Recommender Systems

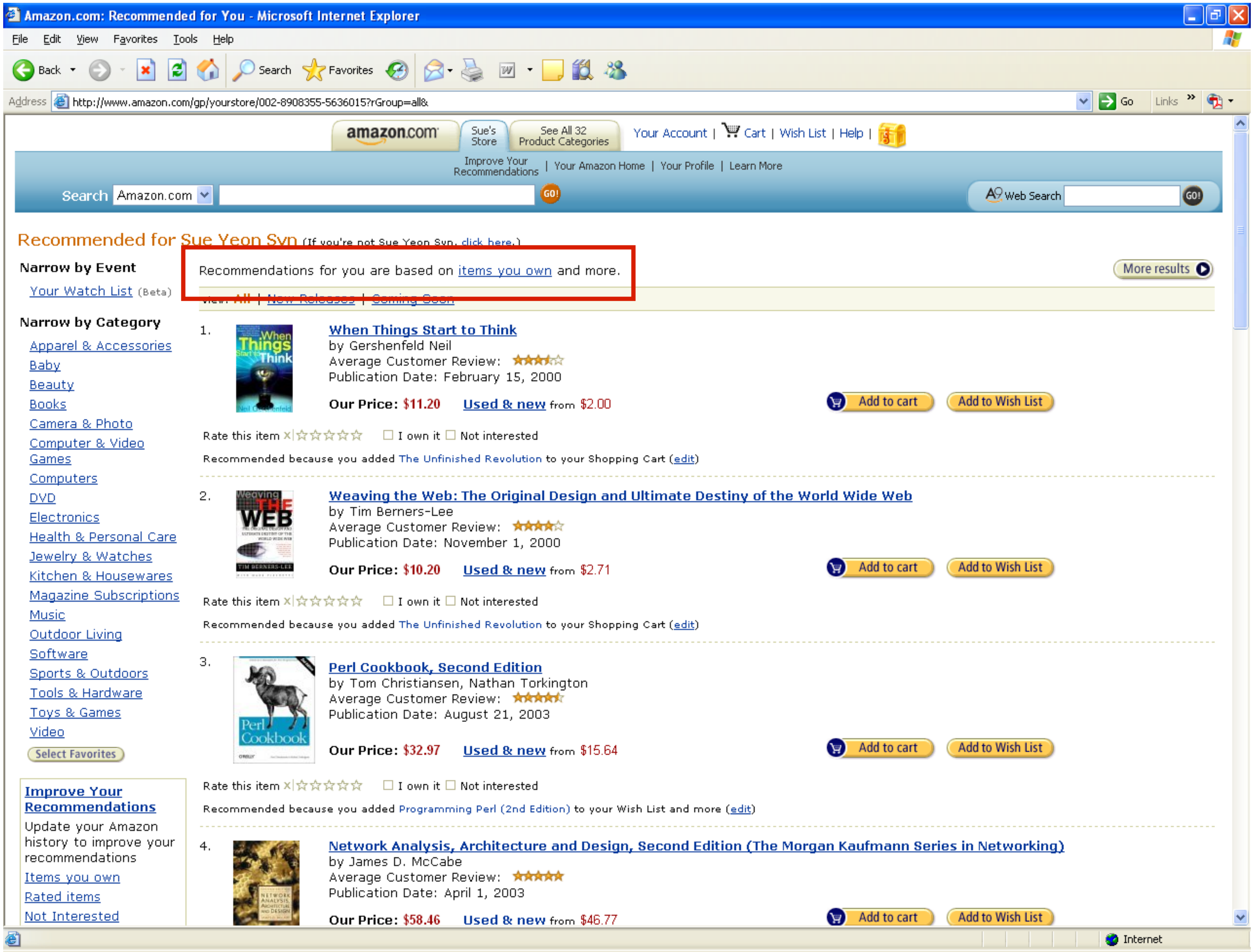
Collaborative Filtering and Matrix Factorization

Narges Razavian

Thanks to lecture slides from Alex Smola@CMU Yahuda Koren@Yahoo labs and Bing Liu@UIC

Recommender systems

Amazon.com



http://www.amazon.com/gp/yourstore/002-8908355-5636015?rGroup=all&

Recommended for Sue Yeon Syn (If you're not Sue Yeon Syn, click here.)

Recommendations for you are based on items you own and more.

View: All | New Releases | Coming Soon

Narrow by Event

Your Watch List (Beta)

Narrow by Category

- Apparel & Accessories
- Baby
- Beauty
- Books
- Camera & Photo
- Computer & Video Games
- Computers
- DVD
- Electronics
- Health & Personal Care
- Jewelry & Watches
- Kitchen & Housewares
- Magazine Subscriptions
- Music
- Outdoor Living
- Software
- Sports & Outdoors
- Tools & Hardware
- Toys & Games
- Video

Improve Your Recommendations

Update your Amazon history to improve your recommendations

Items you own

Rated items

Not Interested

1. **When Things Start to Think**
by Gershenfeld Neil
Average Customer Review: ★★★★☆
Publication Date: February 15, 2000
Our Price: $11.20 Used & new from $2.00
Rate this item ☆☆☆☆☆ ☐ I own it ☐ Not interested
Recommended because you added The Unfinished Revolution to your Shopping Cart (edit)

2. **Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web**
by Tim Berners-Lee
Average Customer Review: ★★★★☆
Publication Date: November 1, 2000
Our Price: $10.20 Used & new from $2.71
Rate this item ☆☆☆☆☆ ☐ I own it ☐ Not interested
Recommended because you added The Unfinished Revolution to your Shopping Cart (edit)

3. **Perl Cookbook, Second Edition**
by Tom Christiansen, Nathan Torkington
Average Customer Review: ★★★★☆
Publication Date: August 21, 2003
Our Price: $32.97 Used & new from $15.64
Rate this item ☆☆☆☆☆ ☐ I own it ☐ Not interested
Recommended because you added Programming Perl (2nd Edition) to your Wish List and more (edit)

4. **Network Analysis, Architecture and Design, Second Edition (The Morgan Kaufmann Series in Networking)**
by James D. McCabe
Average Customer Review: ★★★★★
Publication Date: April 1, 2003
Our Price: $58.46 Used & new from $46.77

An example

Training data

user	movie	score
1	21	1
1	213	5
2	345	4
2	123	4
2	768	3
3	76	5
4	45	4
5	568	1
5	342	2
5	234	2
6	76	5
6	56	4

Test data

user	movie	score
1	62	?
1	96	?
2	7	?
2	3	?
3	47	?
3	15	?
4	41	?
4	28	?
5	93	?
5	74	?
6	69	?
6	83	?

Two basic approaches

- Content-based recommendations:
  - The user will be recommended items based on profile information or similar to the ones the user preferred in the past;
- Collaborative filtering (or collaborative recommendations):
  - The user will be recommended items that people with similar tastes and preferences liked in the past.
- **Hybrids**: Combine collaborative and content-based methods.

Road Map

- Introduction
- **Content-based recommendation**
- Collaborative filtering based recommendation
  - K-nearest neighbor
  - Matrix factorization

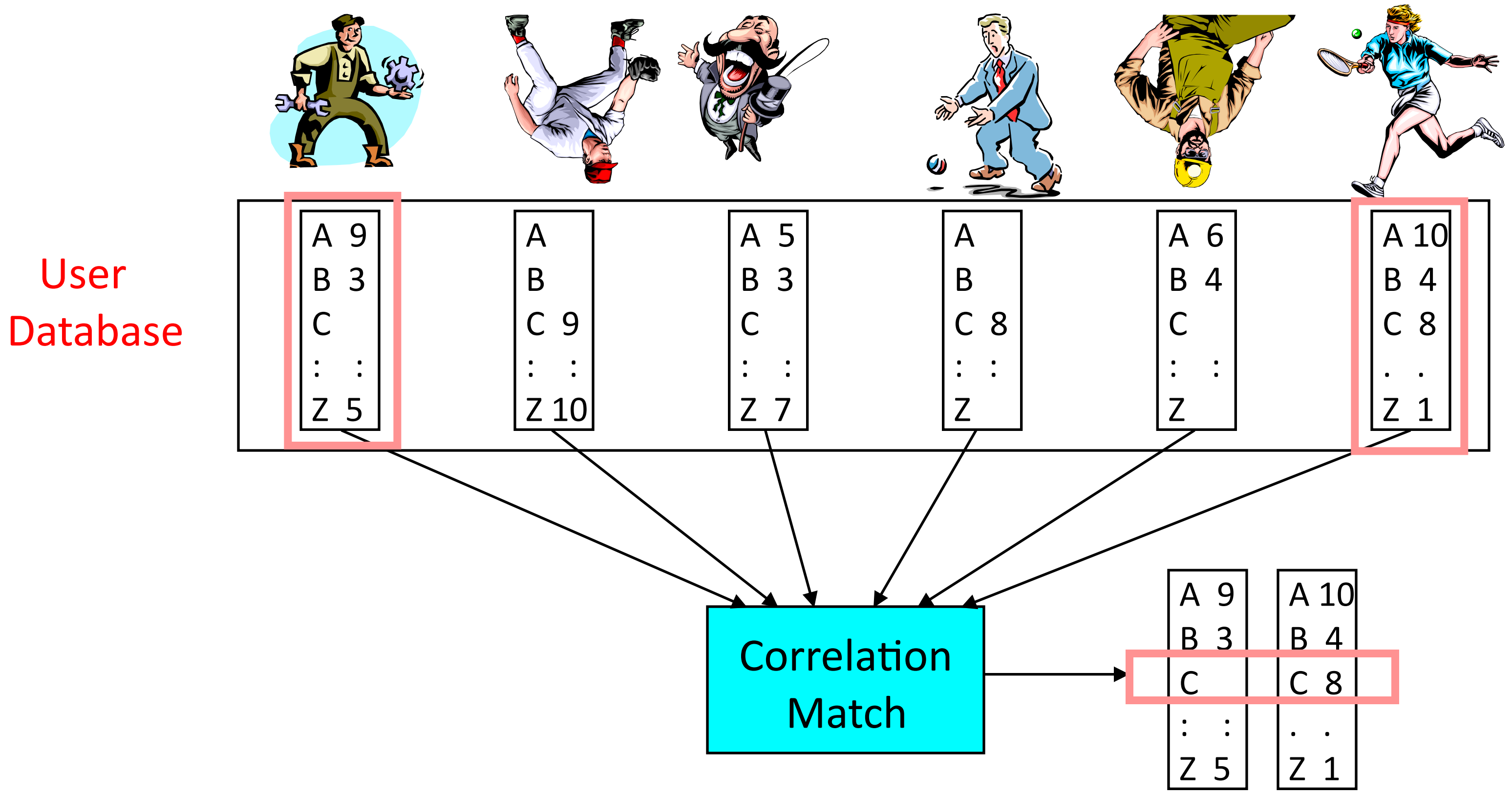
Content-Based Recommendation

- Recommend items that matches the **User Profile.**
- The Profile is based on **items user has liked** in the past or **explicit interests** that he defines.
- A content-based recommender system matches the profile of the item to the user profile to decide on its relevancy to the user.

Road Map

- Introduction
- Content-based recommendation
- Collaborative filtering based recommendations
  - **K-nearest neighbor**
  - Matrix factorization

Collaborative Filtering Idea



User Database

A 9
B 3
C
: :
Z 5

A
B
C 9
: :
Z 10

A 5
B 3
C
: :
Z 7

A
B
C 8
: :
Z

A 6
B 4
C
: :
Z

A 10
B 4
C 8
. .
Z 1

Correlation Match

A 9
B 3
C
: :
Z 5

A 10
B 4
C 8
. .
Z 1

Collaborative filtering

- Collaborative filtering (CF): most widely-used recommendation approach in practice.
  - *k*-nearest neighbor,
  - matrix factorization
- Key characteristic of CF: it predicts the utility of items for a user based on the items previously rated by **other like-minded users**.

k-Nearest Neighbor

- *k*NN :
  - – utilizes the entire user-item database to generate predictions directly, i.e., there is no model building.
- This approach includes both
  - User-based methods
  - Item-based methods
- Two primary phases:
  - the neighborhood formation phase and
  - the recommendation phase.

Neighborhood formation phase

- The similarity between the target user, **u**, and a neighbor, **v**, can be calculated using the **Pearson's correlation coefficient**:

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i \in C}(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

- $r_{u,i}$ is the rating given to item I by user u. C is the list of items rated by BOTH users, u and v

Recommendation Phase

- Then we can compute the rating prediction of **item *i*** for target user **u**

$$p(\mathbf{u},i)=\bar{r}_{\mathbf{u}}+\frac{\sum_{\mathbf{v}\in V} sim(\mathbf{u},\mathbf{v})\times(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v}\in V}\left|sim(\mathbf{u},\mathbf{v})\right|}$$

where V is the set of k similar users(could be all users), $r_{\mathbf{v},i}$ is the rating of user **v** given to item *i*,

Issue with the user-based *k*NN CF

- Lack of scalability:
  - it requires the real-time comparison of the target user to all user records in order to generate predictions.
  - Any suggestions to improve this?

- A variation of this approach that remedies this problem is called **item-based CF**.

Item-based CF

- The item-based approach works by comparing items based on their pattern of ratings across users. The similarity of items *i* and *j* is computed as follows:

$$sim(i,j)=\frac{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})(r_{\mathbf{u},j}-\bar{r}_{\mathbf{u}})}{\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},j}-\bar{r}_{\mathbf{u}})^2}}$$

Recommendation phase

- After computing the similarity between items we select a set of *k* most similar items to the target item and generate a predicted value of user **u**'s rating

$$p(\mathbf{u}, i) = \frac{\sum_{j \in J} r_{\mathbf{u},j} \times sim(i, j)}{\sum_{j \in J} sim(i, j)}$$

where *J* is the set of *k* similar items

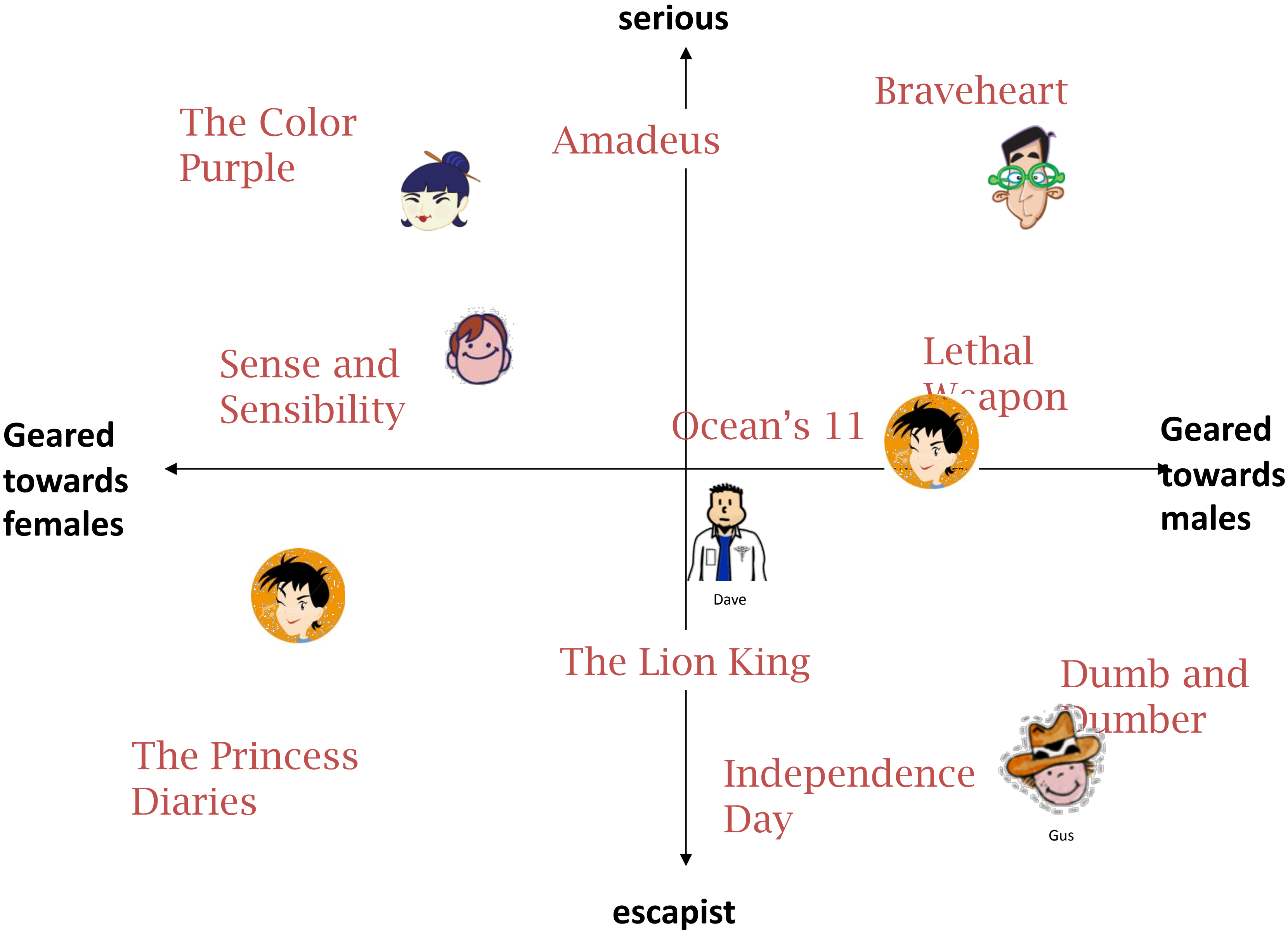
Practical Issues : Cold Start

- New user
  - Rate some initial items
  - Non-personalized recommendations
  - Describe tastes
  - Demographic info.
- New Item
  - Non-CF : content analysis, metadata

Road Map

- Introduction
- Content-based recommendation
- Collaborative filtering based recommendations
  - K-nearest neighbor
  - **Matrix factorization**

Latent factor models



serious

The Color Purple

Amadeus

Braveheart

Sense and Sensibility

Lethal Weapon

Geared towards females

Ocean's 11

Geared towards males

Dave

The Lion King

Dumb and Dumber

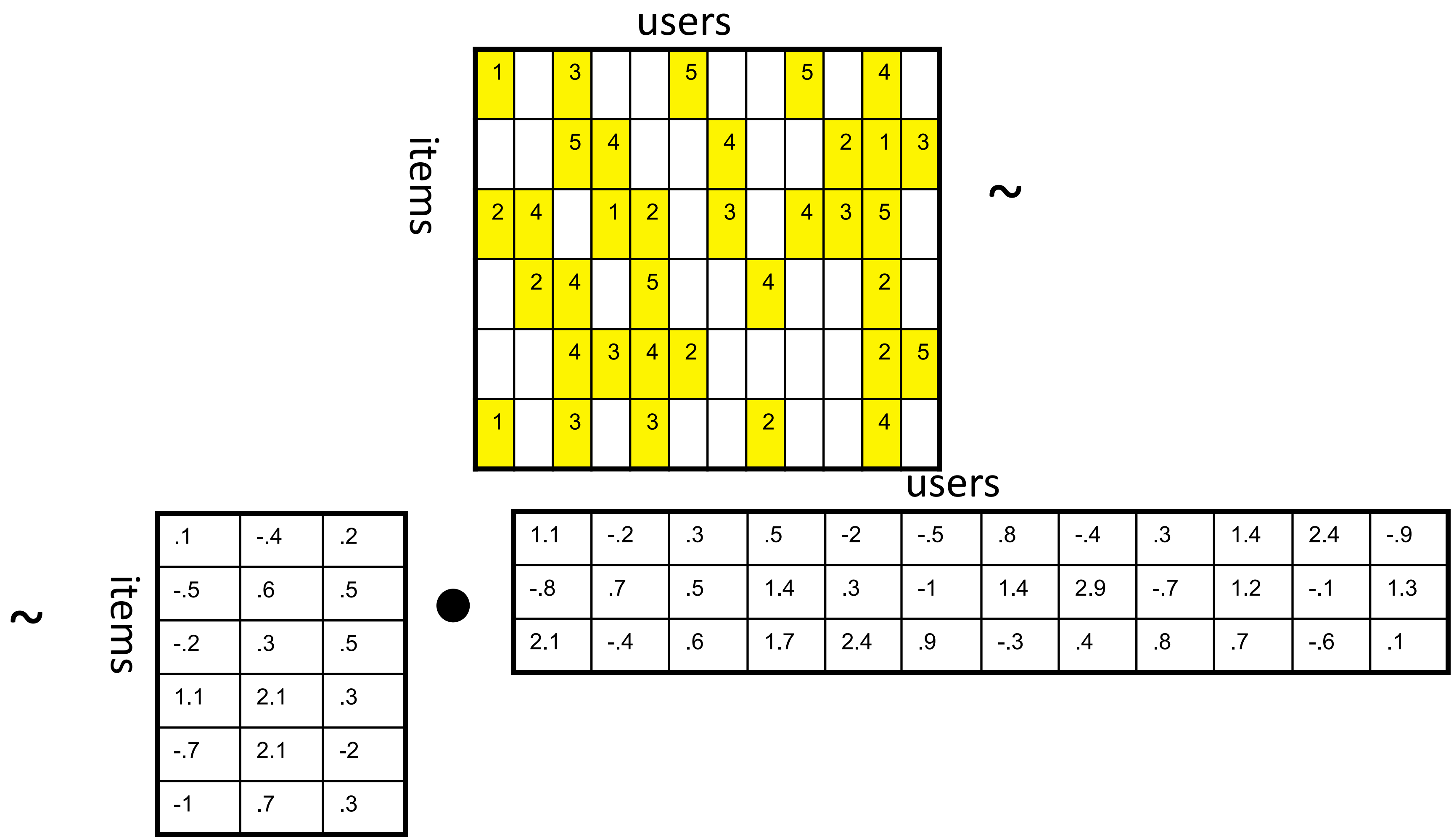
The Princess Diaries

Independence Day

Gus

escapist

Latent factor models



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

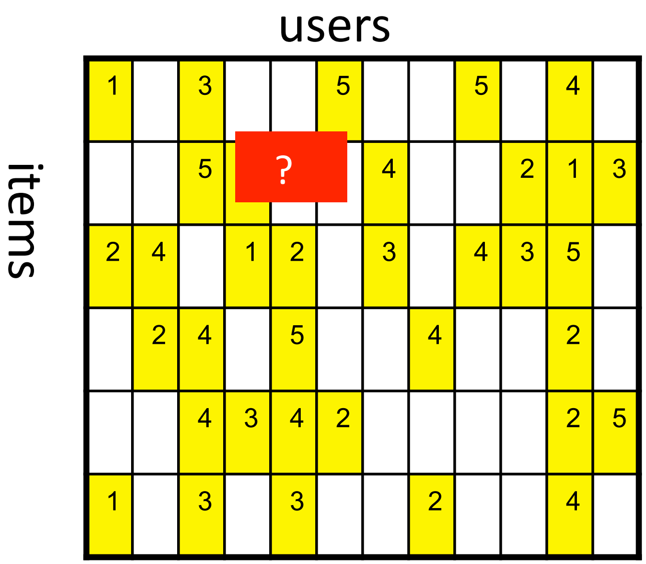
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

●

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

Estimate unknown ratings as inner-products of factors:



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

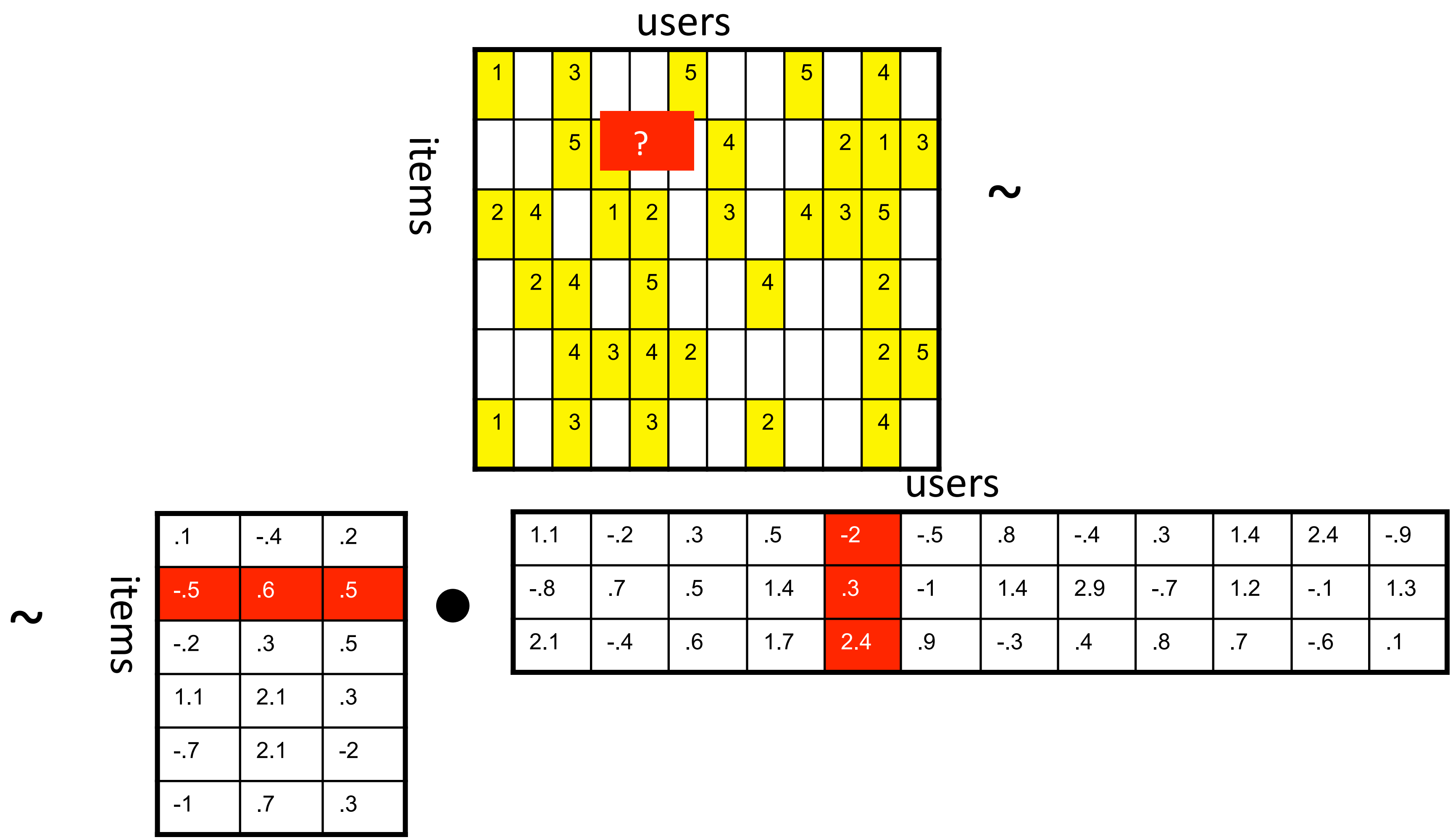
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

●

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

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

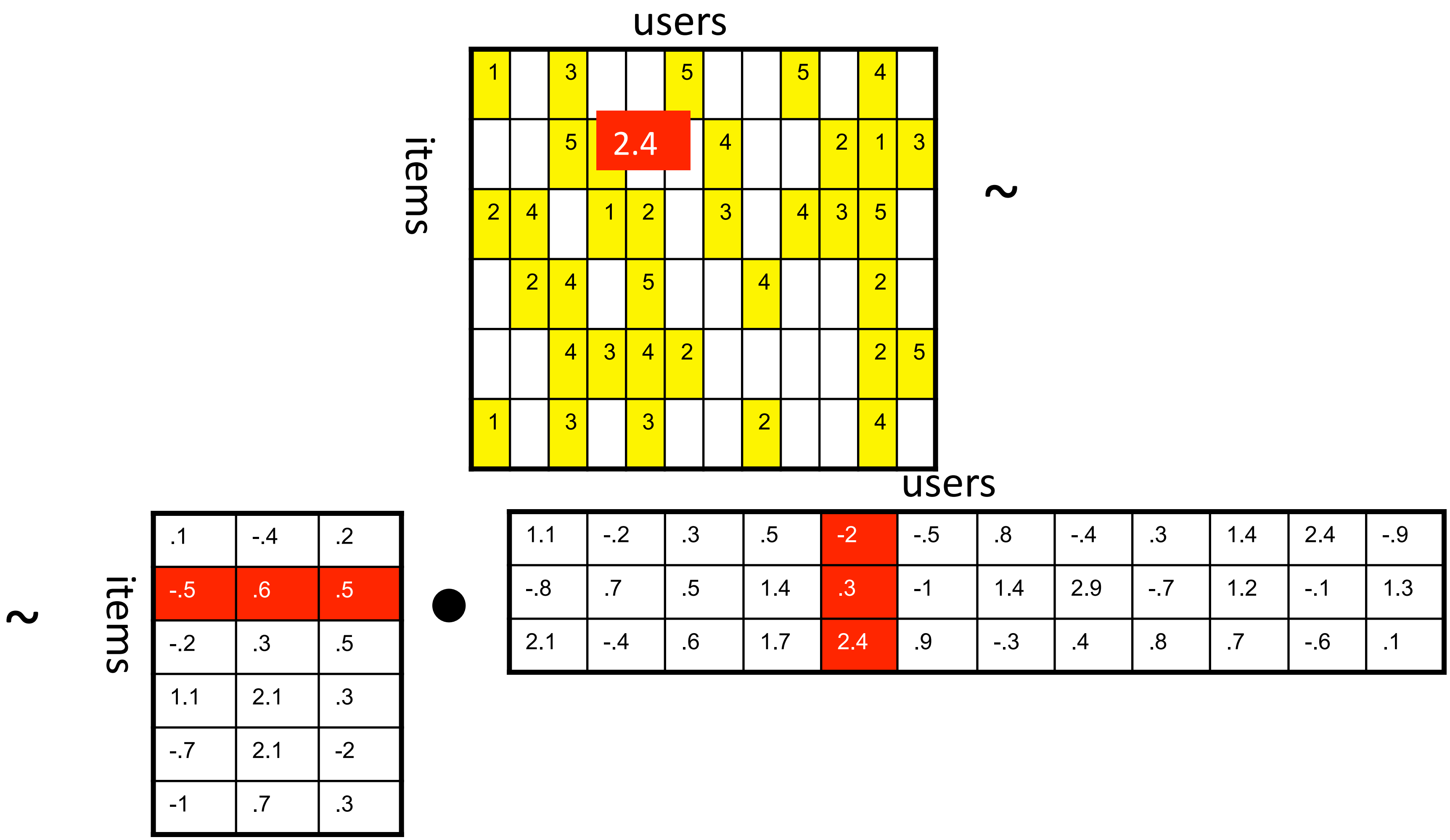
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

●

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

Estimate unknown ratings as inner-products of factors:



users

items

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

.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

Challenges

- Similar to SVD, but less constrained:
  - Factorize with missing values!
- Re-define objective function:

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

- Can use gradient descent to deal with missing values

Stochastic Gradient Descent

- For each data point,

$$e_{ui} = r_{ui} - q_i^T p_u.$$

- Derivatives on variables (q and p) are used for update:

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

- Both p and q are unknown, so we have to alternate
  - Will converge to local optima

Incorporating bias

- Some users rate movies higher than others
- Some movies get hyped and get higher ratings
- The new model: $\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$
- The new objective function

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

- Derivatives:

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

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

Further modeling assumptions

- Changing preferences over time?

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

- Varying confidence levels in ratings?

$$\min_{p_*,q_*,b_*} \sum_{(u,i)\in\kappa} c_{ui}(r_{ui} - \mu - b_u - b_i$$
$$- p_u^T q_i)^2 + \lambda(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2)$$

- Other ideas?

Summary

- Recommendation based on
  - Content
  - Collaborative filtering
- Collaborative filtering
  - Neighborhood method
  - Matrix Factorization
- Possible Further topics
  - Hybrid models of content and collaborative to impute missing values and deal with cold start