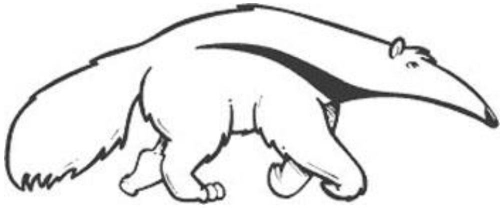
Machine Learning and Data Mining

Collaborative Filtering & Recommender Systems

Kalev Kask



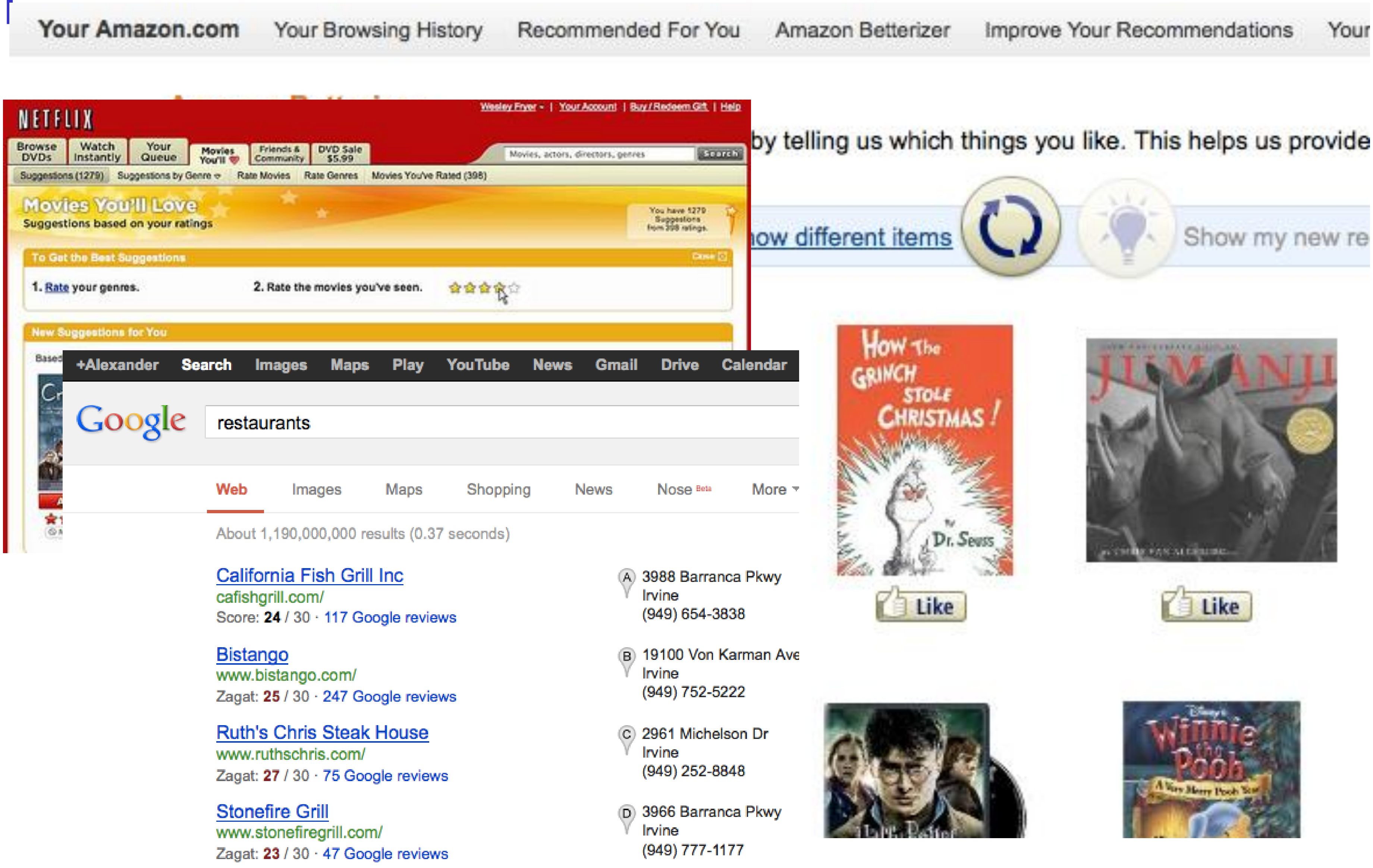
BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Recommender systems

- Automated recommendations
- Inputs
  - User information
    - Situation context, demographics, preferences, past ratings
  - Items
    - Item characteristics, or nothing at all
- Output
  - Relevance score, predicted rating, or ranking

Recommender systems: examples



Your Amazon.com Your Browsing History Recommended For You Amazon Betterizer Improve Your Recommendations Your

NETFLIX

Wesley Fryer ▾ | Your Account | Buy / Redeem Gift | Help

Browse DVDs | Watch Instantly | Your Queue | Movies You'll ♥ | Friends & Community | DVD Sale $5.99

Movies, actors, directors, genres Search

Suggestions (1279) | Suggestions by Genre ▾ | Rate Movies | Rate Genres | Movies You've Rated (398)

Movies You'll Love

Suggestions based on your ratings

You have 1279 Suggestions from 398 ratings.

To Get the Best Suggestions Close

1. Rate your genres.

2. Rate the movies you've seen.

New Suggestions for You

Based

+Alexander Search Images Maps Play YouTube News Gmail Drive Calendar

Google restaurants

Web Images Maps Shopping News Nose Beta More ▾

About 1,190,000,000 results (0.37 seconds)

California Fish Grill Inc
cafishgrill.com/
Score: **24** / 30 · 117 Google reviews

A 3988 Barranca Pkwy
Irvine
(949) 654-3838

Bistango
www.bistango.com/
Zagat: **25** / 30 · 247 Google reviews

B 19100 Von Karman Ave
Irvine
(949) 752-5222

Ruth's Chris Steak House
www.ruthschris.com/
Zagat: **27** / 30 · 75 Google reviews

C 2961 Michelson Dr
Irvine
(949) 252-8848

Stonefire Grill
www.stonefiregrill.com/
Zagat: **23** / 30 · 47 Google reviews

D 3966 Barranca Pkwy
Irvine
(949) 777-1177

by telling us which things you like. This helps us provide

how different items Show my new re

Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance

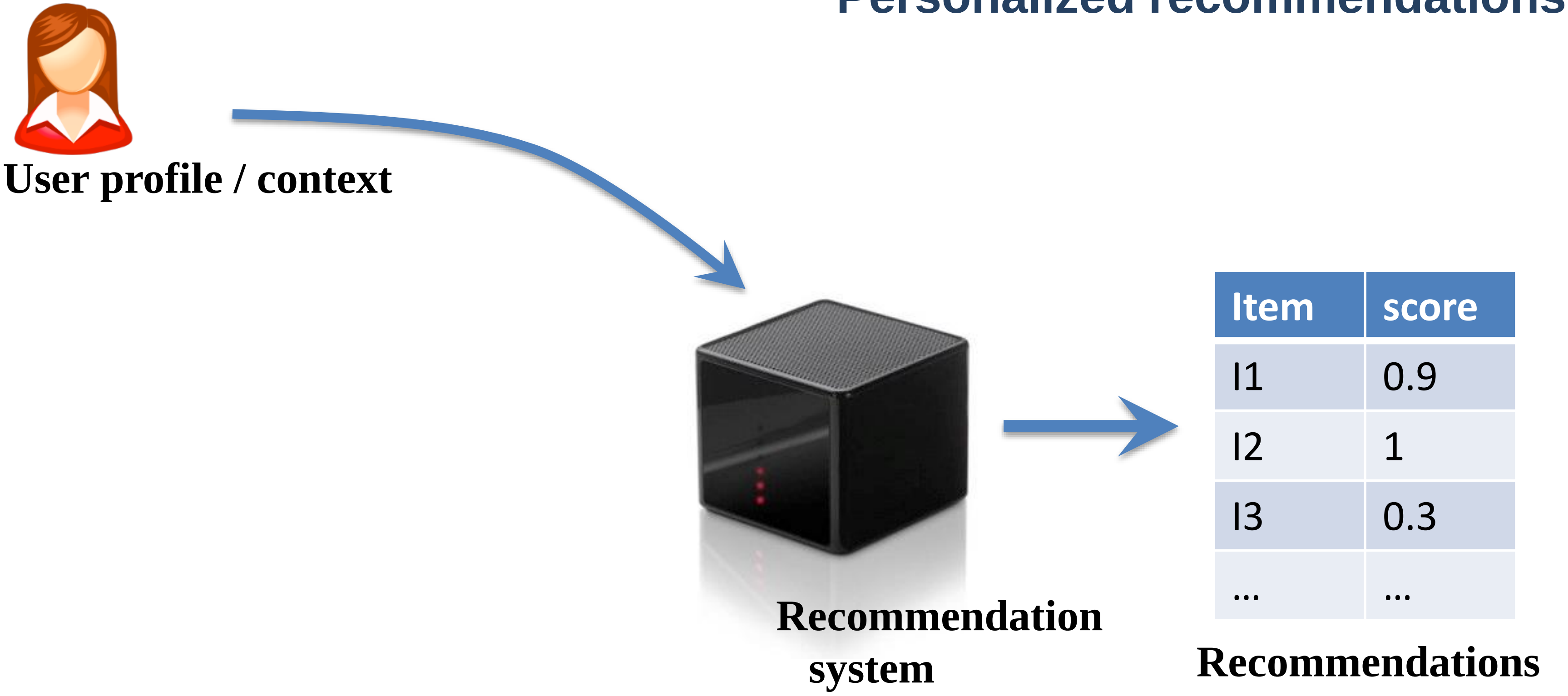
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms of recommender systems

Personalized recommendations



User profile / context

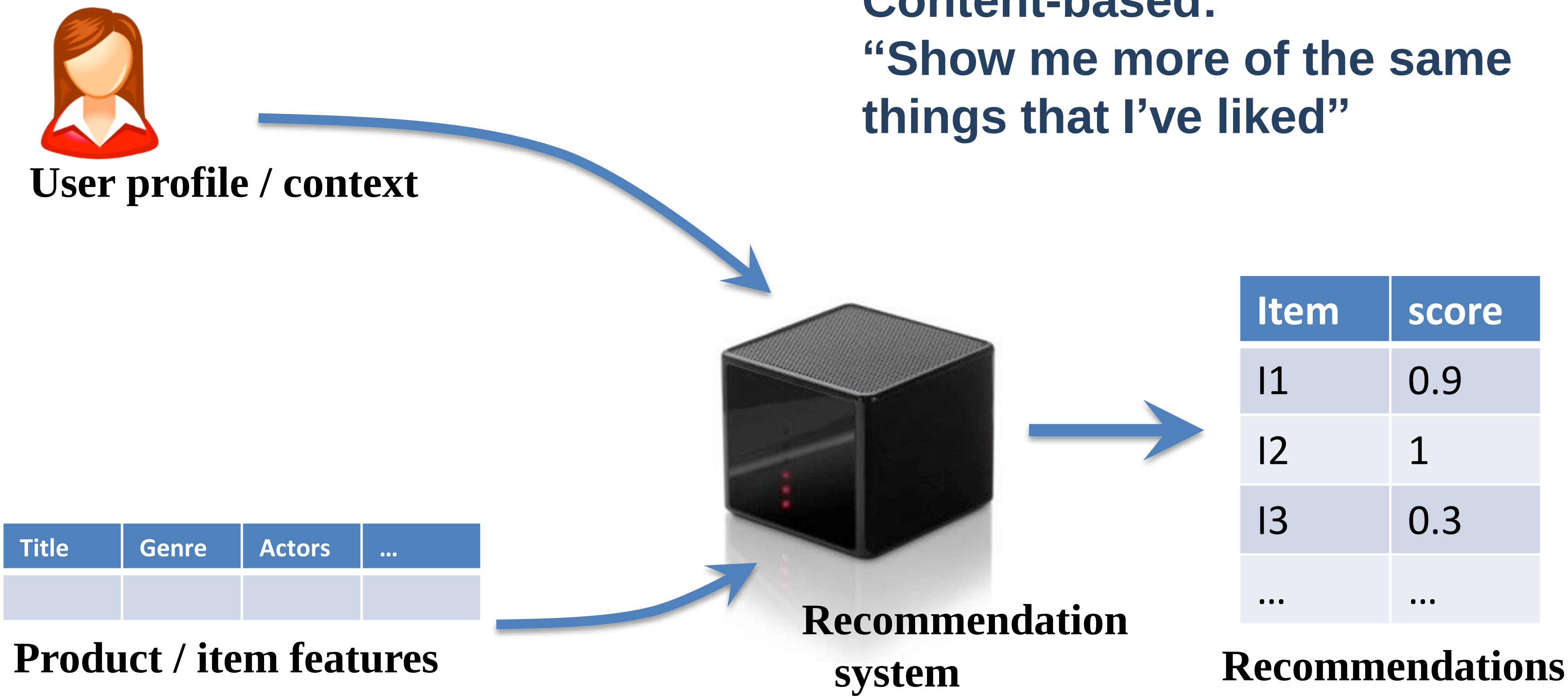
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms of recommender systems

Content-based:
“Show me more of the same things that I’ve liked”



User profile / context

Title	Genre	Actors	…

Product / item features

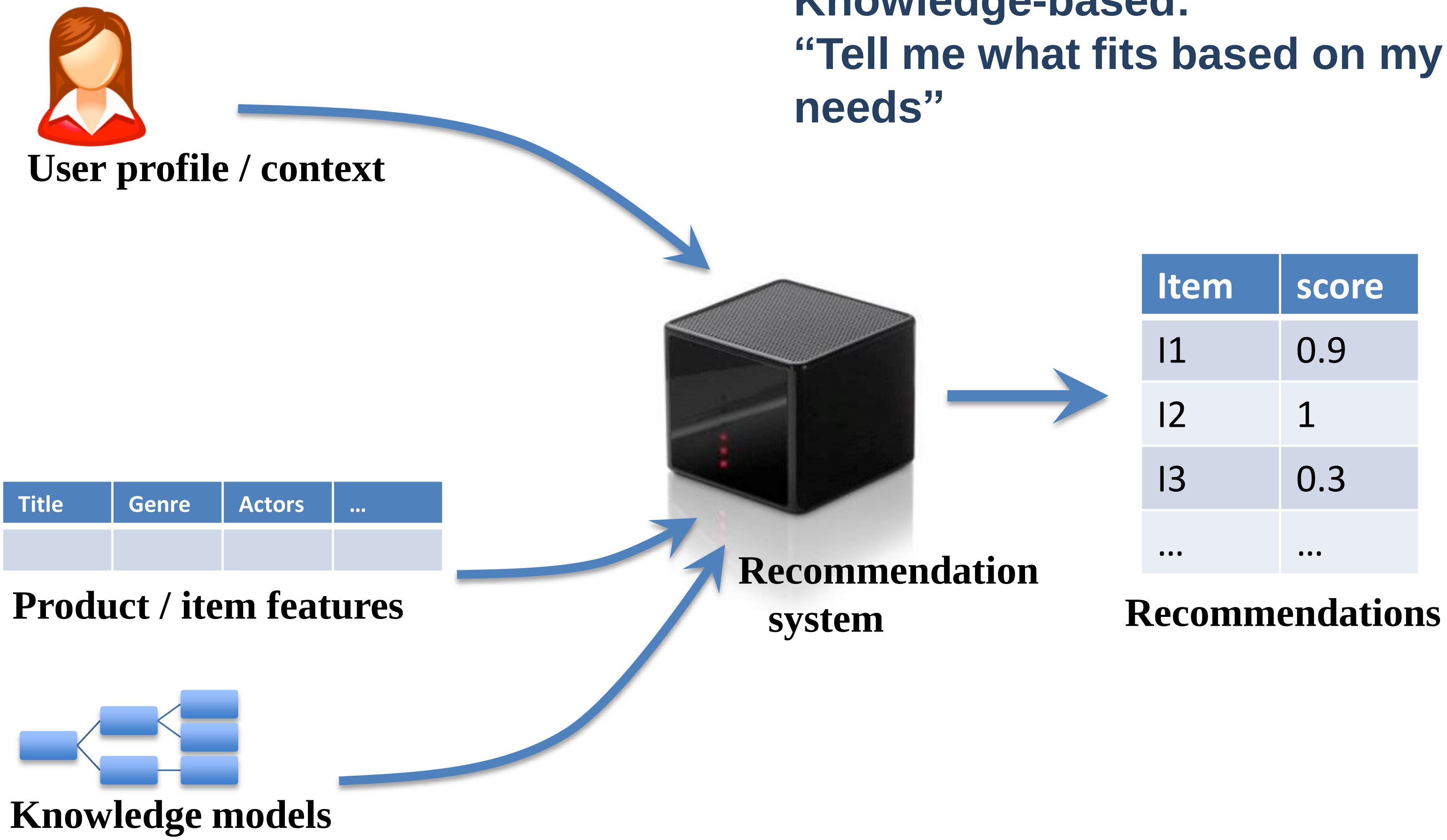
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms of recommender systems

Knowledge-based:
“Tell me what fits based on my needs”



User profile / context

Title	Genre	Actors	…

Product / item features

Knowledge models

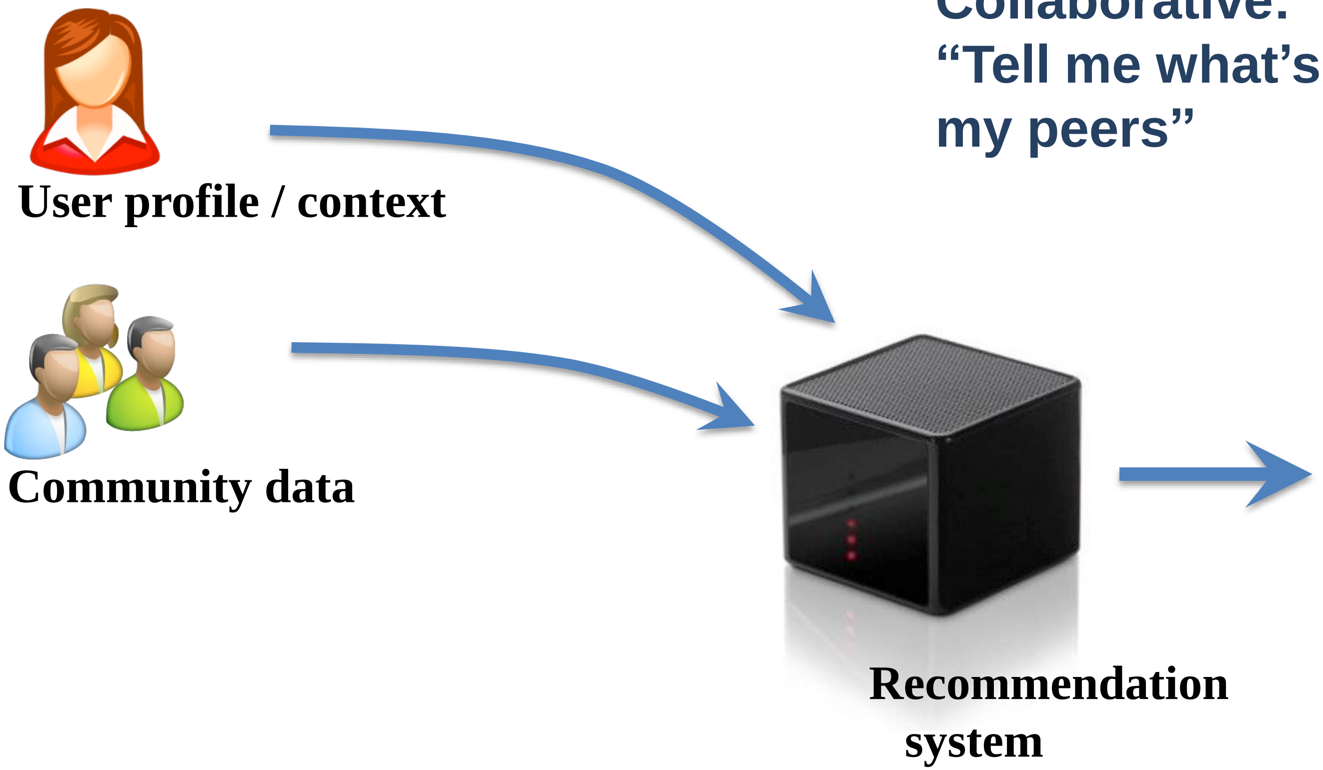
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms of recommender systems

Collaborative:
“Tell me what’s popular among my peers”



User profile / context

Community data

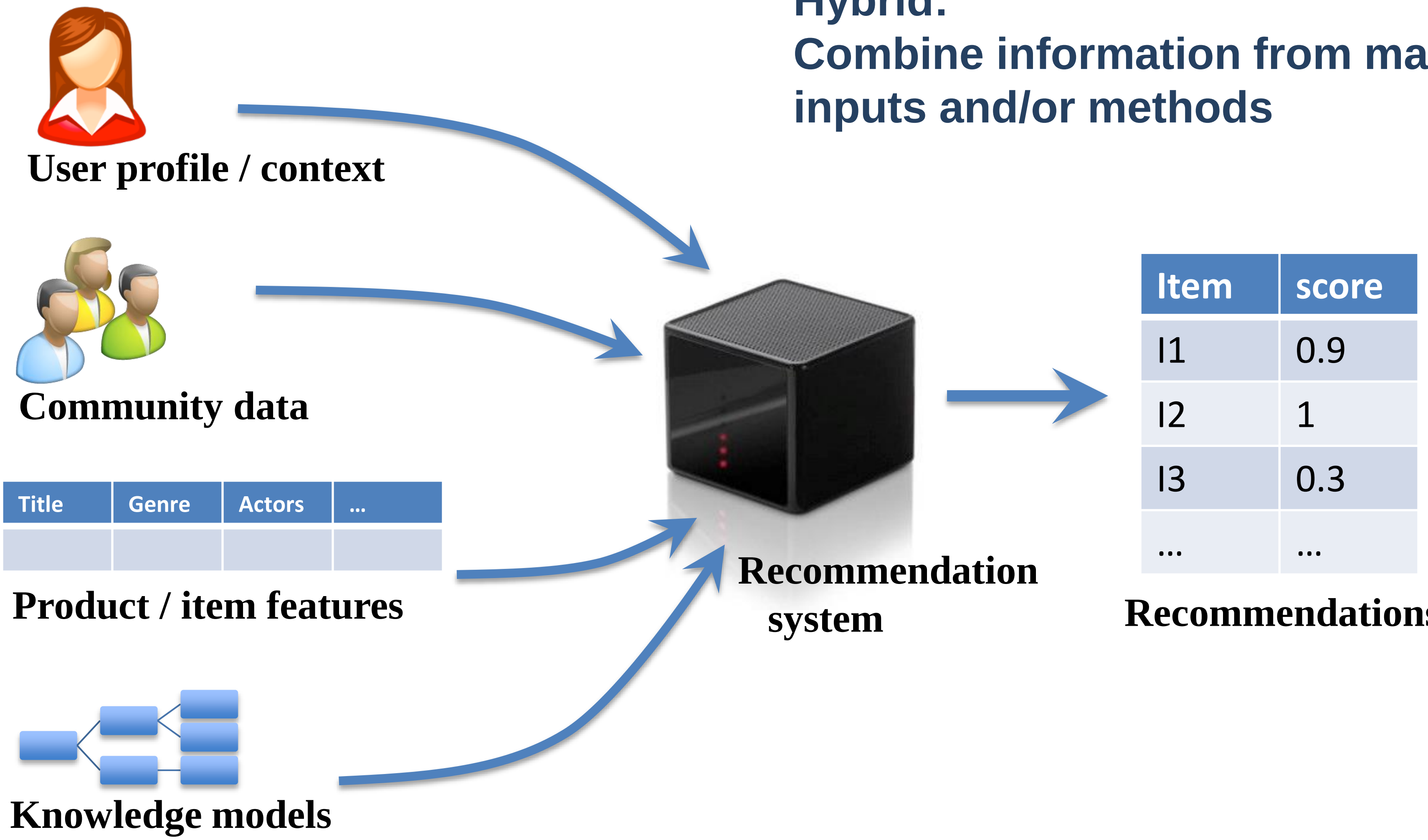
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms of recommender systems

Hybrid:
Combine information from many inputs and/or methods



User profile / context

Community data

Title	Genre	Actors	…

Product / item features

Knowledge models

Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Measuring success

- Prediction perspective
  - Predict to what degree users like the item
  - Most common evaluation for research
  - Regression vs. “top-K” ranking, etc.

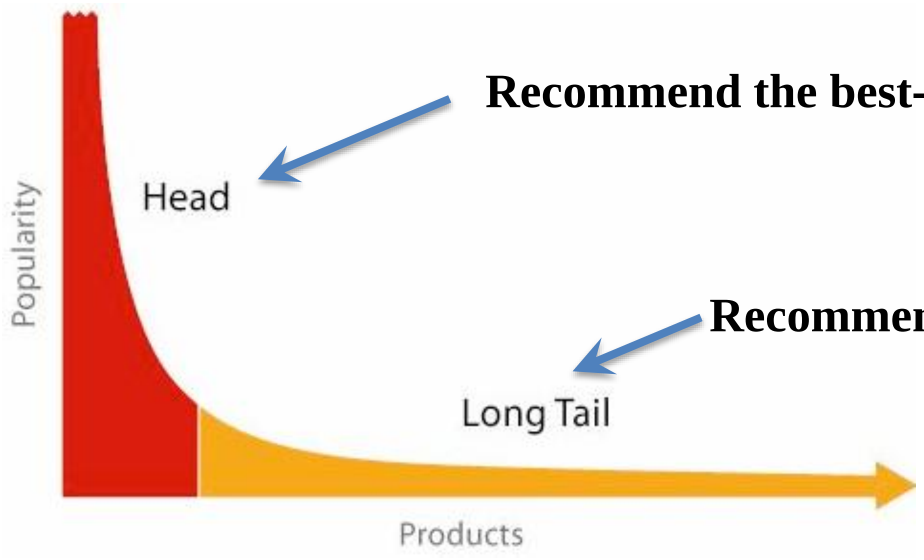
- Interaction perspective
  - Promote positive “feeling” in users (“satisfaction”)
  - Educate about the products
  - Persuade users, provide explanations

- “Conversion” perspective
  - Commercial success
  - Increase “hit”, “click-through” rates
  - Optimize sales and profits

Why are recommenders important?

- The “long tail” of product appeal
  - A few items are very popular
  - Most items are popular only with a few people

- Goal: recommend not-widely known items that the user might like!



Recommend the best-seller list

Recommendations need to be targeted!

Collaborative filtering

users

movies

	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	

Collaborative filtering

- Simple approach: standard regression
  - Use “user features” u~, “item features” i~
  - Train f(u~, i~) ≈ r_{iu}
  - Learn “users with my features like items with these features”

- Extreme case: per-user model / per-item model
- Issues: needs lots of side information!

Features:

1 0 1 0 0 …

0 0 1 0 0 …

…

users

movies

	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	

Collaborative filtering

- Example: nearest neighbor methods
  - Which data are “similar”?

- Nearby items? (based on…)

Features:

1 0 1 0 0 …

0 0 1 0 0 …

…

users

movies

	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	

Collaborative filtering

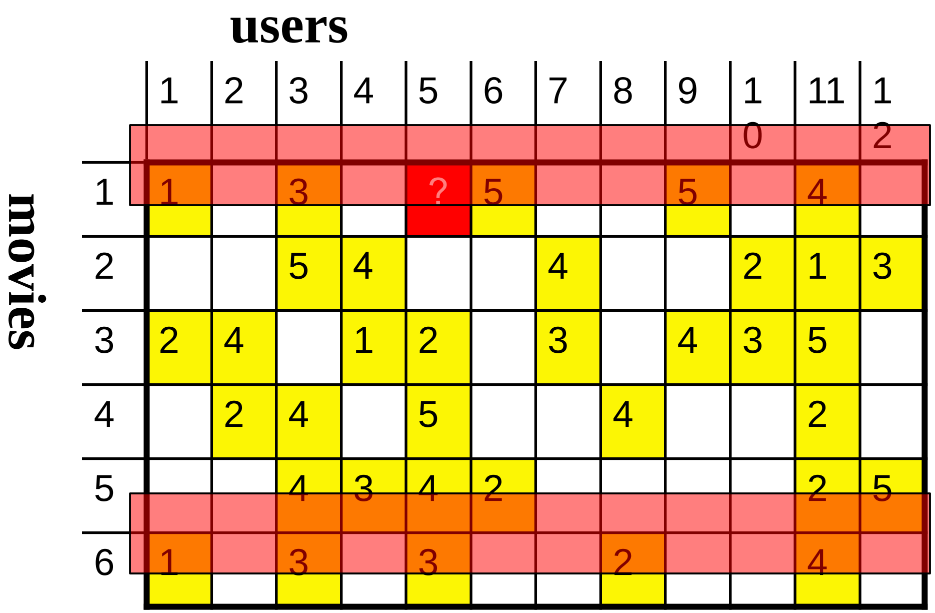
- Example: nearest neighbor methods
  - Which data are “similar”?

- Nearby items? (based on…)

Based on ratings alone?

Find other items that are rated similarly…

Good match on observed ratings



users

movies	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	

Collaborative filtering

- Which data are “similar”?

- Nearby items?
- Nearby users?
  - Based on user features?
  - Based on ratings?

users

movies

	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	

Collaborative filtering

- Some **very simple** examples
  - All users similar, items not similar?
  - All items similar, users not similar?
  - All users and items are equally similar?

users

movies

	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	

Measuring similarity

- Nearest neighbors depends significantly on distance function
  - “Default”: Euclidean distance

- Collaborative filtering:
  - Cosine similarity: $\frac{x^{(i)} \cdot x^{(j)}}{\|x^{(i)}\| \|x^{(j)}\|}$ (measures angle between x^i, x^j)
  - Pearson correlation: measure correlation coefficient between x^i, x^j
  - Often perform better in recommender tasks

- Variant: weighted nearest neighbors
  - Average over neighbors is weighted by their similarity

- Note: with ratings, need to deal with missing data!

Nearest-Neighbor methods

users

movies

	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 movies similar to 1, rated by user 5

Nearest-Neighbor methods

users

movies

	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:

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

Nearest-Neighbor methods

users

movies

	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 weighted average:

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

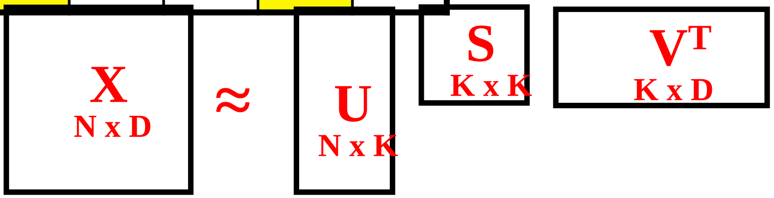
Latent space methods

From Y. Koren
of BellKor team

users

movies

	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	



Latent Space Models

From Y. Koren
of BellKor team

Model ratings matrix as "user" and "movie" positions

Infer values from known ratings

Extrapolate to unranked

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

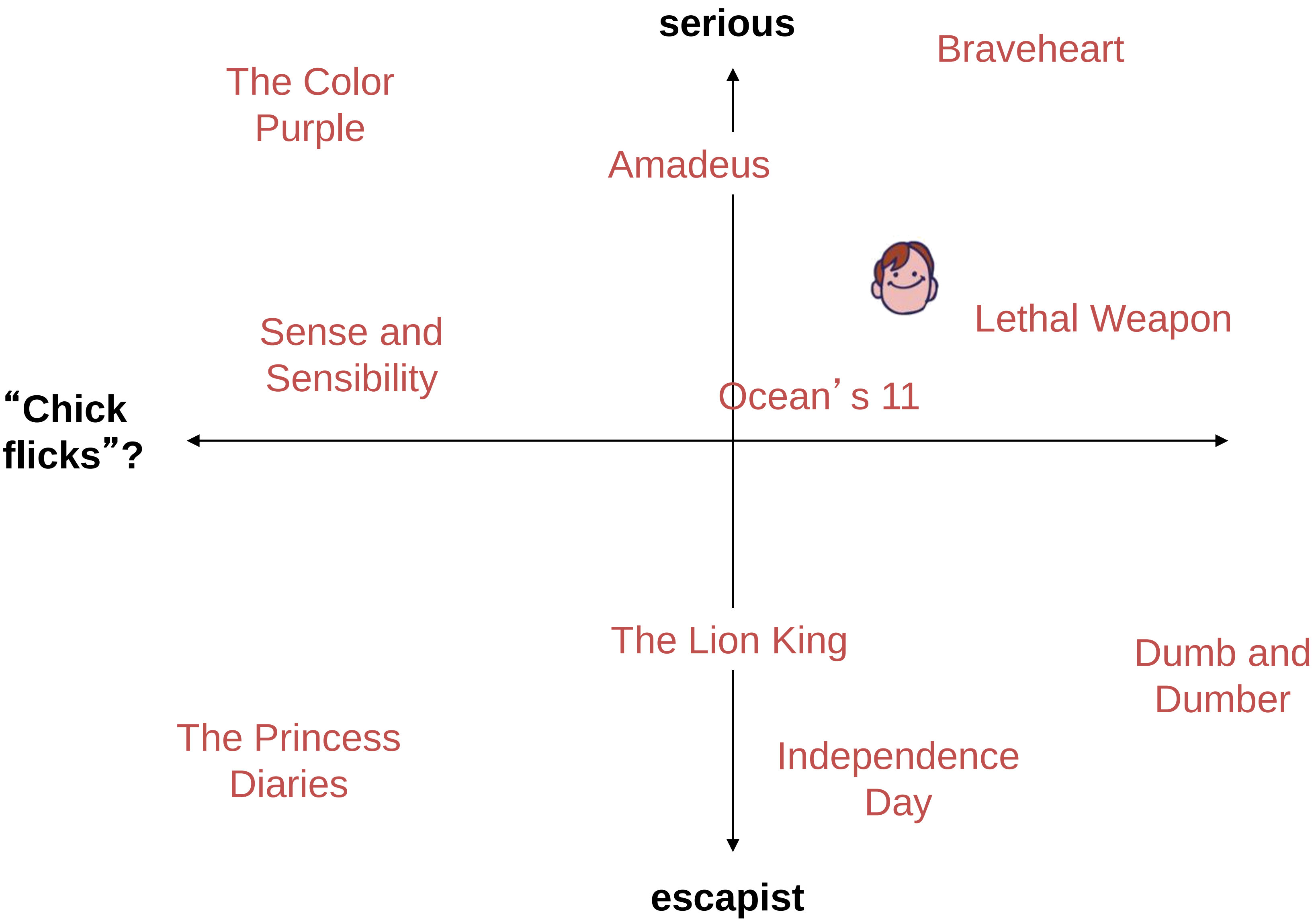
●

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

Latent Space Models

From Y. Koren of BellKor team



serious

Braveheart

The Color Purple

Amadeus

Lethal Weapon

Sense and Sensibility

Ocean's 11

“Chick flicks”?

The Lion King

Dumb and Dumber

The Princess Diaries

Independence Day

escapist

Some SVD dimensions

See timelydevelopment.com

Dimension 1

Offbeat / Dark-Comedy
Lost in Translation
The Royal Tenenbaums
Dogville
Eternal Sunshine of the Spotless Mind
Punch-Drunk Love

Mass-Market / 'Beniffer' Movies
Pearl Harbor
Armageddon
The Wedding Planner
Coyote Ugly
Miss Congeniality

Dimension 2

Good
VeggieTales: Bible Heroes: Lions
The Best of Friends: Season 3
Felicity: Season 2
Friends: Season 4
Friends: Season 5

Twisted
The Saddest Music in the World
Wake Up
I Heart Huckabees
Freddy Got Fingered
House of 1

Dimension 3

What a 10 year old boy would watch
Dragon Ball Z: Vol. 17: Super Saiyan
Battle Athletes Victory: Vol. 4: Spaceward Ho!
Battle Athletes Victory: Vol. 5: No Looking Back
Battle Athletes Victory: Vol. 7: The Last Dance
Battle Athletes Victory: Vol. 2: Doubt and Conflic

What a liberal woman would watch
Fahrenheit 9/11
The Hours
Going Upriver: The Long War of John Kerry
Sex and the City: Season 2
Bowling for Columbine

Latent space models

- Latent representation encodes some “meaning”
- What kind of movie is this? What movies is it similar to?

- Matrix is full of missing data
  - Hard to take SVD directly
  - Typically solve using gradient descent
  - Easy algorithm (see Netflix challenge forum)

$$J(U,V) = \sum_{u,m} (X_{mu} - \sum_k U_{mk} V_{ku})^2$$

```
# for user u, movie m, find the kth eigenvector & coefficient by iterating:
predict_um = U[m,:].dot( V[:,u] )          # predict: vector-vector product
err = ( rating[u,m] – predict_um )         # find error residual
V_ku, U_mk = V[k,u], U[m,k]                # make copies for update
U[m,k] += alpha * err * V_ku               # Update our matrices
V[k,u]  += alpha * err * U_mk              #    (compare to least-squares gradient)
```

Latent space models

- Can be a bit more sophisticated:

  $r_{iu} \approx \mu + b_u + b_i + \sum_k W_{ik} V_{ku}$

  - "Overall average rating"
  - "User effect" + "Item effect"
  - Latent space effects (k indexes latent representation)
  - (Saturating non-linearity?)

- Then, just train some loss, e.g. MSE, with SGD
  - Each (user, item, rating) is one data point
  - E.g. $J=\sum_{iu} (X_{iu} - r_{iu})^2$

Ensembles for recommenders

- Given that we have many possible models:
  - Feature-based regression
  - (Weighted) kNN on items
  - (Weighted) kNN on users
  - Latent space representation

  perhaps we should combine them?

- Use an ensemble average, or a stacked ensemble
  - "Stacked" : train a weighted combination of model predictions