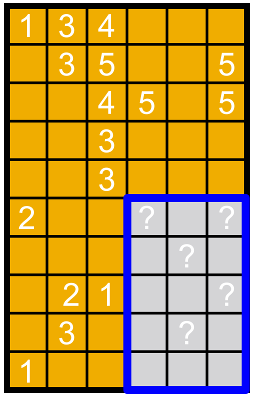
Latent Factor Recommender System

Mining of Massive Datasets

Leskovec, Rajaraman, and Ullman

Stanford University

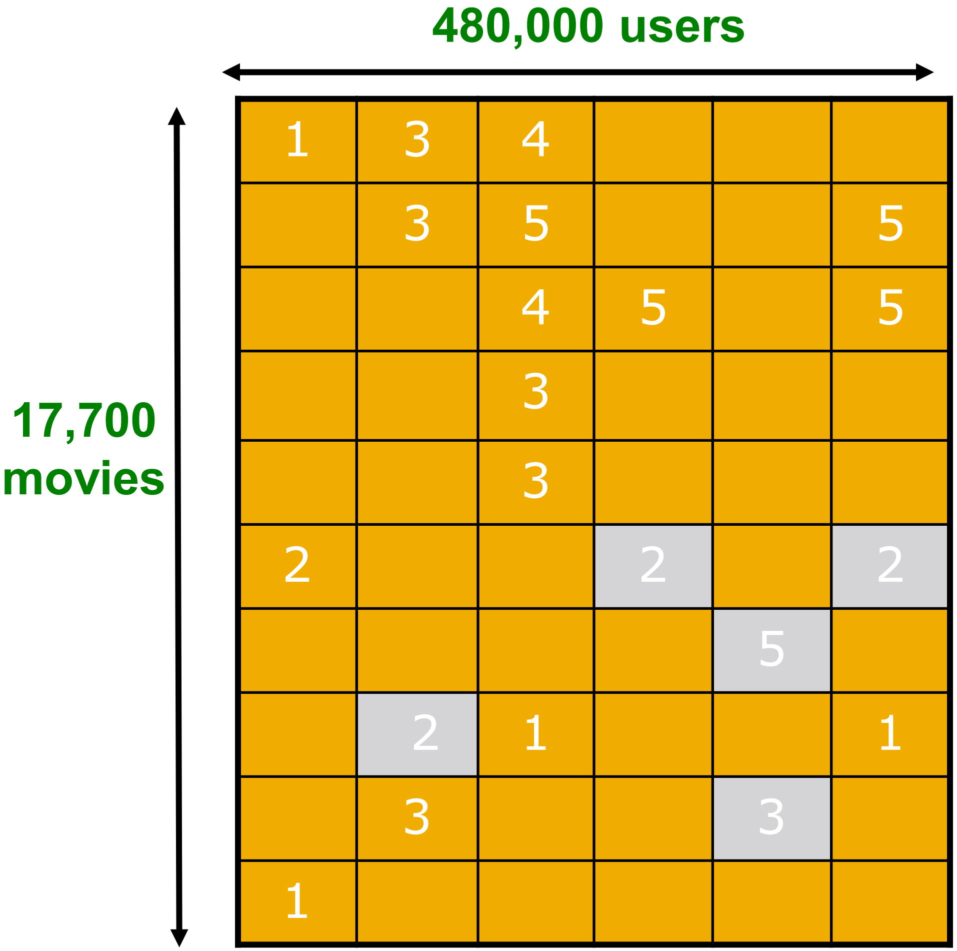
Recommendations via Optimization



1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			?		?
				?	
	2	1			?
	3			?	
1					

- **Goal:** Make good recommendations
  - Quantify goodness using **RMSE:**
    Lower RMSE ⇒ better recommendations
  - Want to make good recommendations on items that user has not yet seen. Can't really do this!
  - **Let's set build a system such that it works well on known (user, item) ratings**
    And **hope** the system will also predict well the **unknown ratings**

The Netflix Utility Matrix *R*



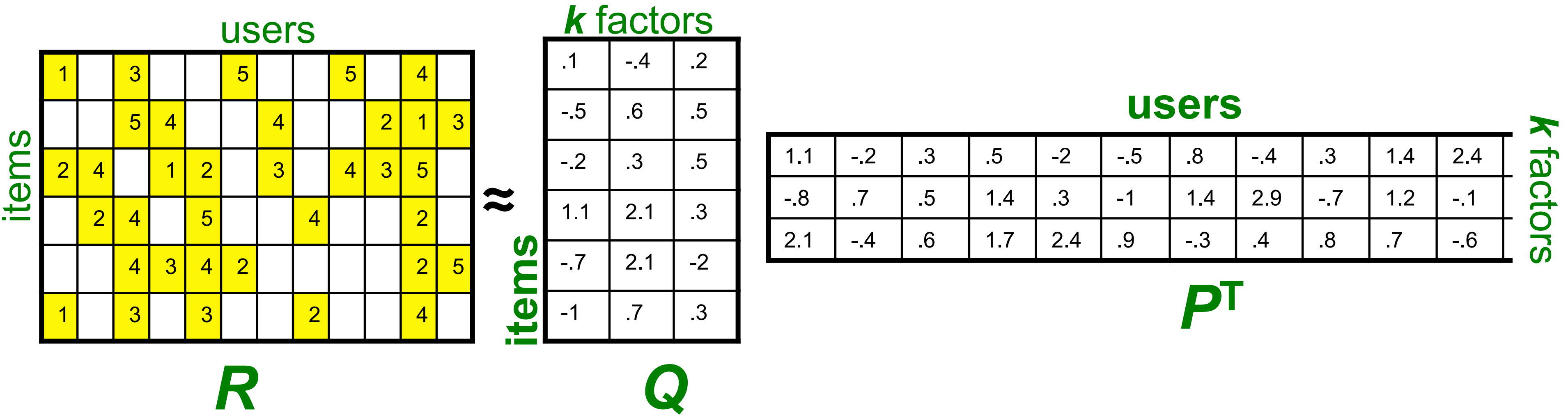
1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			2		2
				5	
	2	1			1
	3			3	
1					

We want our system to predict well the hidden (known) ratings

Latent Factor Models

SVD: $A = U\,\Sigma\,V^T$

- “SVD” on Netflix data: $\mathbf{R} \approx \boldsymbol{Q} \cdot \boldsymbol{P}^T$



R

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	

Q

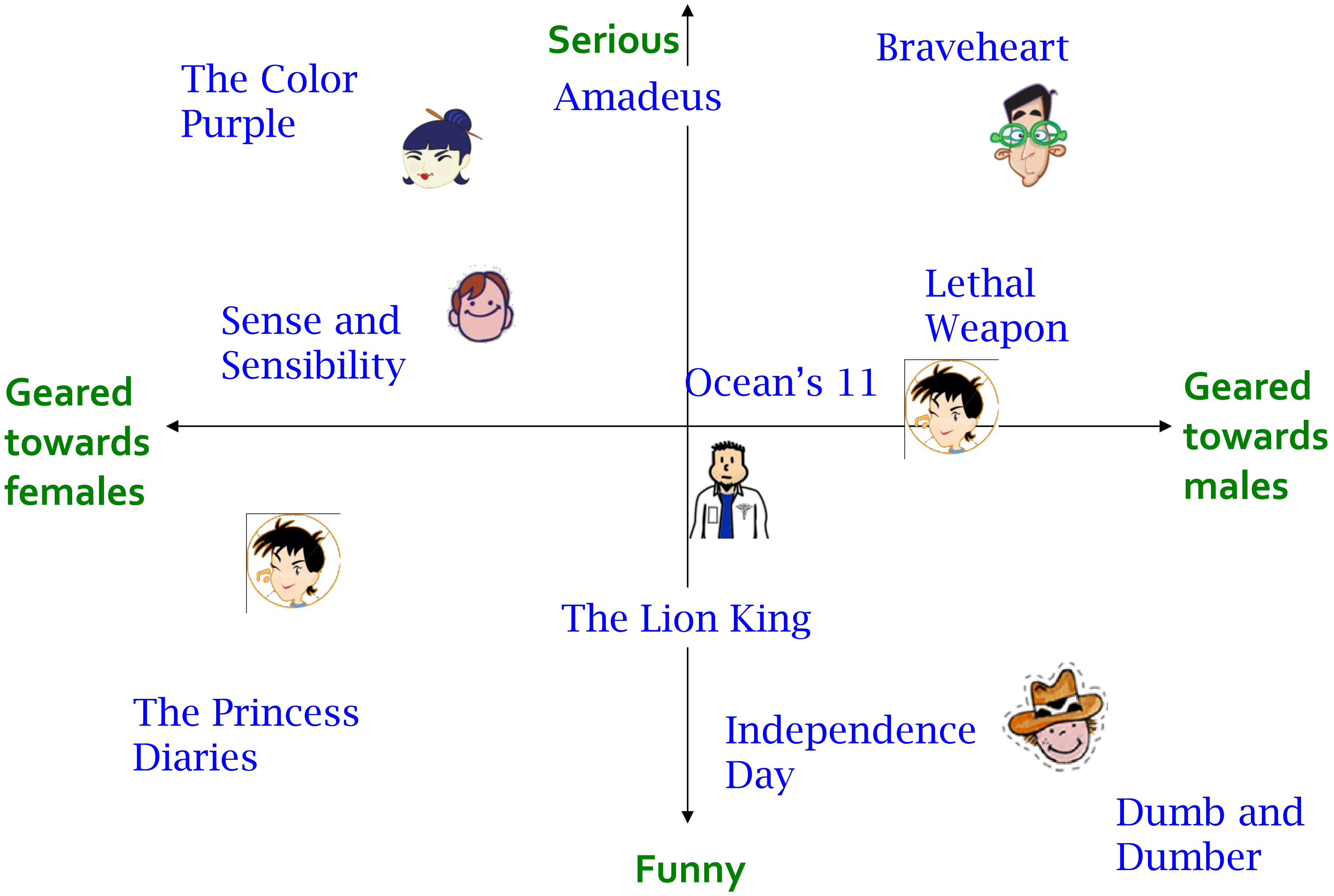
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

P^T

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

- For now let’s assume we can approximate the rating matrix $\boldsymbol{R}$ as a product of “thin” $\boldsymbol{Q} \cdot \boldsymbol{P}^T$
  - ***R*** has missing entries but let’s ignore that for now!
    - Basically, we will want the reconstruction error to be small on known ratings and we don’t care about the values on the missing ones

Latent Factor Models (e.g., SVD)



Serious

The Color Purple

Amadeus

Braveheart

Sense and Sensibility

Lethal Weapon

Geared towards females

Ocean’s 11

Geared towards males

The Lion King

The Princess Diaries

Independence Day

Dumb and Dumber

Funny

Ratings as Products of Factors

- How to estimate the missing rating of user *x* for item *i*?

$$\hat{r}_{xi} = q_i \cdot p_x^T$$

$$= \sum_k q_{ik} \cdot p_{xk}$$

q_i = row i of Q

p_x = column x of P^T

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

k factors

Q

•

users

k factors

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

P^T

Ratings as Products of Factors

- How to estimate the missing rating of user *x* for item *i*?

$$\hat{r}_{xi} = q_i \cdot p_x^T$$

$$= \sum_f q_{if} \cdot p_{xf}$$

q_i = row *i* of *Q*

p_x = column *x* of P^T

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

f factors

Q

•

users

f factors

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

P^T

Ratings as Products of Factors

- **How to estimate the missing rating of user *x* for item *i*?**

$$\hat{r}_{xi} = q_i \cdot p_x^T$$

$$= \sum_f q_{if} \cdot p_{xf}$$

$\boldsymbol{q_i}$ = row ***i*** of ***Q***

$\boldsymbol{p_x}$ = column ***x*** of $\boldsymbol{P}^{\mathrm{T}}$

users

items

1		3			5			5		4	
		5	4	**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

f factors

Q

•

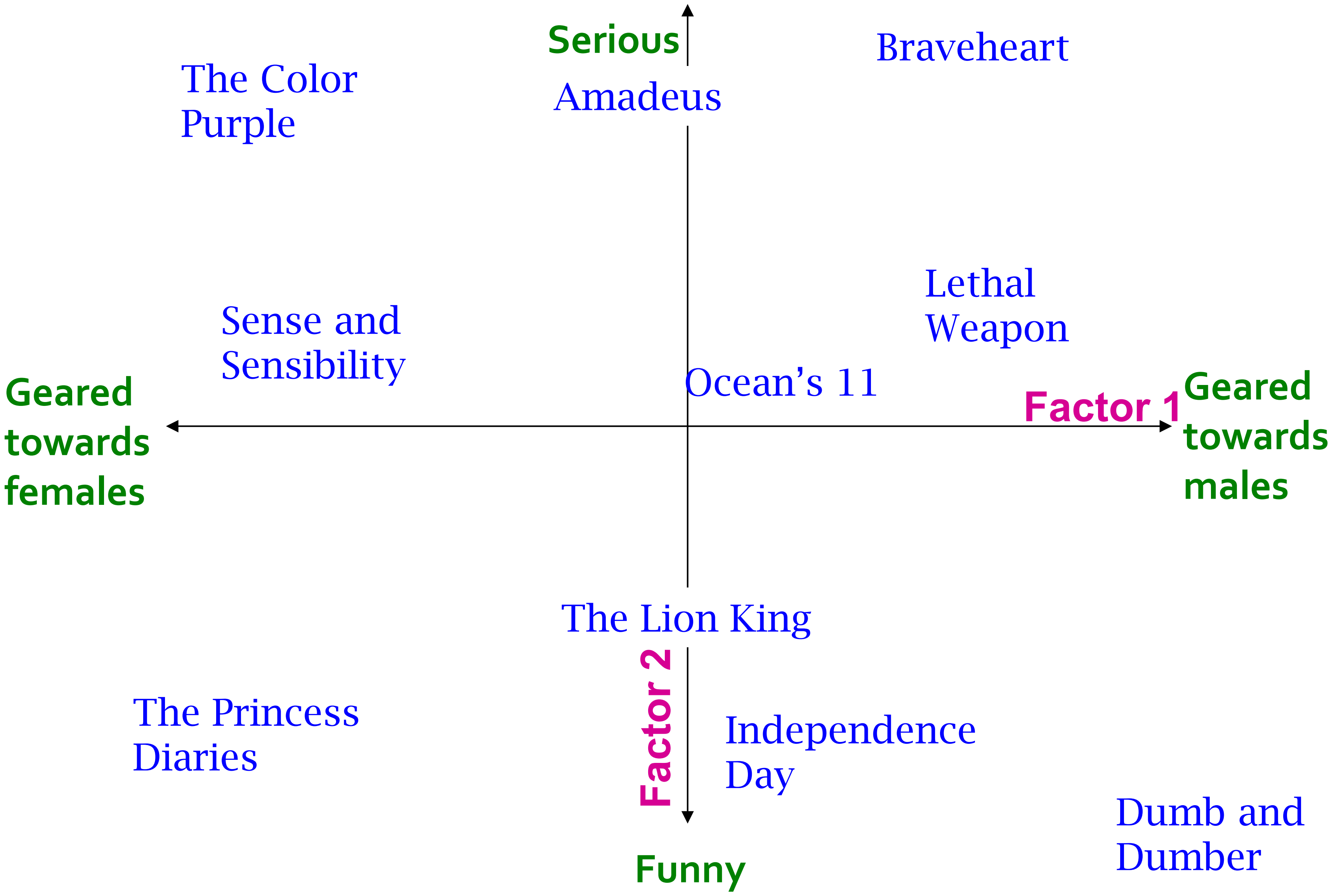
users

f factors

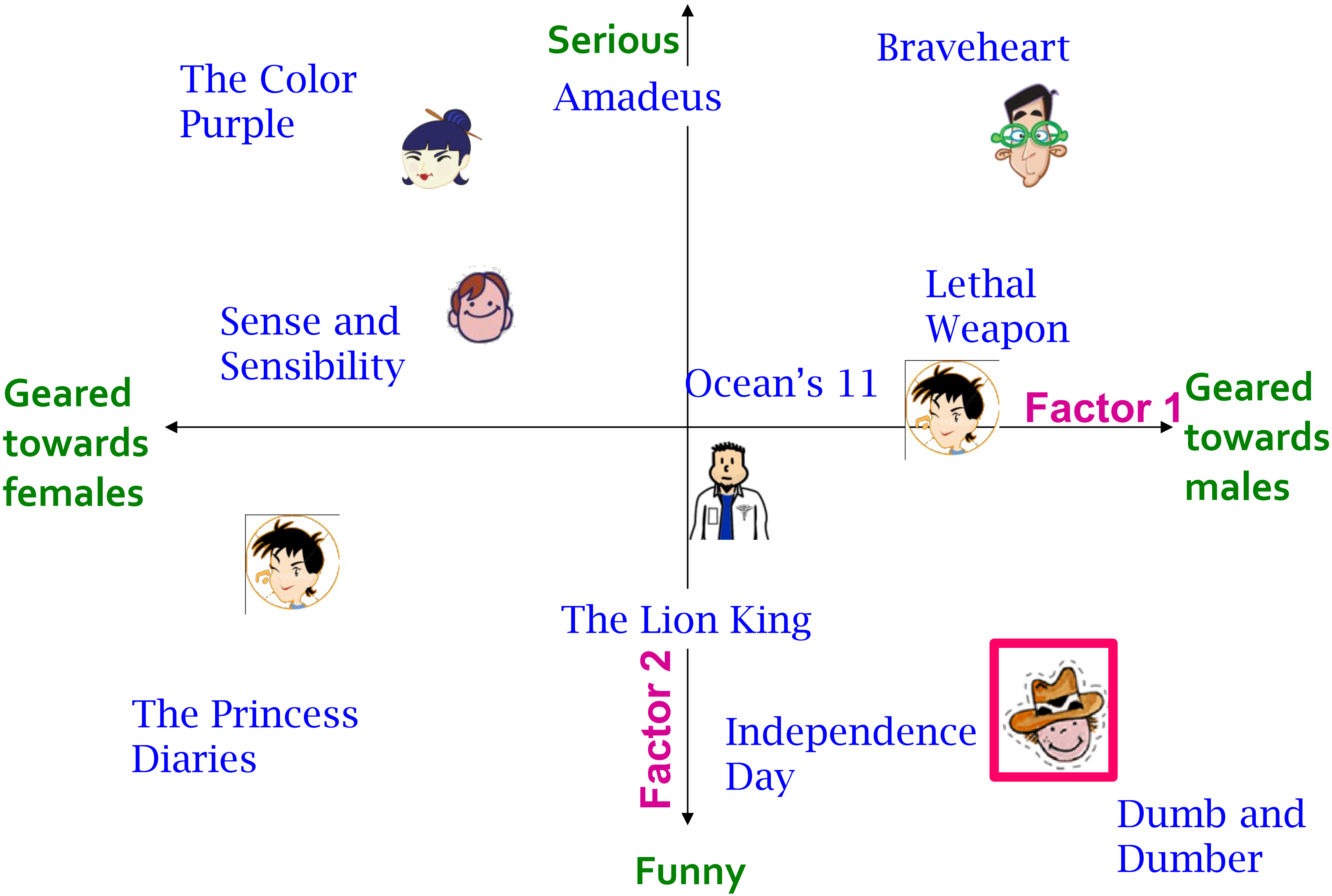
1.1	-.2	.3	.5	**-2**	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	**.3**	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	**2.4**	.9	-.3	.4	.8	.7	-.6	.1

$\boldsymbol{P}^{\mathrm{T}}$

Latent Factor Models



Latent Factor Models



Serious

The Color Purple

Amadeus

Braveheart

Lethal Weapon

Sense and Sensibility

Ocean’s 11

Geared towards females

Factor 1

Geared towards males

The Lion King

Factor 2

The Princess Diaries

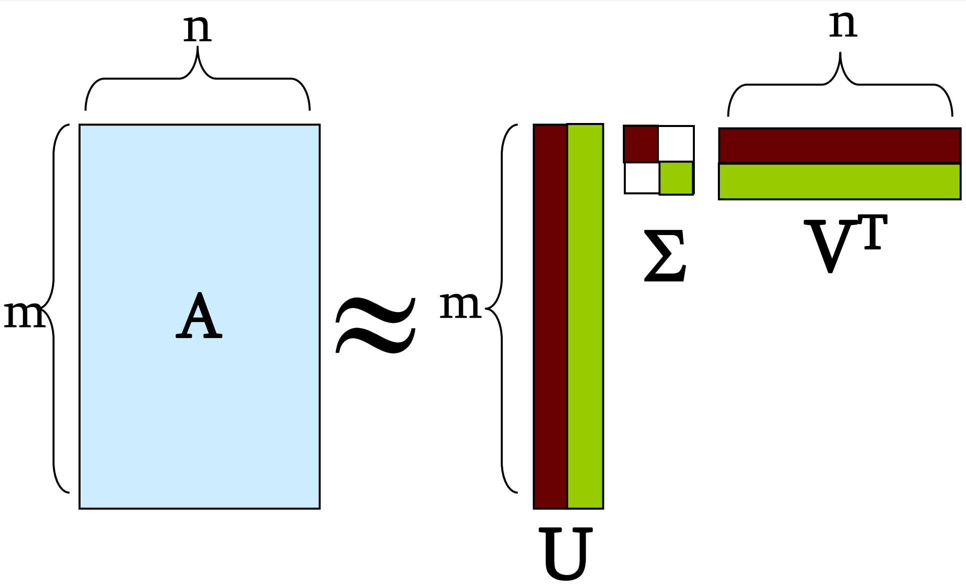
Independence Day

Dumb and Dumber

Funny

Recap: SVD

- **Remember SVD:**
  - **A**: Input data matrix
  - **U**: Left singular vecs
  - **V**: Right singular vecs
  - **Σ**: Singular values



- So in our case:
  "SVD" on Netflix data: $R \approx Q \cdot P^T$
  $A = R$, $Q = U$, $P^T = \Sigma\, V^T$

$$\hat{r}_{xi} = q_i \cdot p_x^T$$

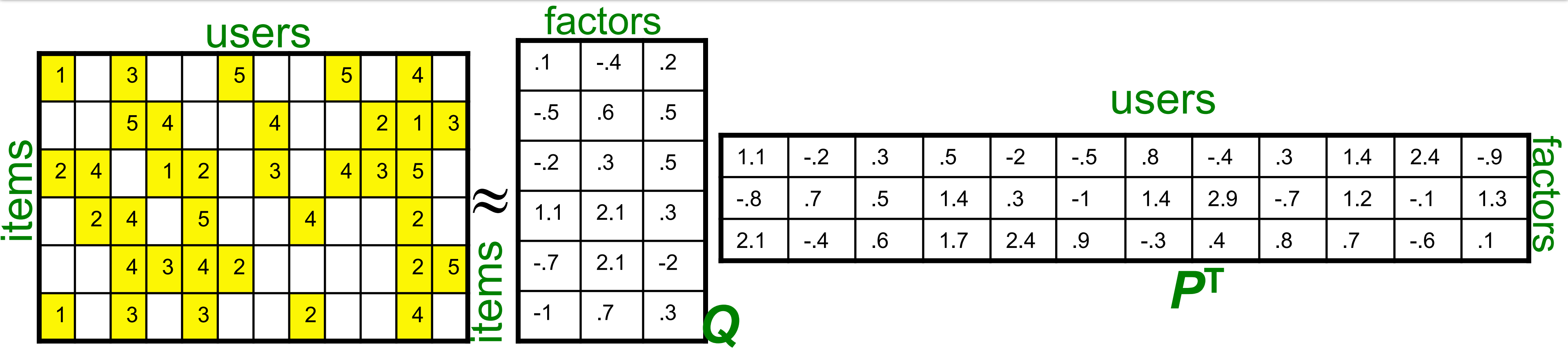
SVD: More good stuff

- We already know that SVD gives minimum reconstruction error **(Sum of Squared Errors):**

$$\min_{U,\mathrm{V},\Sigma} \sum_{ij\in A} \left(A_{ij} - [U\Sigma V^{\mathrm{T}}]_{ij}\right)^2$$

- Note two things:
  - **SSE** and **RMSE** are monotonically related:
    - $\boldsymbol{RMSE = \frac{1}{c}\sqrt{SSE}}$ Great news: SVD is minimizing RMSE
  - **Complication:** The sum in SVD error term is over all entries (no-rating in interpreted as zero-rating). But our ***R*** has missing entries!

Latent Factor Models



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 isn't defined when entries are missing!**
- **Use specialized methods to find *P, Q:***

$$\min_{P,Q} \sum_{(i,x)\in R} \left(r_{xi} - q_i \cdot p_x^T\right)^2$$

- **Note:**
  - We don't require cols of ***P, Q*** to be orthogonal/unit length
  - ***P, Q*** map users/movies to a latent space
  - The most popular model among Netflix contestants