Introduction to Data Science

BRIAN D'ALESSANDRO

FALL 2018

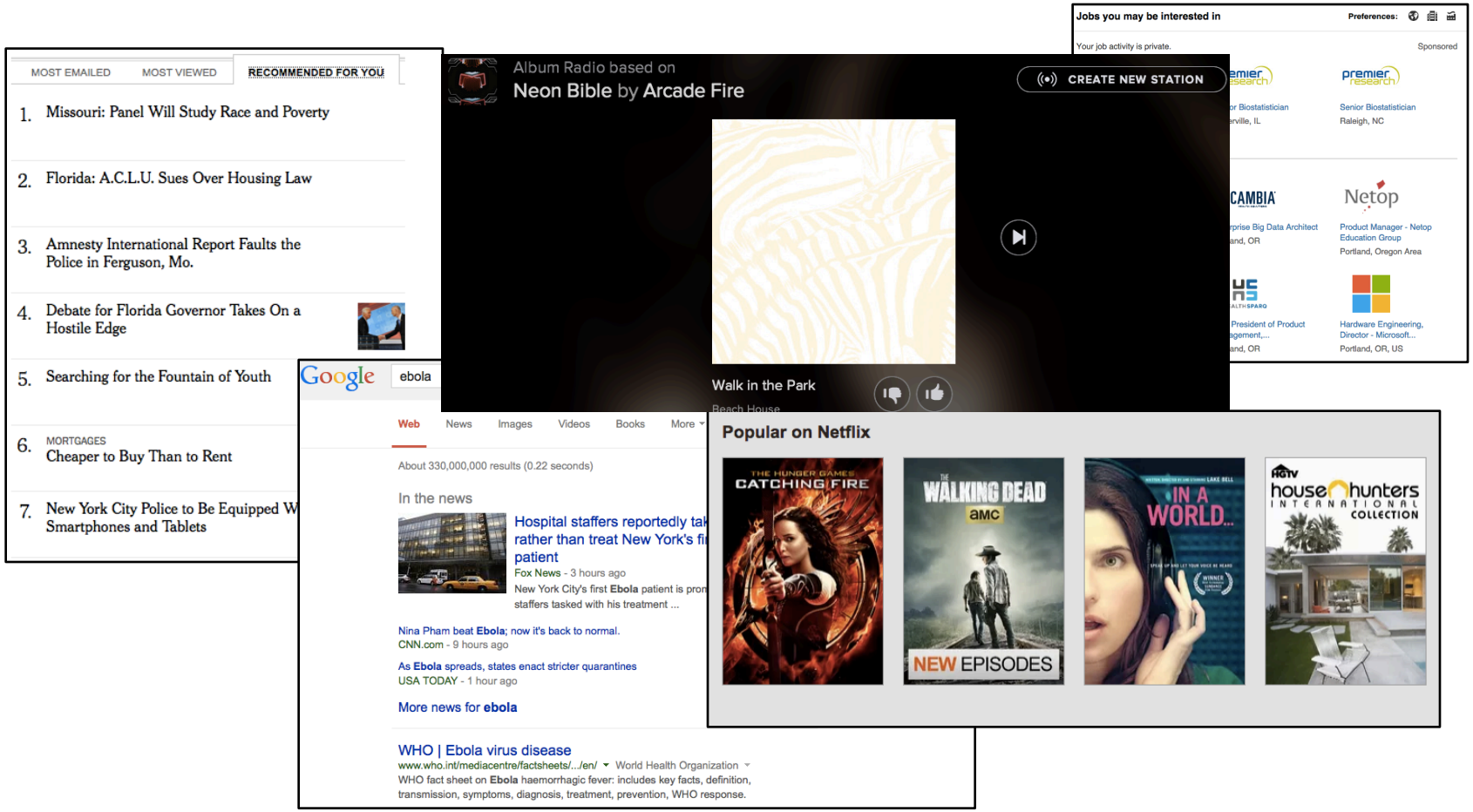
Fine Print: these slides are, and always will be a work in progress. The material presented herein is original, inspired, or borrowed from others' work. Where possible, attribution and acknowledgement will be made to content's original source. Do not distribute, except for as needed as a pedagogical tool in the subject of Data Science.

RECOMMENDATIONS EVERYWHERE

RECOMMENDATIONS ARE EVERYWHERE

If you use the internet, you likely suffer from this little problem - *too much information and too little time.*

Most companies try to solve this problem for you using data science

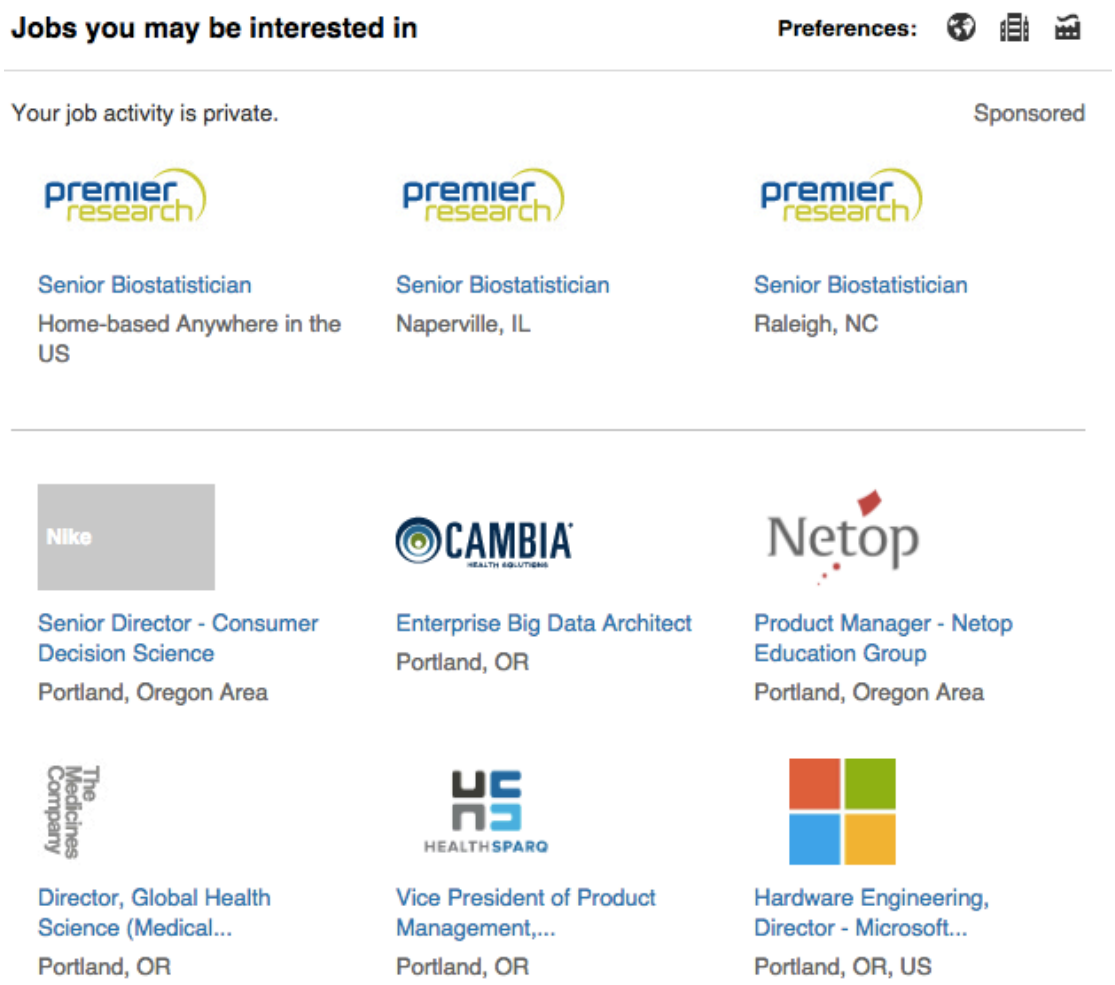


RECOMMENDATIONS DRIVE

Who gets to be your friend…

RECOMMENDATIONS DRIVE

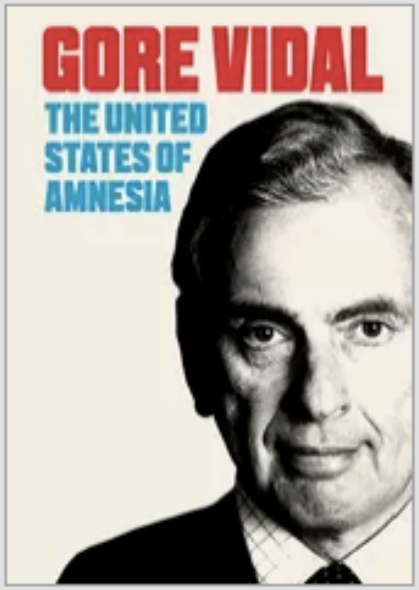
Where you might work …

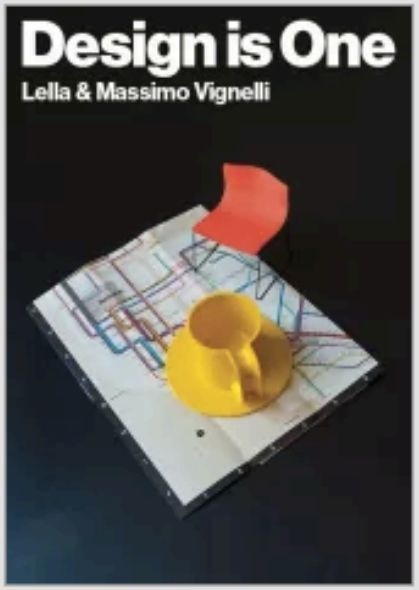


RECOMMENDATIONS DRIVE

What you'll watch tonight…

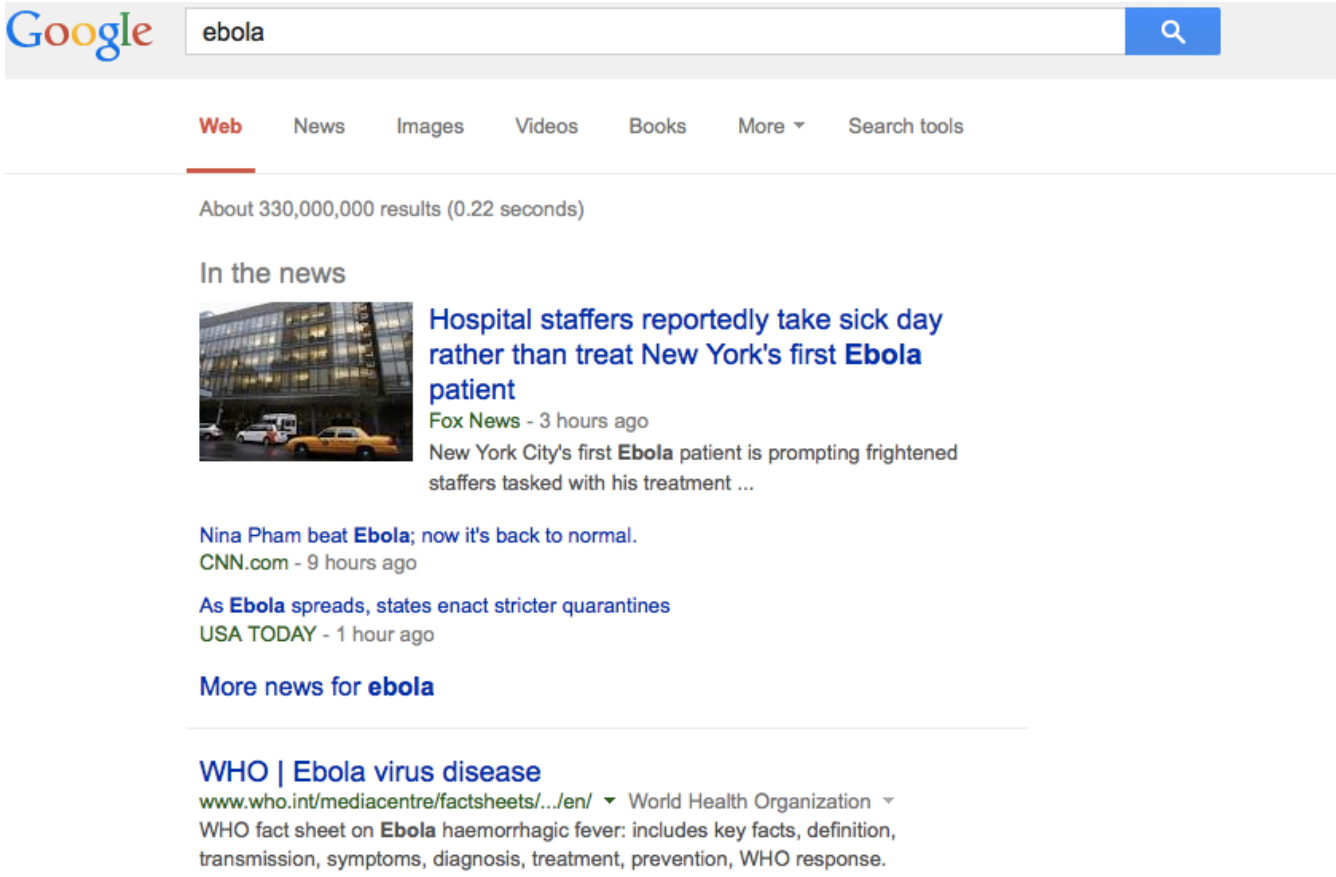
Documentaries





RECOMMENDATIONS DRIVE

The information I am exposed to…



RECOMMENDATIONS ARE VERY INTERESTING

- There is no single technique, and each problem is unique, though there are some core fundamentals

- Evaluation is not obvious, requires creativity and ingenuity

- Very few data mining products are this exposed to the public.
  - What are the design implications of a recommender system?
  - What are the ethical implications of a recommender system?

HOW TO RECOMMEND

TWO PHILOSOPHICAL APPROACHES

Recommend what is popular

MOST EMAILED | **MOST VIEWED** | RECOMMENDED FOR YOU

1. Parachutist's Record Fall: Over 25 Miles in 15 Minutes 
2. Why the Strong Reaction to Renée Zellweger's Face? 
3. First Patient Quarantined Under Strict New Policy Tests Negative for Ebola 
4. WELL
   The Advanced 7-Minute Workout 
5. 2 Die, Including Gunman, in Shooting at Washington State High School 
6. Kissing Your Socks Goodbye 

Try to understand your tastes

MOST EMAILED | MOST VIEWED | **RECOMMENDED FOR YOU**

1. Missouri: Panel Will Study Race and Poverty
2. Florida: A.C.L.U. Sues Over Housing Law
3. Amnesty International Report Faults the Police in Ferguson, Mo.
4. Debate for Florida Governor Takes On a Hostile Edge 
5. Searching for the Fountain of Youth 
6. MORTGAGES
   Cheaper to Buy Than to Rent 
7. New York City Police to Be Equipped With Smartphones and Tablets 

COLLABORATIVE FILTERING

2 TYPES

User based

Find like users and make recommendations based on ratings/scores of like users

Item Based

Find similar items and recommend items similar to an item a user has shown interest in.

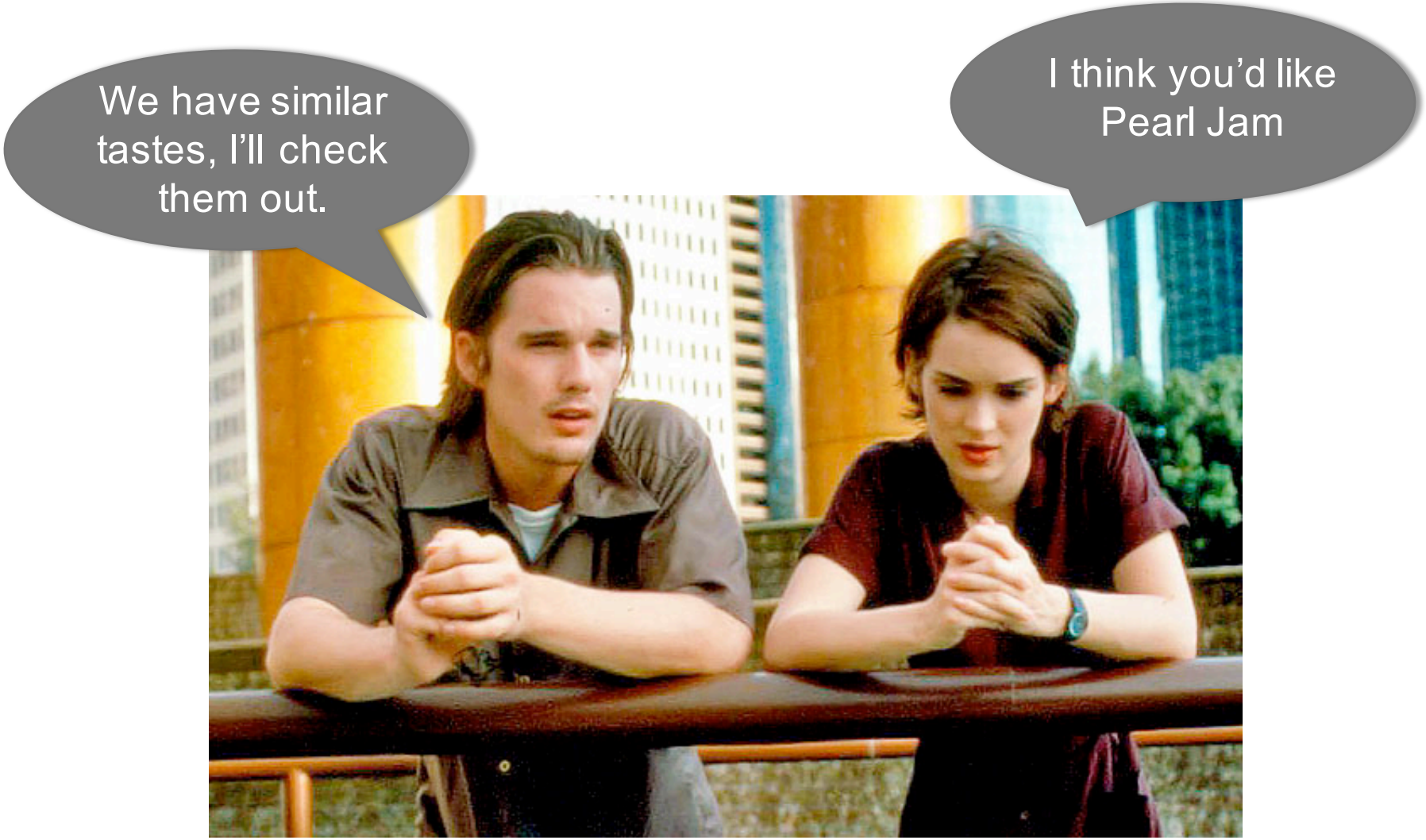
USER BASED

INTUITION

1. Find a group of people who like the same things similarly
2. Not everyone will like the exact same set of things

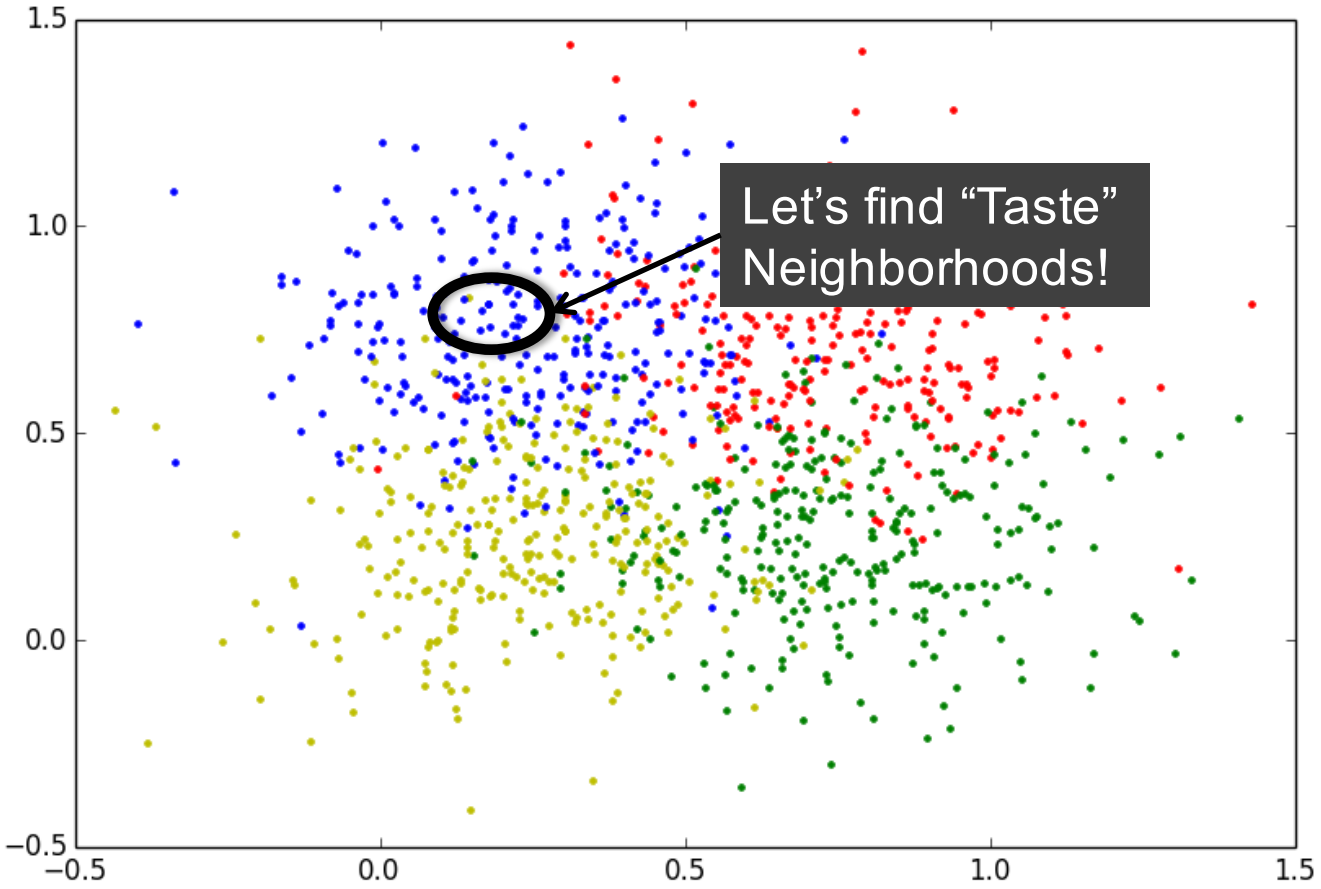
INTUITION

1. Find a group of people who like the same things similarly
2. Not everyone will like the exact same set of things
3. Recommend the non-overlapping items.



DEFINING A MECHANISM FOR "WE HAVE SIMILAR TASTES"

How do we translate "Find a group of people who like the same things" into a data science algorithm?



TOWARDS A "TASTE" NEIGHBORHOOD

First let's define the data structure.

Let A be the user-item matrix. Each entry a_{ij} can either be a rating or some binary indicator for user i on item j.

$A =$

	Item 1	Item 2	Item 3	Item 4	…	Item K
User 1	2	1			…	
User 2		2	4		…	2
User 3	3				…	
User 4	1	2	5	3	…	
User 5	3	2			…	
User 6					…	1
User 7		4	1		…	4
User 8		4	2		…	5
User 9	1				…	
User 10			3	4	…	1
…	…	…	…	…	…	
User N				1	…	4

TOWARDS A "TASTE" NEIGHBORHOOD

Second let's create a neighborhood for user i.

$A =$

	Item 1	Item 2	Item 3	Item 4	…	Item K
User 1	2	1			…	
User 2		2	4		…	2
User 3	3				…	
User 4	1	2	5	3	…	
User 5	3	2			…	
User 6					…	1
User 7		4	1		…	4
User 8		4	2		…	5
User 9	1				…	
User 10			3	4	…	1
…	…	…	…	…	…	
User N				1	…	4

TOWARDS A "TASTE" NEIGHBORHOOD

To do this, we need to define user-user similarity or distance.

$A =$

	Item 1	Item 2	Item 3	Item 4	…	Item K
User 1	2	1			…	
User 2		2	4		…	2
User 3	3				…	
User 4	1	2	5	3	…	
User 5	3	2			…	
User 6					…	1
User 7		4	1		…	4
User 8		4	2		…	5
User 9	1				…	
User 10			3	4	…	1
…	…	…	…	…	…	
User N				1	…	4

TOWARDS A "TASTE" NEIGHBORHOOD

Each user i corresponds to a row in our user-item matrix A. There are two popular ways to define similarity between similar rows of A.

Let A_i and A_k be two row vectors corresponding to the items i and k have rated. Let S be the set of items both users have rated/selected. Then,

Cosine Similarity

$$sim(A_i, A_k) = cos(A_i, A_k) = \frac{A_i \cdot A_k}{\|A_i\|_2 \|A_k\|_2}$$

Pearson Correlation

$$sim(A_i, A_k) = \frac{\sum_{j \in S} (A_{ij} - \mu_i)(A_{kj} - \mu_k)}{\sqrt{\sum_{j \in S} (A_{ij} - \mu_i)^2 \sum_{j \in S} (A_{kj} - \mu_k)^2}}$$

A PRACTICAL ASIDE

Things to consider when choosing a similarity measure:

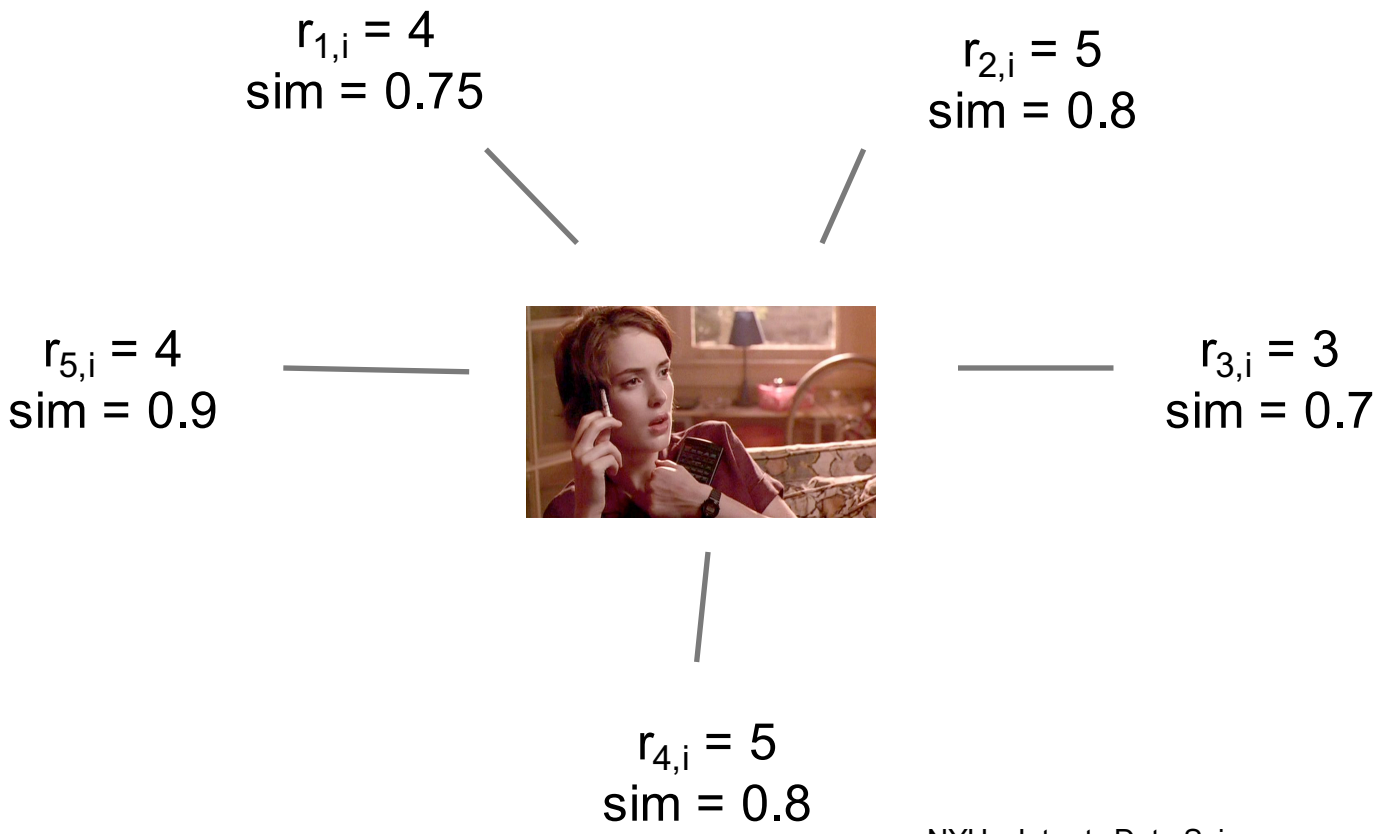
- Which metric is better for ratings vs. binary indicators?
- Should you mean normalize? I.e., subtract user and/or item average rating from each rating.
- Should take Similarity over all items for each user, or just those in common. I.e, should S be the intersection or union of A_i and A_k.

The right answer will likely depend on your problem.
Testing and experimentation is important in each case.

MAKING THE RECOMMENDATION

The predicted score/rating for user *u* on product *i* is then a function of scores/ratings that all users in u's neighborhood gave to the same product.

$$r_{u,i} = Agg_{u' \in U}(r_{u',i})$$



DIFFERENT WAYS TO AGGREGATE

Take a simple average.

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$

Take a weighted avg, weighted by similarity…

$$r_{u,i} = \frac{1}{k} \sum_{u' \in U} sim(u, u') * r_{u',i}$$

$$k = \sum_{u' \in U} sim(u, u')$$

There are many other ways to define the aggregation function. Other variants use averages but normalize out the means of the individual users to account for user-specific biases.

EG.

Once you have defined the neighborhood, aggregation is pretty straightforward.

$r_{1,i} = 4$
sim = 0.75

$r_{2,i} = 5$
sim = 0.8

$r_{3,i} = 3$
sim = 0.7

$r_{4,i} = 5$
sim = 0.8

$r_{5,i} = 4$
sim = 0.9

User	ri,j	sim	ri,j*sim
1	4	0.75	3.0
2	5	0.8	4.0
3	3	0.6	1.8
4	4	0.8	3.2
5	5	0.9	4.5

Average	**4.20**
Weighted Average	**4.29**

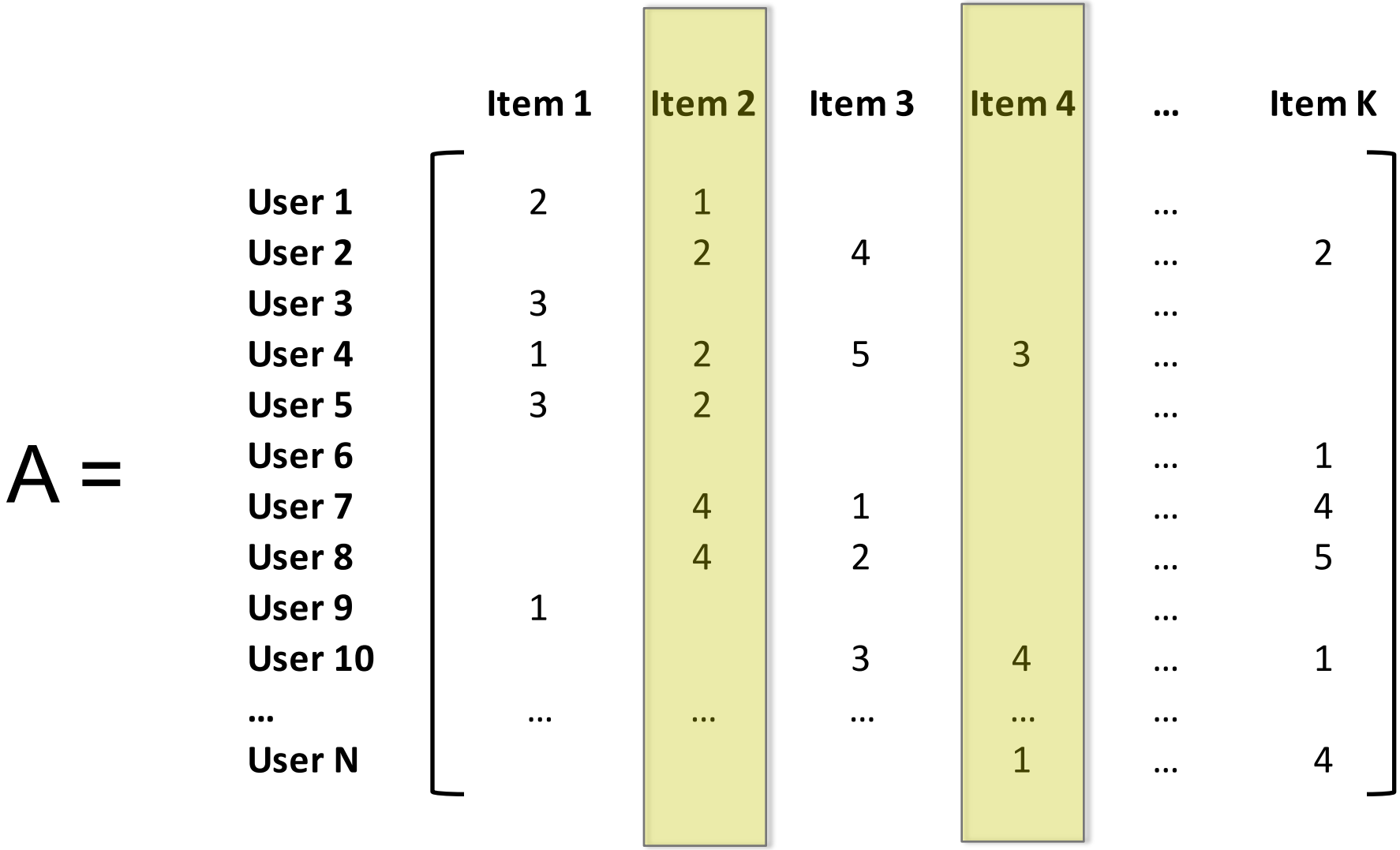
ITEM BASED

DEVELOPED BY AMAZON

User based methods can be unscalable as user-item matrix grows.

START WITH USER-ITEM MATRIX

But this time we care about similarity between columns, not rows. We can use the same type of similarity functions that we used in the user based system.



DERIVE THE ITEM-ITEM MATRIX

2 approaches to recommendation:

1. If a user has selected/purchased an item, find the k most similar items

2. For each item the user hasn't selected/purchased, predict user's rating/score for that product as a function of the user's rating/score on similar items (similar to the user based kNN approach)

$M =$

	Item 1	Item 2	Item 3	Item 4	…	Item K
Item 1		.8	0.2	0	…	0.1
Item 2			0.5	0.11	…	0.6
Item 3				0.3	…	0.4
Item 4					…	0.8
…	…	…	…	…	…	0.3
Item K					…	

MATRIX FACTORIZATION BASED RECOMMENDATIONS

BEYOND COLLABORATIVE FILTERING

The Netflix recommendation system contest in the mid-aughts ushered in a new paradigm for making recommendations.

Given a ratings matrix, decompose:

$$A = U \Sigma V^T$$

Instead of creating user taste, or item similarity neighborhoods, we can predict a user's rating on an item by uncovering the latent dimensions of the ratings matrix.

FACTORIZING THE RATINGS MATRIX

We simplify the factorization to:

$$A = UV^T$$

V

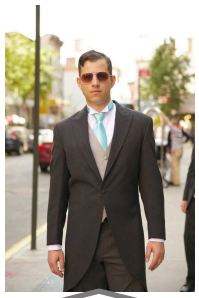
Each column of V represents a latent movie dimension. The value V_{ij} tells us how much movie i can be described by the latent movie dimension j.

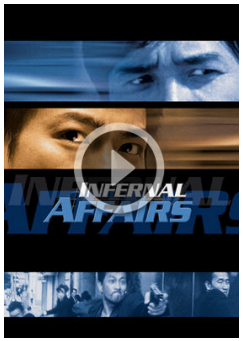
U

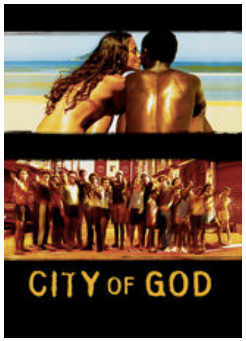
Each element U_{ij} of U represents how much user i has an affinity for the latent movie dimension V_j.

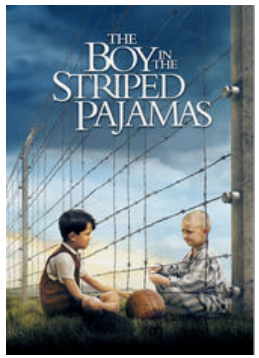
LATENT FACTORS

We observe things like:









Crime Thrillers

Artistic Dramas

Handsome Leading Men

LATENT FACTORS

What drives these observations? In other words, why do we do watch what we watch?

- Genre (comedy vs. romance)
- Audience (children vs. adults)
- Style (quirky vs. serious)
- Depth of Character
- Presence of certain actors

We want a computerized way to be able to extract these properties and not rely on human curation.

LATENT FACTORS

When uncovering latent factors, we usually only want a subset k, where k<<min(M,N)

U_k

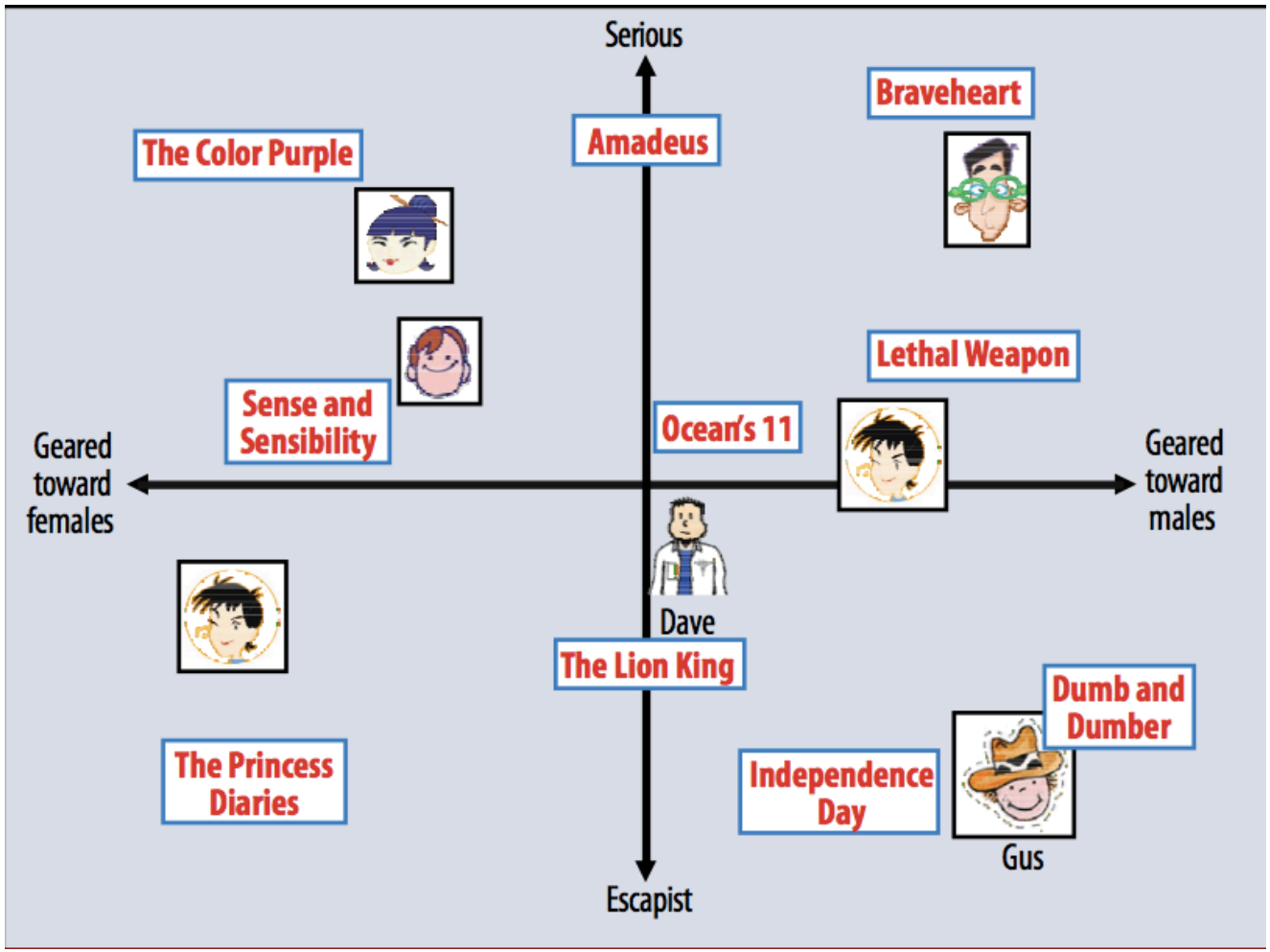
	LF 1	LF 1	LF 3
User 1	-0.7	-0.8	-0.4
User 2	0.2	-0.6	1.0
User 3	-0.8	-0.1	-0.8
User 4	0.4	0.3	-0.1
...			
User N	-0.3	1.0	0.4

V_k^T

	Item 1	Item 2	Item 3	...	Item M
LF 1	-0.4	0.6	0.8	...	-0.8
LF 2	-0.8	-0.5	-0.7	...	-0.4
LF 3	0.1	0.9	0.7	...	-0.7

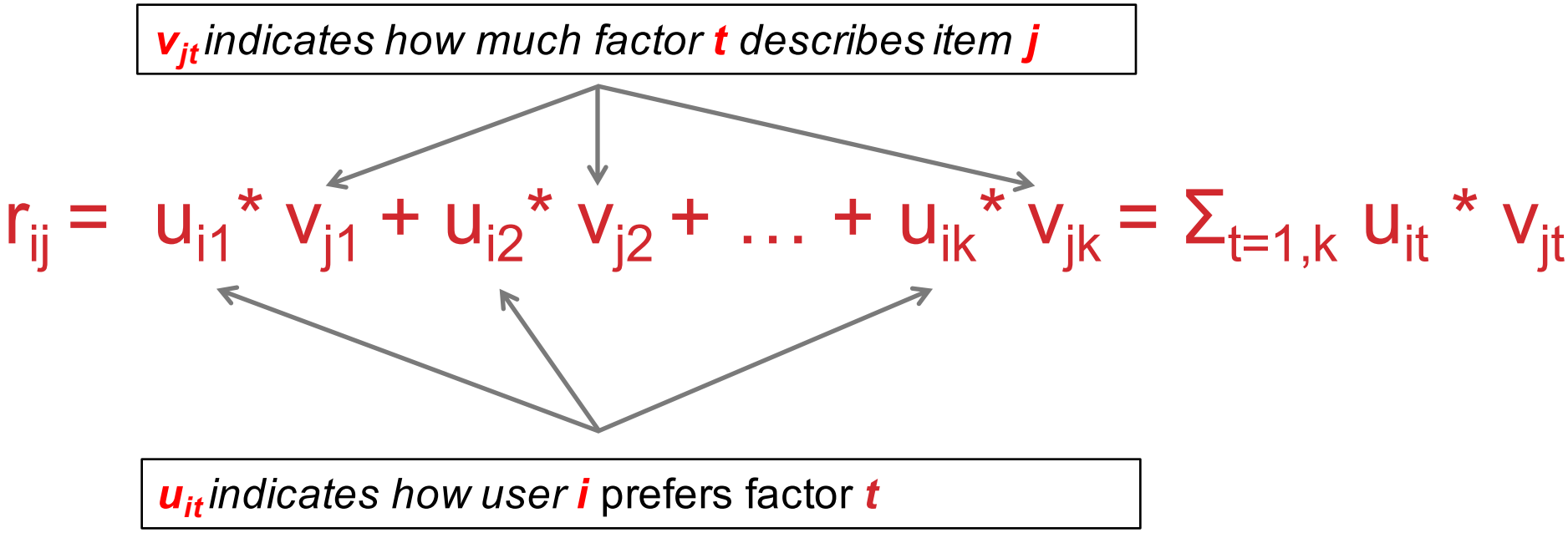
LATENT FACTORS

An example set of movies and how they load on two latent variables.



THE RATING PREDICTION

The rating prediction for user i on item j is then an inner product between the the user's preferences for each latent factor and the item's strength on that factor.



LEARNING THE FACTORIZATION

We can set this recommendation problem up as a supervised learning problem.

Minimize the sum of squares predictions of observed ratings.

$$\hat{U}, \hat{V} = \underset{U,V}{\operatorname{argmin}} \sum_{r_{ij} \in A} (r_{ij} - \sum_{t}^{k} u_{it} * v_{jt})^2 + \lambda \left(\|U\|^2 + \|V\|^2 \right)$$

We regularize the components of U and V to avoid over-fitting