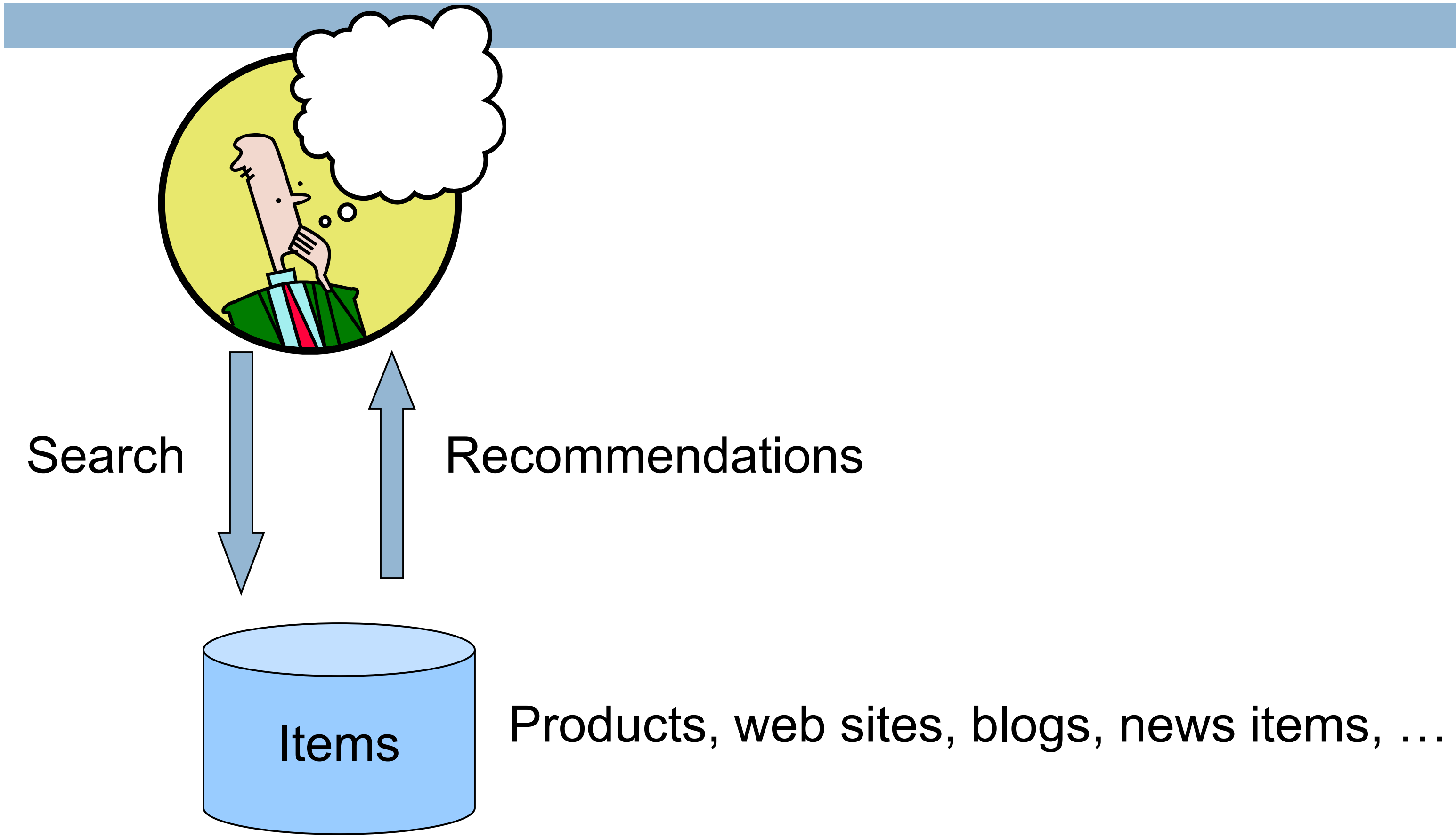
LECTURE 13: RECOMMENDER SYSTEMS

COMP4641: Social Information Network Analysis and Engineering

Wednesday April 1st 2015

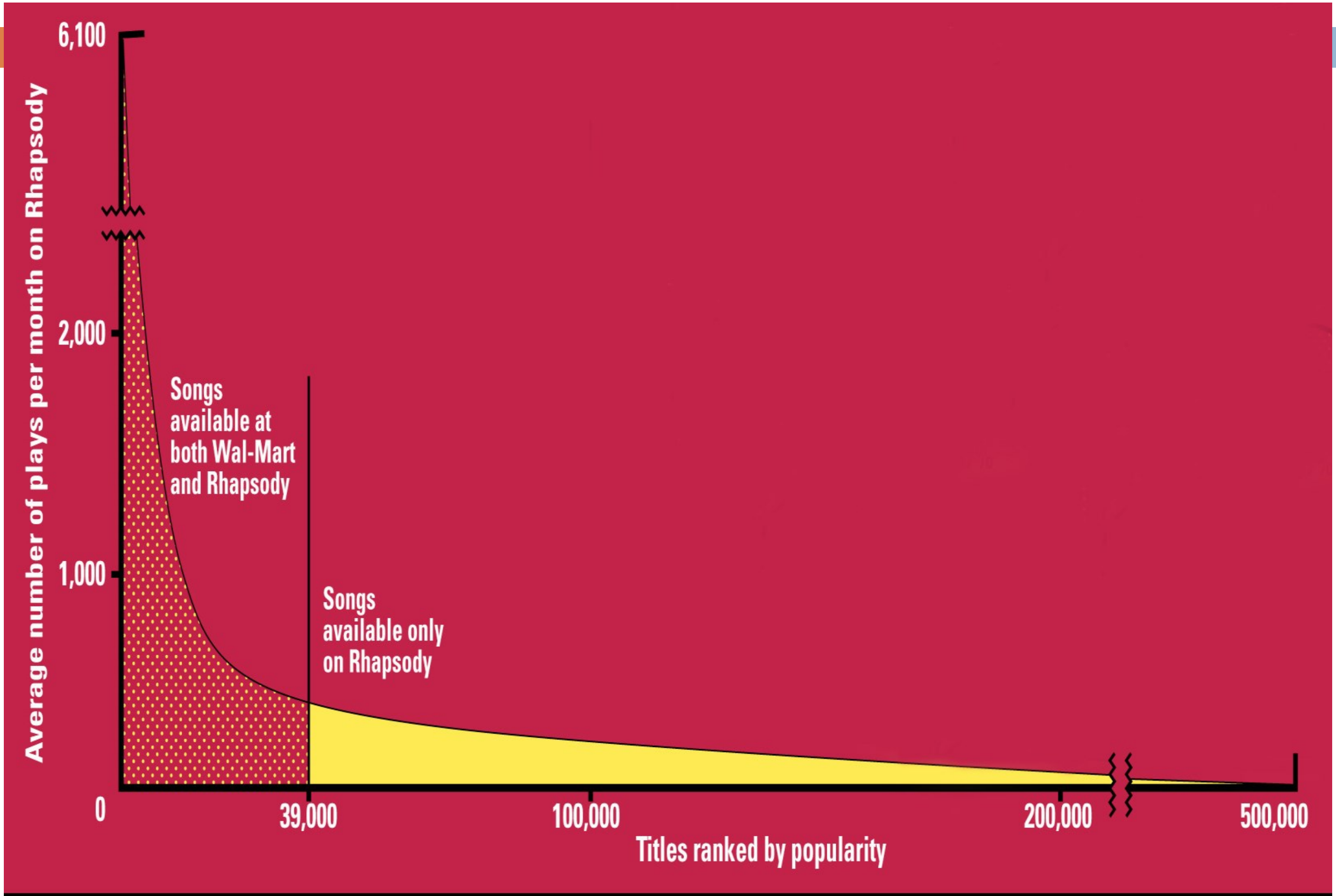
Recommendations



From scarcity to abundance

- Shelf space is a scarce commodity for traditional retailers
  - Also: TV networks, movie theaters,…
- The web enables near-zero-cost dissemination of information about products
  - From scarcity to abundance
- More choice necessitates better filters
  - Recommendation engines
  - How *Into Thin Air* made *Touching the Void* a bestseller

The Long Tail



Source: Chris Anderson (2004)

Sources: Erik Brynjolfsson and Jeffrey Hu, MIT, and Michael Smith, Carnegie Mellon; Barnes & Noble; Netflix; RealNetworks

Recommendation Types

- Editorial
- Simple aggregates
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users
  - Amazon, Netflix, …

Formal Model

- C = set of Customers
- *S* = set of Items
- Utility function *u*: C x *S* -> *R*
  - *R* = set of ratings
  - *R* is a totally ordered set
  - e.g., 0-5 stars, real number in [0,1]

Utility Matrix

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

Key Problems

- Gathering "known" ratings for matrix
- Extrapolate unknown ratings from known ratings
  - Mainly interested in high unknown ratings
- Evaluating extrapolation methods

Gathering Ratings

- Explicit
  - Ask people to rate items
  - Doesn't work well in practice – people can't be bothered
- Implicit
  - Learn ratings from user actions
  - e.g., purchase implies high rating
  - What about low ratings?

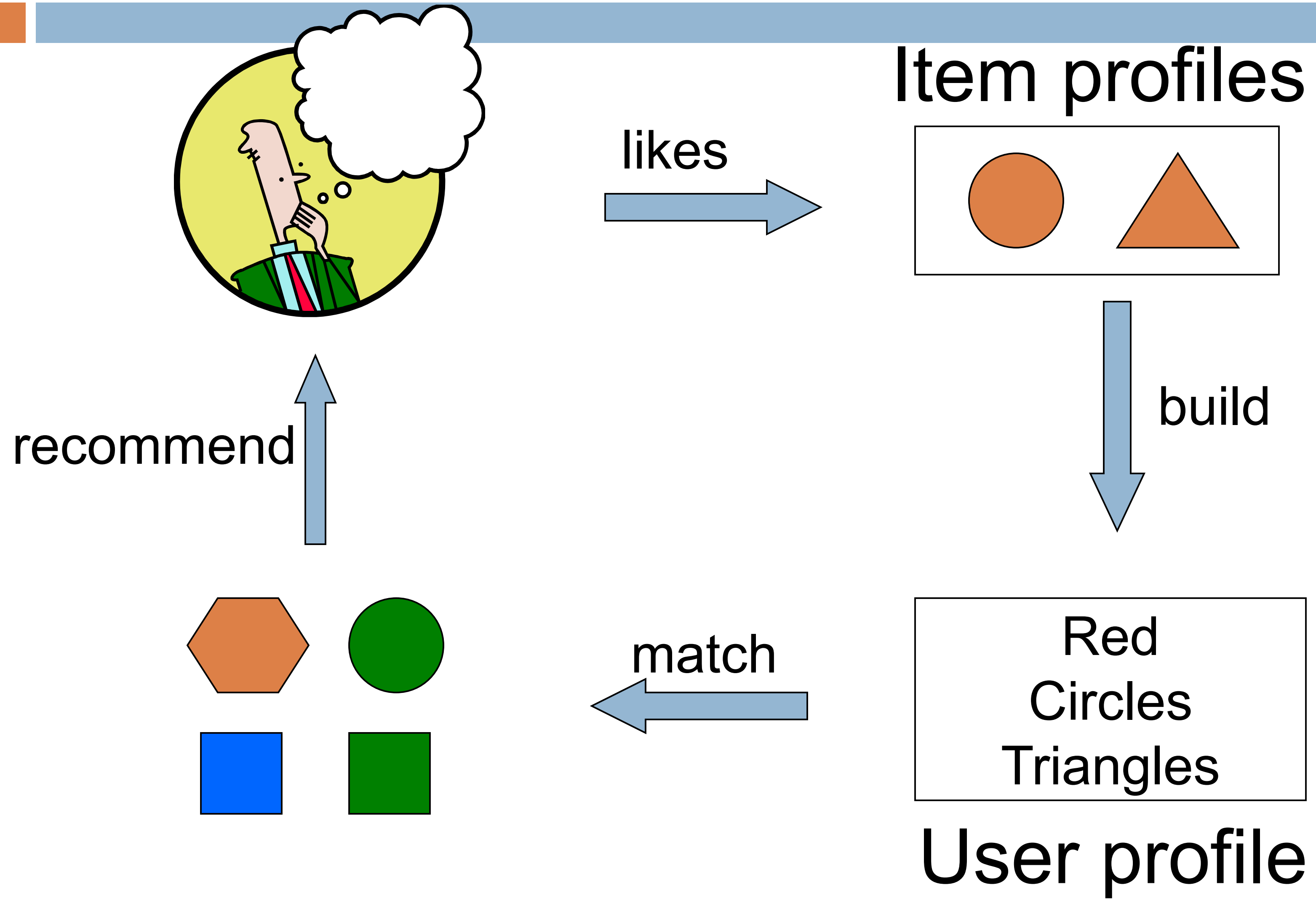
Extrapolating Utilities

- Key problem: matrix U is sparse
  - most people have not rated most items
- Three approaches
  - Content-based
  - Collaborative
  - Hybrid

Content-based recommendations

- Main idea: recommend items to customer C similar to previous items rated highly by C
- Movie recommendations
  - recommend movies with same actor(s), director, genre, …
- Websites, blogs, news
  - recommend other sites with "similar" content

Plan of action



Item Profiles

- For each item, create an item profile
- Profile is a set of features
  - movies: author, title, actor, director,…
  - text: set of "important" words in document
- How to pick important words?
  - Usual heuristic is TF.IDF (Term Frequency times Inverse Doc Frequency)

TF.IDF

f_{ij} = frequency of term t_i in document d_j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

n_i = number of docs that mention term i

N = total number of docs

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF score $w_{ij} = Tf_{ij} \times IDF_i$

Doc profile = set of words with highest TF.IDF scores, together with their scores

User profiles and prediction

- User profile possibilities:
  - Weighted average of rated item profiles
  - Variation: weight by difference from average rating for item
  - …
- Prediction heuristic
  - Given user profile **c** and item profile **s**, estimate $u(\mathbf{c},\mathbf{s}) = \cos(\mathbf{c},\mathbf{s}) = \mathbf{c}.\mathbf{s}/(|\mathbf{c}||\mathbf{s}|)$
  - Need efficient method to find items with high utility: later

Advantages of Content-Based Approach

- No need for data on other users.
  - No cold-start or sparsity problems.
- Able to recommend to users with unique tastes.
- Able to recommend new and unpopular items
  - No first-rater problem.
- Can provide explanations of recommended items by listing content-features that caused an item to be recommended.

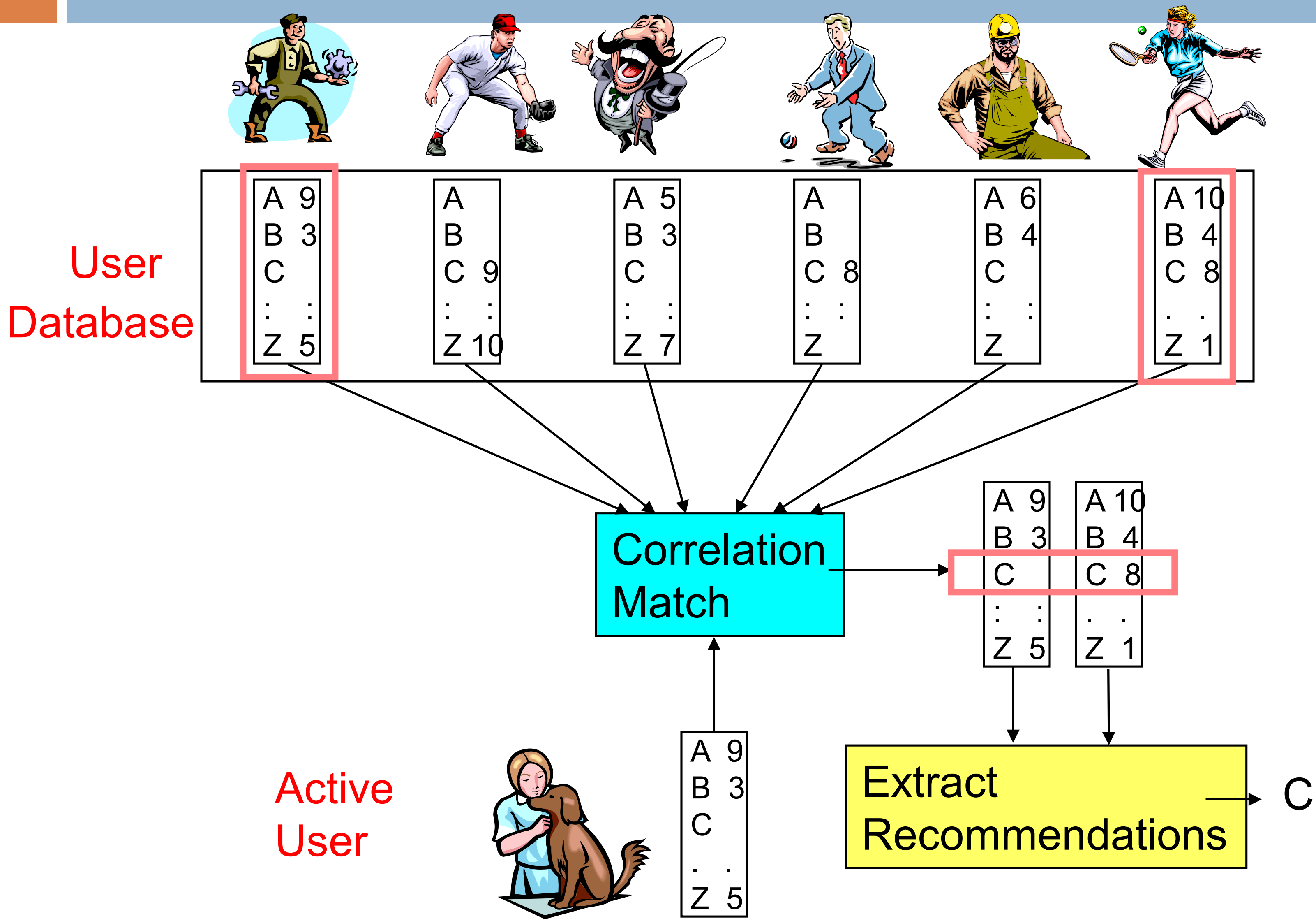
Disadvantages of Content-Based Method

- Requires content that can be encoded as meaningful features.
- Users' tastes must be represented as a learnable function of these content features.
- Unable to exploit quality judgments of other users.
  - Unless these are somehow included in the content features.

Collaborative Filtering

- Maintain a database of many users' ratings of a variety of items.
- For a given user, find other similar users whose ratings strongly correlate with the current user.
- Recommend items rated highly by these similar users, but not rated by the current user.
- Almost all existing commercial recommenders use this approach (e.g. Amazon).

Collaborative Filtering



Collaborative Filtering Method

- Weight all users with respect to similarity with the active user.
- Select a subset of the users (*neighbors*) to serve as predictors.
- Normalize ratings and compute a prediction from a weighted combination of the selected neighbors' ratings.
- Present items with highest predicted ratings as recommendations.

Similarity Weighting

- Typically use Pearson correlation coefficient between ratings for active user, *a*, and another user, *u*.

$$c_{a,u} = \frac{\mathrm{covar}(r_a, r_u)}{\sigma_{r_a} \sigma_{r_u}}$$

r_a and r_u are the rating vectors for the *m* items rated by **both** *a* and *u*

$r_{i,j}$ is user *i*'s rating for item *j*

Covariance and Standard Deviation

- Covariance:

$$\text{covar}(r_a, r_u) = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

- Standard Deviation:

$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^{m}(r_{x,i} - \bar{r}_x)^2}{m}}$$

Significance Weighting

- Important not to trust correlations based on very few co-rated items.
- Include *significance weights*, $s_{a,u}$, based on number of co-rated items, *m*.

$$w_{a,u} = s_{a,u} c_{a,u}$$

$$s_{a,u} = \begin{Bmatrix} 1 \text{ if } m > 50 \\ \dfrac{m}{50} \text{ if } m \leq 50 \end{Bmatrix}$$

Neighbor Selection

- For a given active user, *a*, select correlated users to serve as source of predictions.
- Standard approach is to use the most similar *n* users, *u*, based on similarity weights, $w_{a,u}$
- Alternate approach is to include all users whose similarity weight is above a given threshold.

Rating Prediction

- Predict a rating, $p_{a,i}$, for each item *i*, for active user, *a*, by using the *n* selected neighbor users, $u \in \{1,2,\ldots n\}$.
- To account for users different ratings levels, base predictions on *differences* from a user's *average* rating.
- Weight users' ratings contribution by their similarity to the active user.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} w_{a,u}}$$

Problems with Collaborative Filtering

- **Cold Start**: There needs to be enough other users already in the system to find a match.
- **Sparsity**: If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.
- **First Rater**: Cannot recommend an item that has not been previously rated.
  - New items
  - Esoteric items
- **Popularity Bias**: Cannot recommend items to someone with unique tastes.
  - Tends to recommend popular items.

Item-Item Collaborative Filtering

- So far: User-user collaborative filtering
- Another view
  - For item s, find other similar items
  - Estimate rating for item based on ratings for similar items
  - Can use same similarity metrics and prediction functions as in user-user model
- In practice, it has been observed that item-item often works better than user-user

Pros and cons of collaborative filtering

- Works for any kind of item
  - No feature selection needed
- New user problem
- New item problem
- Sparsity of rating matrix
  - Cluster-based smoothing?

Hybrid Methods

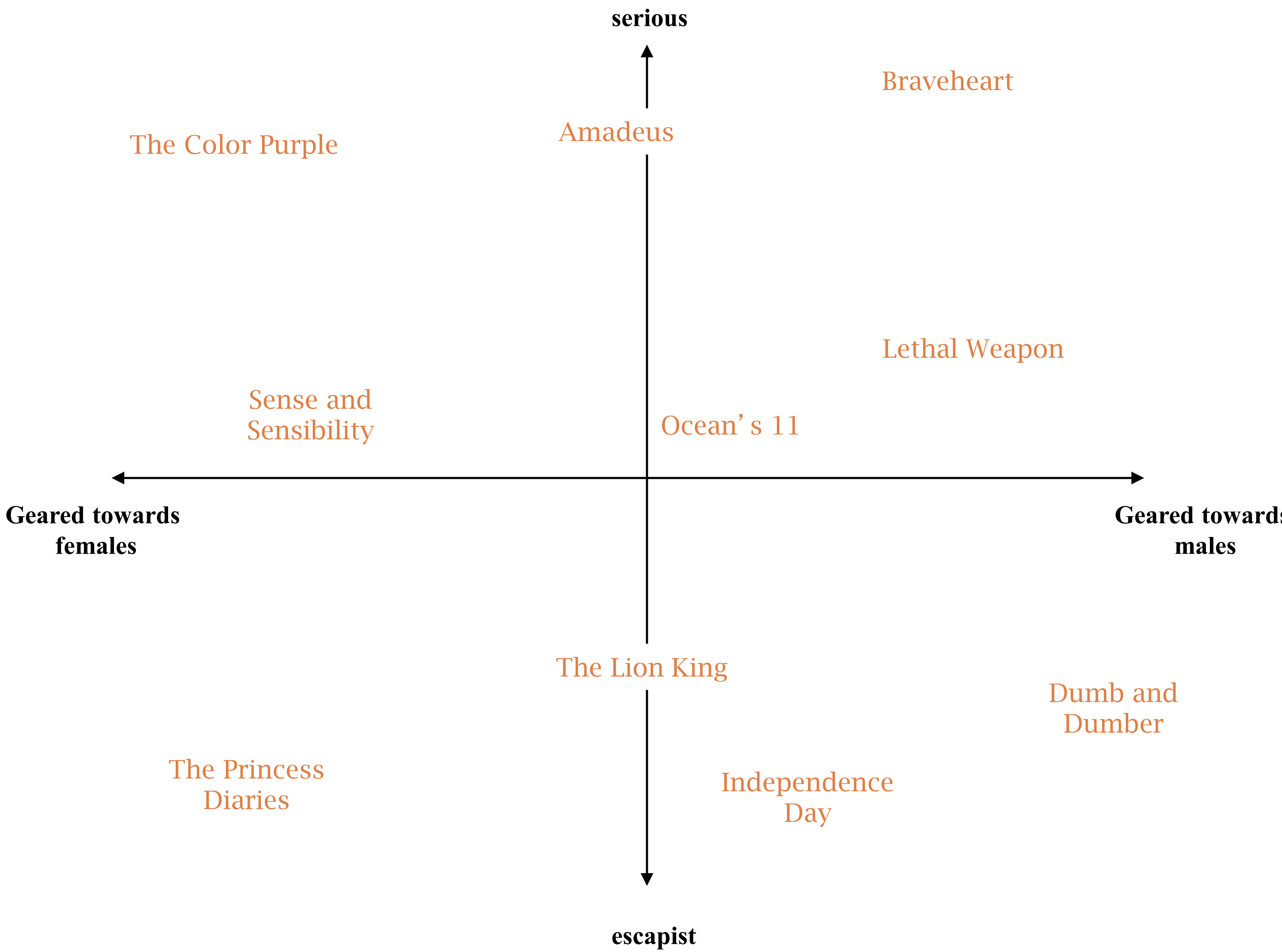
- Implement two separate recommenders and combine predictions
- Add content-based methods to collaborative filtering
  - item profiles for new item problem
  - demographics to deal with new user problem

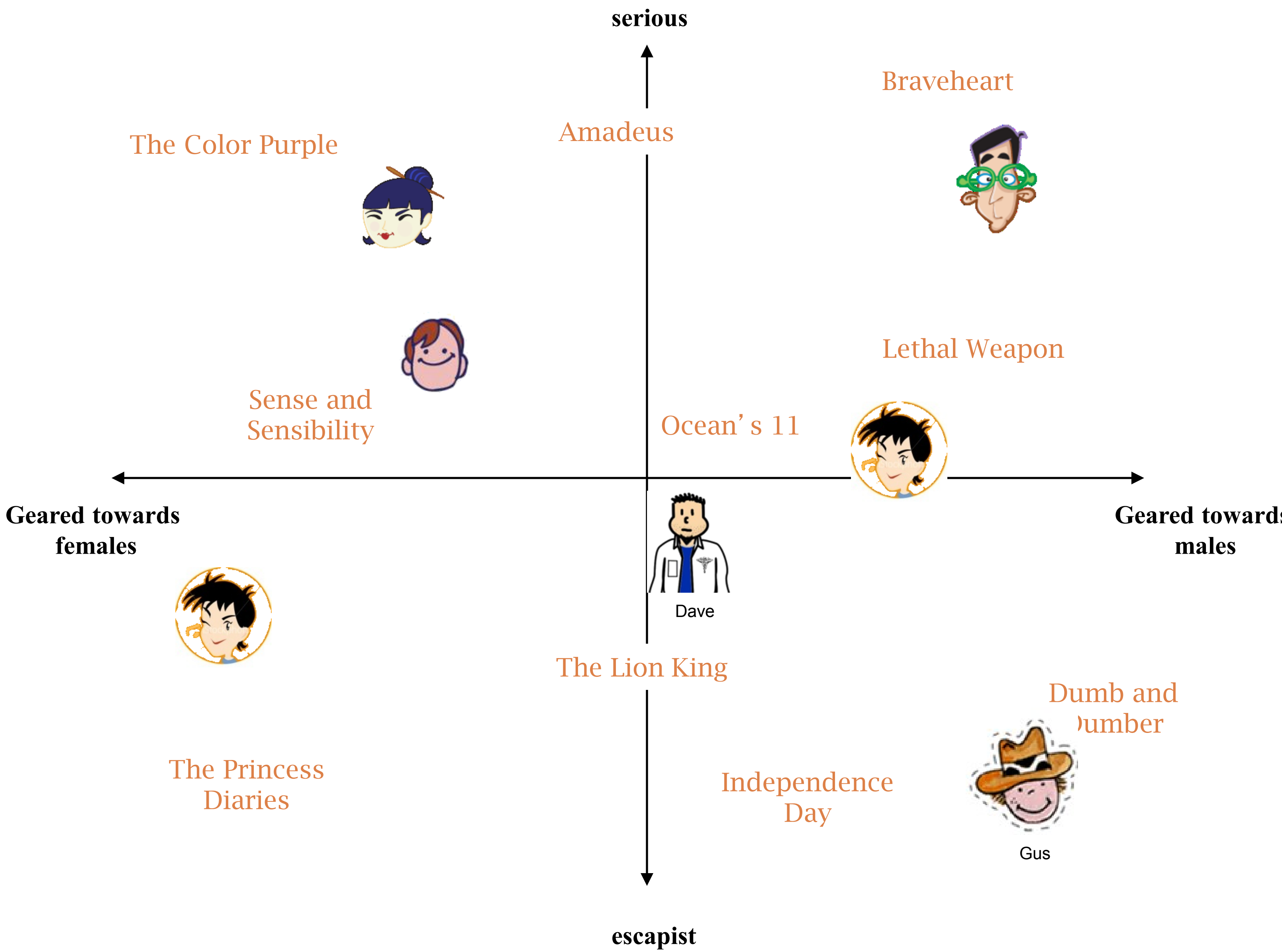
Latent Factor Models

- Models with latent classes of items and users
  - Individual items and users are assigned to either a single class or a mixture of classes
- Neural networks
  - Restricted Boltzmann machines
- Singular Value Decomposition (SVD)
  - matrix factorization
  - Items and users described by unobserved factors
  - Main method used by leaders of Netflixprize competition

Matrix Factorization (SVD)

- Dimension reduction technique for matrices
- Each item summarized by a *d*-dimensional vector q_i
- Similarly, each user summarized by p_u
- Choose *d* much smaller than number of items or users
  - *e.g.*, $d = 50 \ll 18{,}000$ or $480{,}000$
- Predicted rating for Item *i* by User *u*
  - Inner product of q_i and p_u
  - $\hat{r}_{ui} = q_i^T p_u \quad \text{or} \quad \hat{r}_{ui} = \mu + a_u + b_i + q_i^{'} p_u$


serious
Braveheart
Amadeus
The Color Purple
Lethal Weapon
Sense and Sensibility
Ocean's 11
Geared towards females
Geared towards males
The Lion King
Dumb and Dumber
The Princess Diaries
Independence Day
escapist



serious
escapist
Geared towards females
Geared towards males
The Color Purple
Amadeus
Braveheart
Lethal Weapon
Sense and Sensibility
Ocean’s 11
Dave
The Lion King
Dumb and
umber
The Princess Diaries
Independence Day
Gus

Netflixprize

"We're quite curious, really. To the tune of one million dollars." – Netflix Prize rules

- Goal to improve on Netflix's existing movie recommendation technology
- Contest began October 2, 2006
- Prize
  - Based on reduction in root mean squared error (RMSE) on test data
  - $1,000,000 grand prize for 10% drop
  - Or, $50,000 progress for best result each year

Data Details

- Training data
  - 100 million ratings (from 1 to 5 stars)
  - 6 years (2000-2005)
  - 480,000 users
  - 17,770 “movies”
- Test data
  - Last few ratings of each user
  - Split as shown on next slide

Data about the Movies

Most Loved Movies	Avg rating	Count
The Shawshank Redemption	4.593	137812
Lord of the Rings :The Return of the King	4.545	133597
The Green Mile	4.306	180883
Lord of the Rings :The Two Towers	4.460	150676
Finding Nemo	4.415	139050
Raiders of the Lost Ark	4.504	117456

Most Rated Movies
Miss Congeniality
Independence Day
The Patriot
The Day After Tomorrow
Pretty Woman
Pirates of the Caribbean

Highest Variance
The Royal Tenenbaums
Lost In Translation
Pearl Harbor
Miss Congeniality
Napolean Dynamite
Fahrenheit 9/11

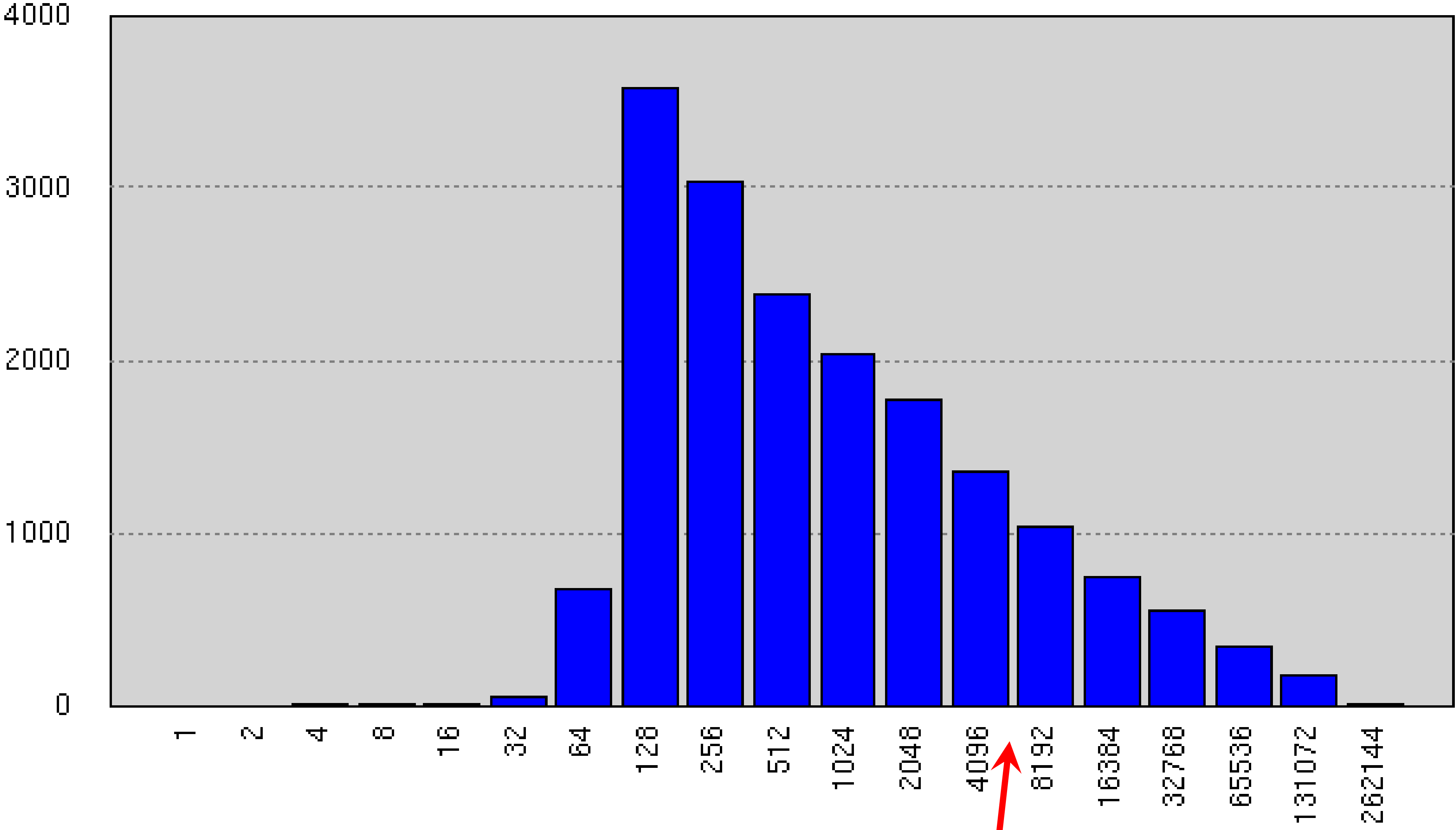
Major Challenges

1. Size of data
   - Places premium on efficient algorithms
   - Stretched memory limits of standard PCs
2. 99% of data are missing
   - Eliminates many standard prediction methods
   - Certainly *not* missing at random
3. Training and test data differ systematically
   - Test ratings are later
   - Test cases are spread uniformly across users

Major Challenges (cont.)

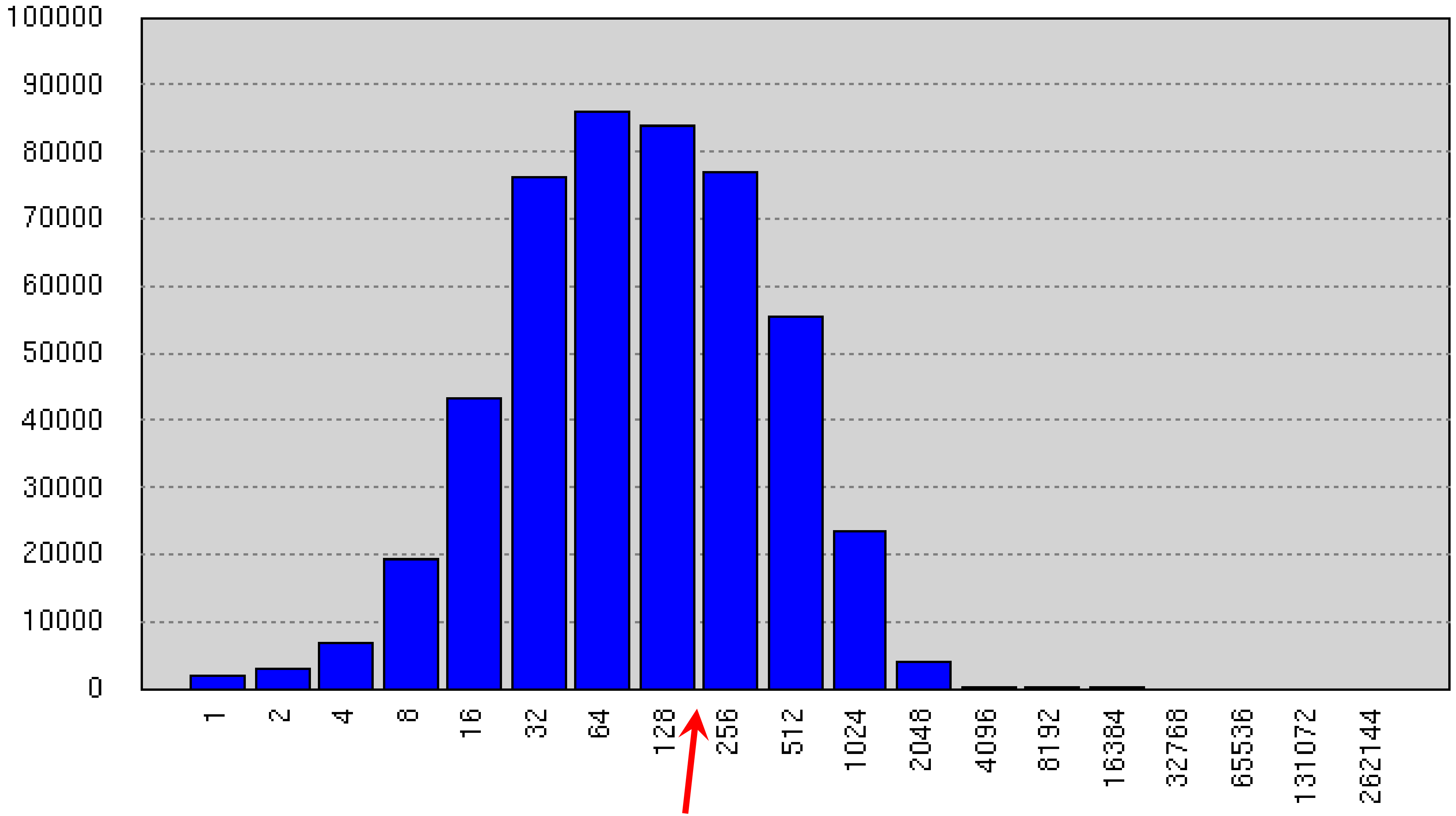
4. Countless factors may affect ratings
   - Genre, movie/TV series/other
   - Style of action, dialogue, plot, music et al.
   - Director, actors
   - Rater’s mood
5. Large imbalance in training data
   - Number of ratings per user or movie varies by several orders of magnitude
   - Information to estimate individual parameters varies widely

Ratings per Movie in Training Data



Avg #ratings/movie: 5627

Ratings per User in Training Data



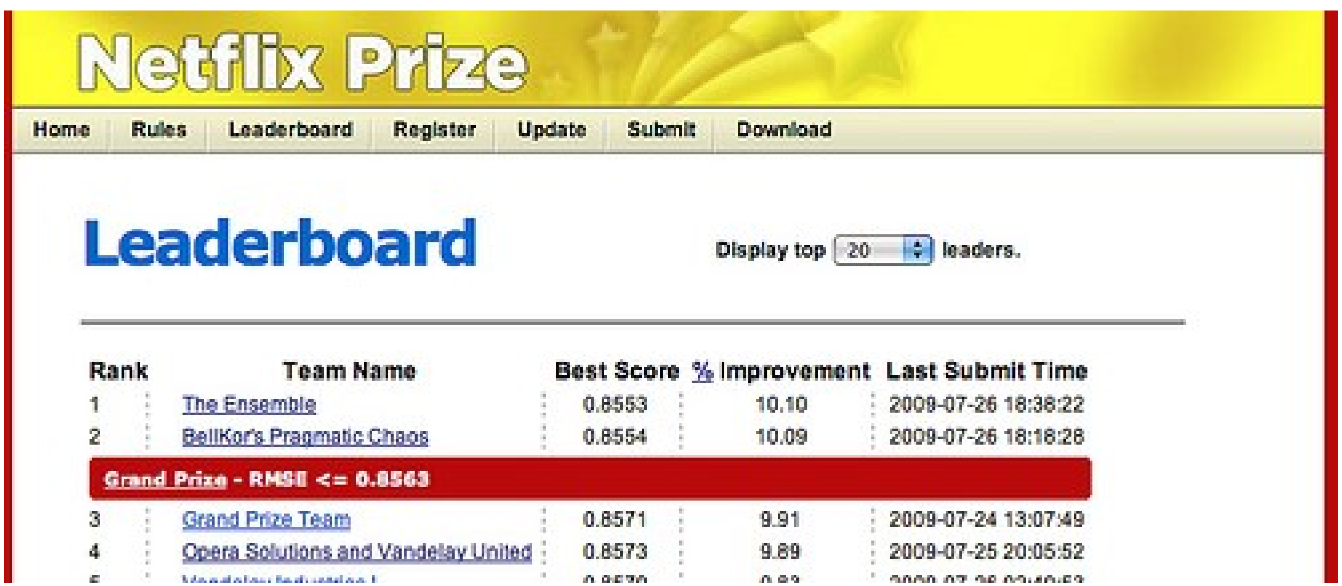
Avg #ratings/user: 208

The Fundamental Challenge

- How can we estimate as much signal as possible where there are sufficient data, without over fitting where data are scarce?

Test Set Results

- The Ensemble: 0.856714
- BellKor’s Pragmatic Theory: 0.856704
- Both scores round to 0.8567
- Tie breaker is submission date/time



Lessons from Netflixprize

- Lesson #1: Data >> Models
- Lesson #2: The Power of Regularized SVD Fit by Gradient Descent
- Lesson #3: The Wisdom of Crowds (of Models)

Recommendation System Incorporating Social Information-Problem Formulation

- Goal: Make recommendations for a target user or a group of users given:
  - m users $U = \{u_1, u_2, \ldots, u_m\}$
  - n items $I = \{i_1, i_2, \ldots, i_n\}$
  - Each user u_i has a list of rated items $I_{\{u_i\}}$
  - $m \times n$ (sparse) rating matrix
  - Social network $S = (U, Es)$.
    - U: set of nodes
    - Es: set of edges
    - For all $u, v \in U$, $(u,v) \in Es$ if v is a friend of u.

Problem Formulation Cont.

- Other information may also be available, e.g.:
  - k-dimensional tagging information $T = \{t_1, t_2, \dots, t_k\}$
    - e.g. sci-fi, romance, comedy, horror, etc.
  - l-dimensional user profile information $P = \{p_1, p_2, \dots, p_l\}$
    - e.g. age, gender, occupation, etc.

Why Incorporating Social Information?

- Known fact:
  - Homophily effect among friends (birds of a feather)
  - [McPherson et all 2001].