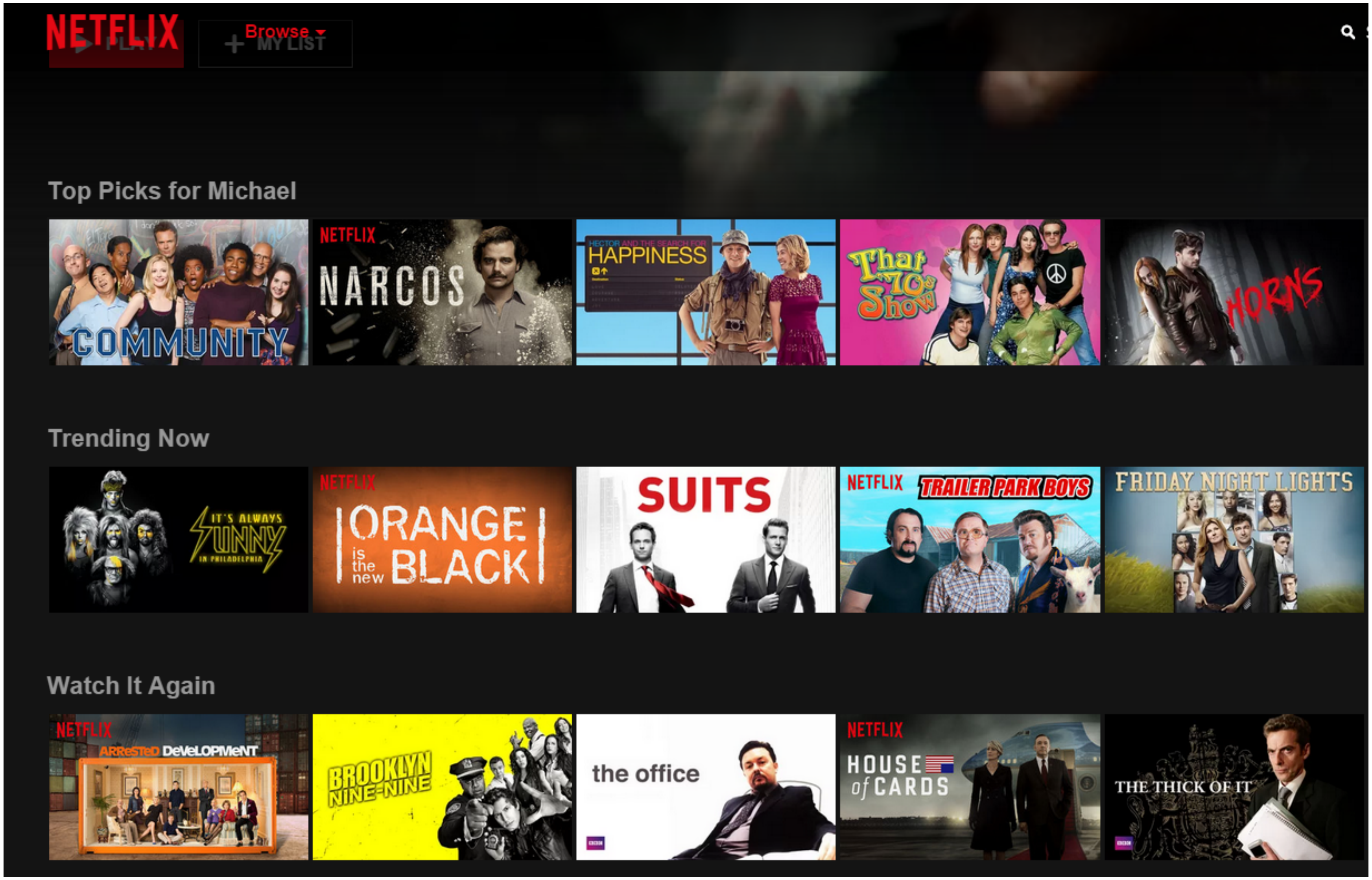
Recommender Systems



Agenda

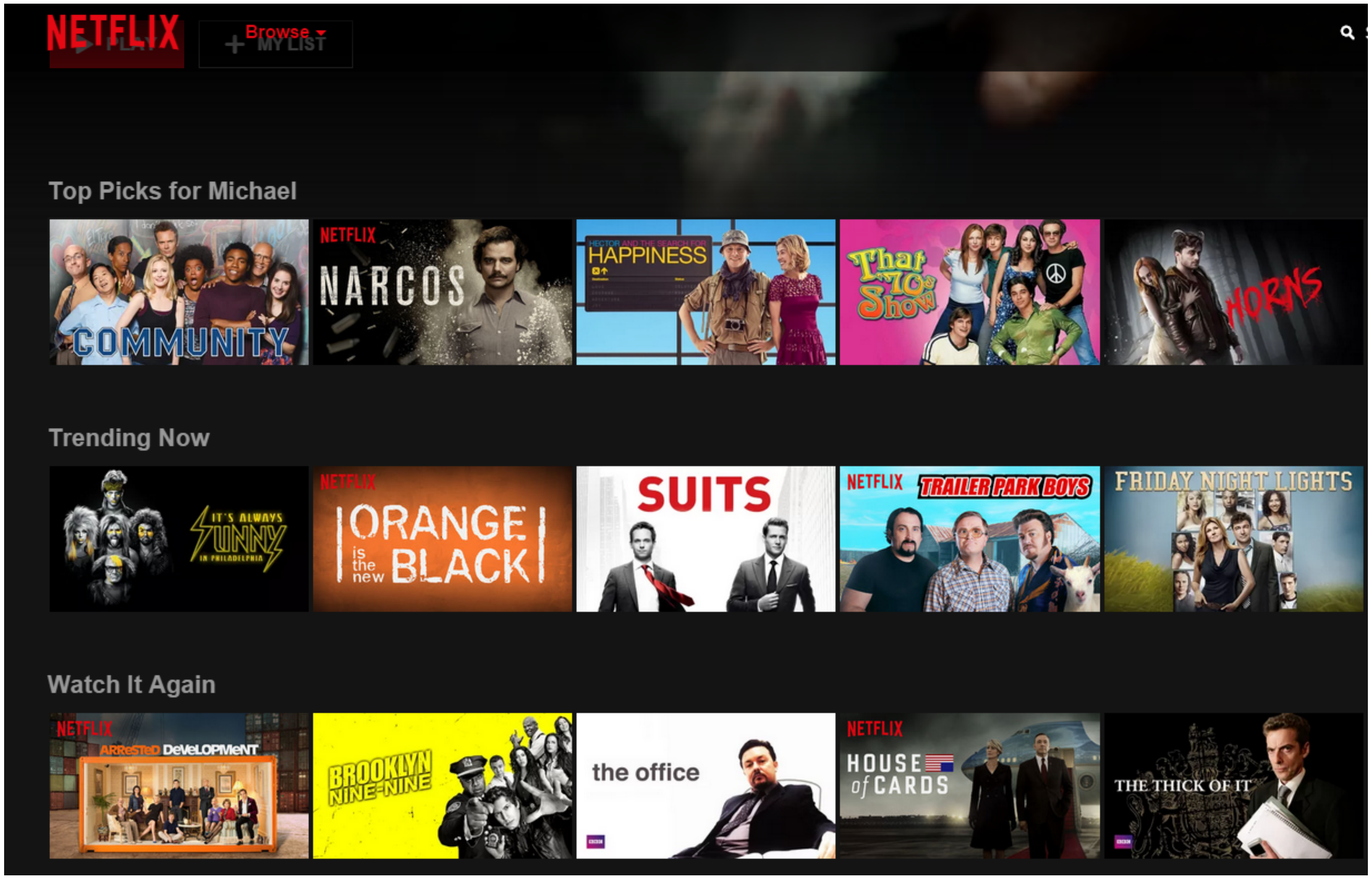
- Introduction
- User-based collaborative filtering
- Content-based filtering
- Historical examples

Introduction

- *Recommender Systems* predict what you'll like
- Put another way: they predict your preferences
- What are some examples?

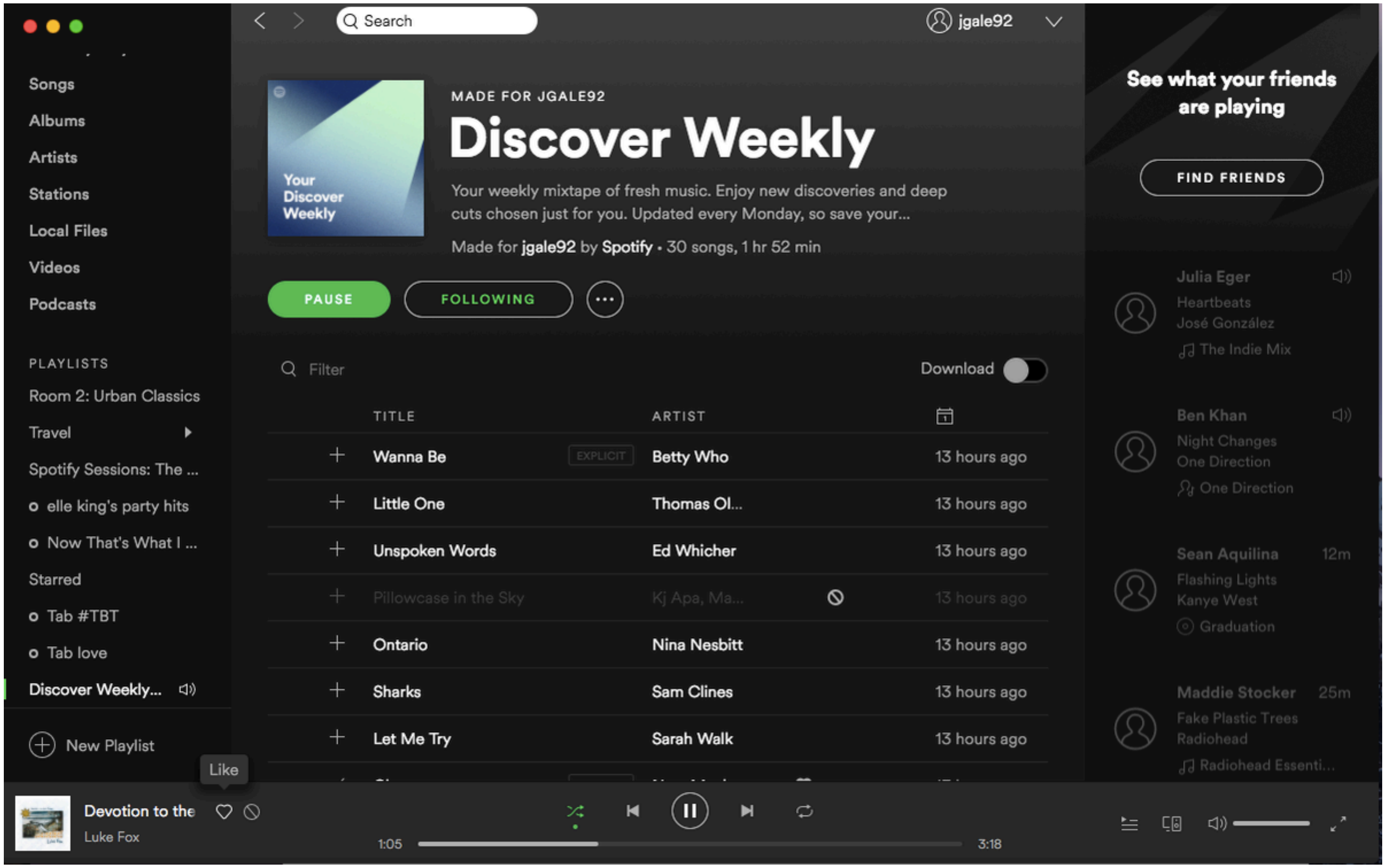
Example

- Movies/shows (e.g., Netflix, Amazon, Hulu, etc.)



Example

- Music (e.g., Spotify, Apple Music, etc.)



MADE FOR JGALE92

Discover Weekly

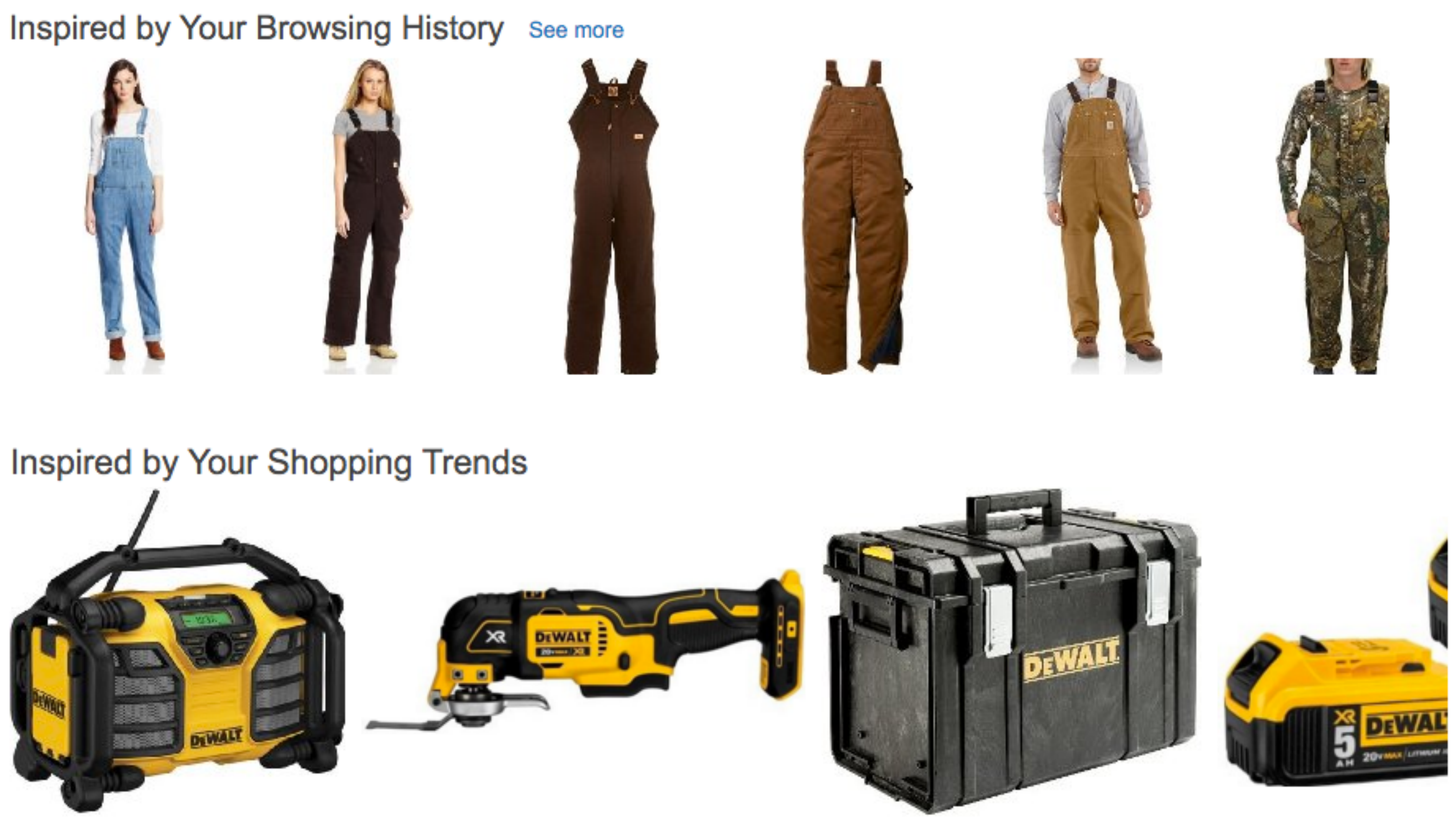
Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your...

Made for jgale92 by Spotify • 30 songs, 1 hr 52 min

TITLE		ARTIST	
Wanna Be	EXPLICIT	Betty Who	13 hours ago
Little One		Thomas Ol...	13 hours ago
Unspoken Words		Ed Whicher	13 hours ago
Pillowcase in the Sky		Kj Apa, Ma...	13 hours ago
Ontario		Nina Nesbitt	13 hours ago
Sharks		Sam Clines	13 hours ago
Let Me Try		Sarah Walk	13 hours ago

Example

- Products (e.g., Amazon, etc.)


Inspired by Your Browsing History
See more
Inspired by Your Shopping Trends

Example

- More examples
  - Movies
  - Music
  - Products
  - Books
  - Video games
  - Colleagues
  - Friends

Recommendation challenges

- Recommendation is common, but surprisingly hard
- Lots of recommendations to make
- Millions of products
- Users have very little tolerance for adding preference data; system knows almost nothing about you
- Everyone is different (right?)

Data collection

- Explicit data collection
  - Ask users to rate movies, products, whatever
  - Ask users for demographic information
- Implicit data collection
  - Web logs of past user activity
  - Timing information, e.g., how long you paused on a post before scrolling

Data collection

- How to predict what movies you like? Features?
- One approach: do it like Web pages
  - Collect data on my movie likes

The Godfather	4
Ernest Goes to Camp	3
Casablanca	2
36 Hours	5
Love and Death	4

  - Collect features: genre, length, year, etc.
  - Build score-predictor; recommend high-scorers
- Problems?

Traditional ML Doesn't Work

- Unfortunately, film-qualities (*features*) may be difficult to extract
- Problem: How to recommend movies without knowing anything about movies?

Agenda

- Introduction
- **User-based collaborative filtering**
- Content-based filtering
- Historical examples

Collaborative filtering

- Problem: How to recommend movies without knowing anything about movies?
- Solution: Recommend movies enjoyed by *people who are similar to you*

- People who agreed in the past will agree in the future

Averaging

- Filling in missing scores
- Average for each item over all users

Averaging example

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	?	2	5	1
Chris	4	2	4	?
Donna	3	?	5	1

Averaging example

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	3.66	2	5	1
Chris	4	2	4	2
Donna	3	2	5	1

Averaging

- Problem: Averaging ignores uniqueness of a user
  - Terrible when there is large variation in interest
  - Movies, music are a few examples

- Solution: Nearest neighbor algorithm

Nearest neighbor algorithm

- Find another user with similar properties
  - Movie ratings in this example
- Use other user's rating to "fill in the blank"
- People who agreed in the past will agree in the future

Nearest neighbor example

- Who are the nearest neighbors?

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	**?**	2	5	1
Chris	4	2	4	**?**
Donna	3	**?**	5	1

Nearest neighbor example

- Bob's nearest neighbor is Donna

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	?	2	**5**	**1**
Chris	4	2	4	?
Donna	3	?	**5**	**1**

Nearest neighbor example

- Use Donna's rating to fill in Bob's

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	**3**	2	5	1
Chris	4	2	4	?
Donna	**3**	?	5	1

Nearest neighbor example

- Chris's nearest neighbor is Alice

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	3	2	5	1
Chris	4	2	4	**4**
Donna	3	?	5	1

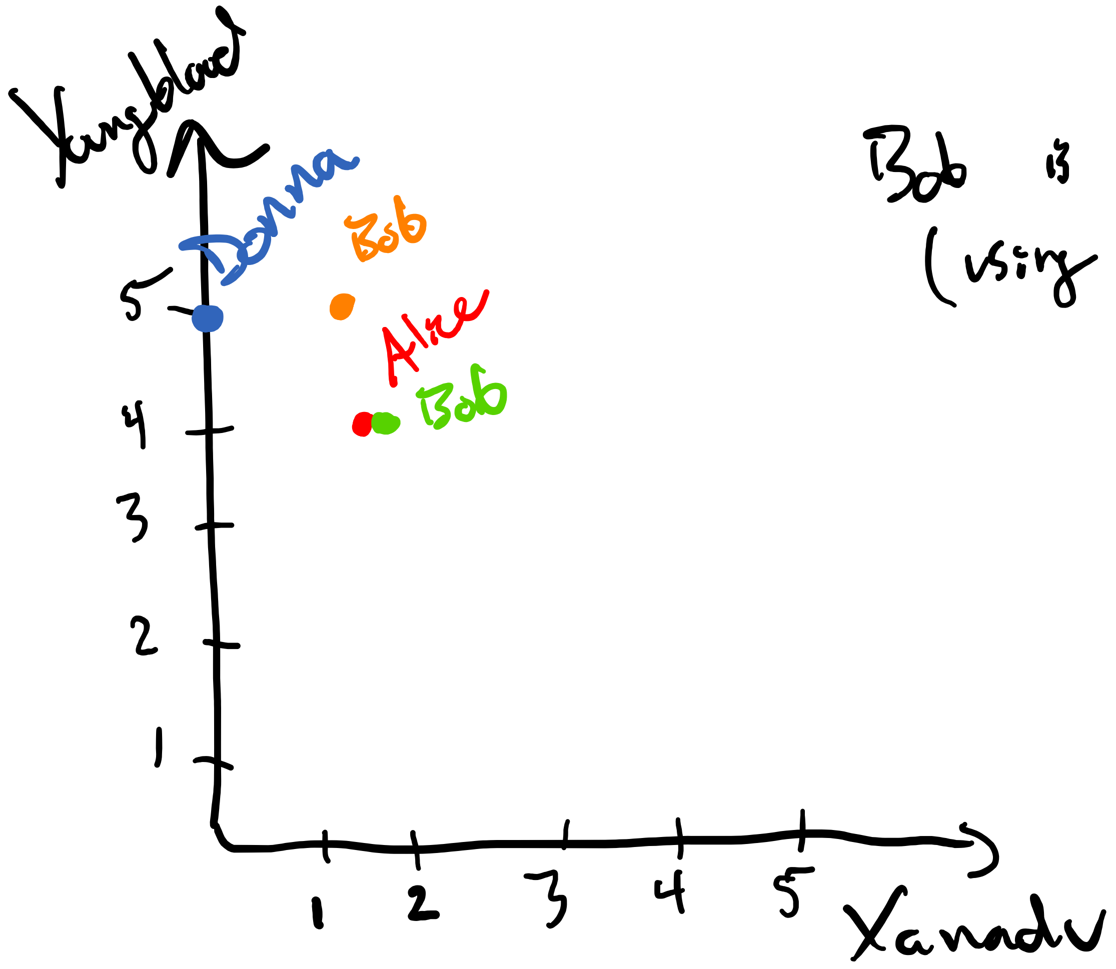
Nearest neighbor example

- Donna's nearest neighbor is Bob

	W.	Xanadu	Youngblood	Zorro
Alice	4	2	4	4
Bob	3	2	5	1
Chris	4	2	4	4
Donna	3	**2**	5	1

Nearest neighbor diagram

- We'll use only two movies to make it easier to draw



Bob is Donna's nearest neighbor (using Euclidean distance)

	Xanadu	Youngblood
Alice	2	4
Bob	2	5
Chris	2	4
Donna		5

Distance metric

- How to measure distance between neighbors?
- Many possibilities, including
  - Euclidean distance
  - Cosine similarity
  - Pearson correlation coefficient
- A lot in common with information retrieval and the document vector model

Euclidean distance

- Each movie rated by both users is a dimension
- One user is one vector in multidimensional space
- Measure distance between the two vectors

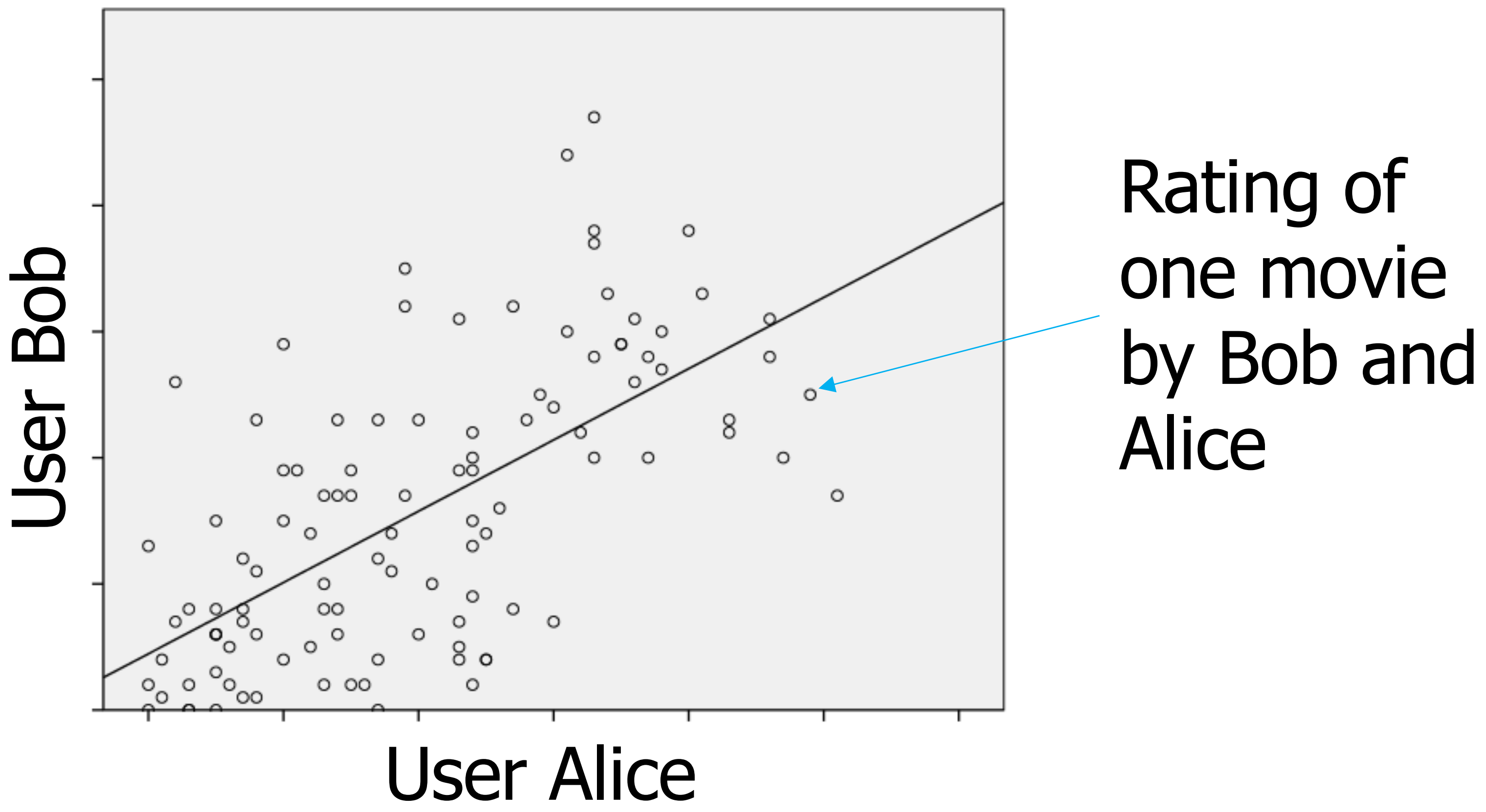
Cosine similarity

- Each movie rated by both users is a dimension
- One user is one vector in multidimensional space

- Cosine similarity = 1
  - Vectors "pointed in the same direction"
- Cosine similarity = 0
  - Vectors orthogonal
- Cosine similarity doesn't consider the length of the vector, only the angle

Pearson correlation coefficient

- Each movie rated by both users is a point



Pearson correlation coefficient

- Value between +1 and −1
    - 1 is total positive correlation
    - 0 is no correlation
    - −1 is total negative correlation

Pearson correlation coefficient

- S = set of movies
- $r_{u,i}$ = rating of user u on movie *i*
- S_u = {i ∈ S |*u* saw movie *i*}
- S_{uv} = {i ∈ S | both *u* and *v* saw movie *i*}

$$r_u = \sum_{i \in S_u} r_{u,i} / |S_u|$$

$$\frac{\sum_{i \in S_{uv}} (r_{u,i} - r_u)(r_{v,i} - r_v)}{\sqrt{\sum_{i \in S_{uv}} (r_{u,i} - r_u)^2 \sum_{i \in S_{uv}} (r_{v,i} - r_v)^2}}$$

K-nearest neighbors (k-NN)

- Can we do better than selecting just one nearest neighbor?
- Select several.

1. Find the *k* closest users
2. Select the most frequent score (mode)

k-NN problems

1. Find the k closest users
2. Select the most frequent score

- How would a really popular film affect recommendations?

k-NN problems

1. Find the k closest users
2. Select the most frequent score

- How would a really popular film affect recommendations?
- The popular film will be recommended most of the time because it will often be the most frequent score
- The popular choice might not always be the best choice for a user

k-NN with weights

- Solution: weight other users' scores by similarity

$$r_{u,i} = k \sum_{v \in Top-Sim(u)} sim(u,v) r_{v,i}$$

- Top-Sim(u) is the *n* most-similar user neighbors to *u*
- *k* is a normalizer

$$k = 1 / \sum_{v \in Top-Sim(u)} \left| sim(u,v) \right|$$

k-NN problems

- How would a "cult classic" film affect recommendations?
  - Enjoyed by a few, who REALLY like it
- What could you do about this?

k-NN problems

- How would a "cult classic" film affect recommendations?
  - Enjoyed by few, but they REALLY like it
- Rare film less likely to be predicted by k-NN
- Can use *inverse-user-frequency*
  - Similar to TFxIDF's inverse document frequency
  - Rarely seen movies are more influential

k-NN and machine learning

- K-nearest neighbors is a classification algorithm
- Example of recommender systems using machine learning techniques
- Many other techniques that we don't cover

Collaborative filtering weaknesses

- Cold start
  - Need user data before you can start making accurate recommendations
- Scalability
  - Millions of users, n-choose-2 pairs
  - Large amount of computation power
- Sparsity
  - Most users will only have rated a small subset of the overall database
  - Even the most popular items have very few ratings

Agenda

- Introduction
- User-based collaborative filtering
- **Content-based filtering**
- Historical examples

Motivation

- Problems with user-based collaborative filtering
  - Cold start, scalability, sparsity
- Solution: focus on content instead of users
  - Avoid nearest-neighbor operations on users

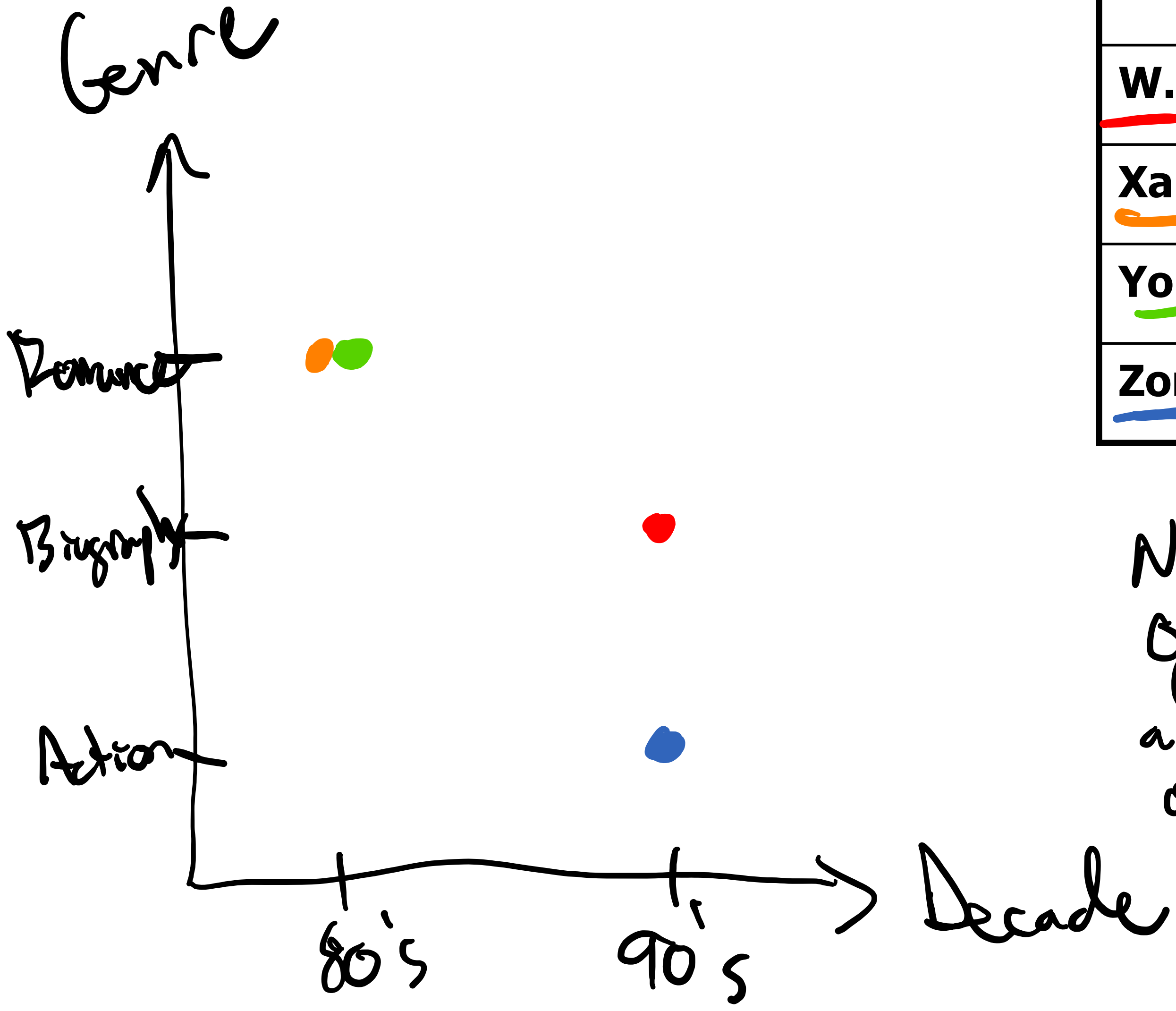
Content-based filtering

- *Content-based filtering:* Recommend items similar to items the user has liked in the past
- People will like the same things in the future that they liked in the past

Content-based filtering synonyms

- Content-based filtering
- AKA Content-based collaborative filtering
- AKA Item-based collaborative filtering
- AKA "Users who bought *x* also bought *y*"

Content-based filtering example



	Decade	Genre
W.	90's	Biography
Xanadu	80's	Romance
Youngblood	80's	Romance
Zorro	90's	Action

Note: closeness of two genres is arbitrary. Can avoid this problem with one-hot encoding.

Content-based filtering

- How can we compute item similarity?
  - Similar techniques to user-based collaborative filtering
  - Euclidean distance, cosine similarity, correlation, tf-idf

Content-based filtering weaknesses

- Problem: Website doesn't know all your preferences
  - You've rated horror movies on Netflix, but you also like comedies
- Problem: Lost opportunity to introduce you to new content
- Solution: Hybrid filtering

Hybrid filtering

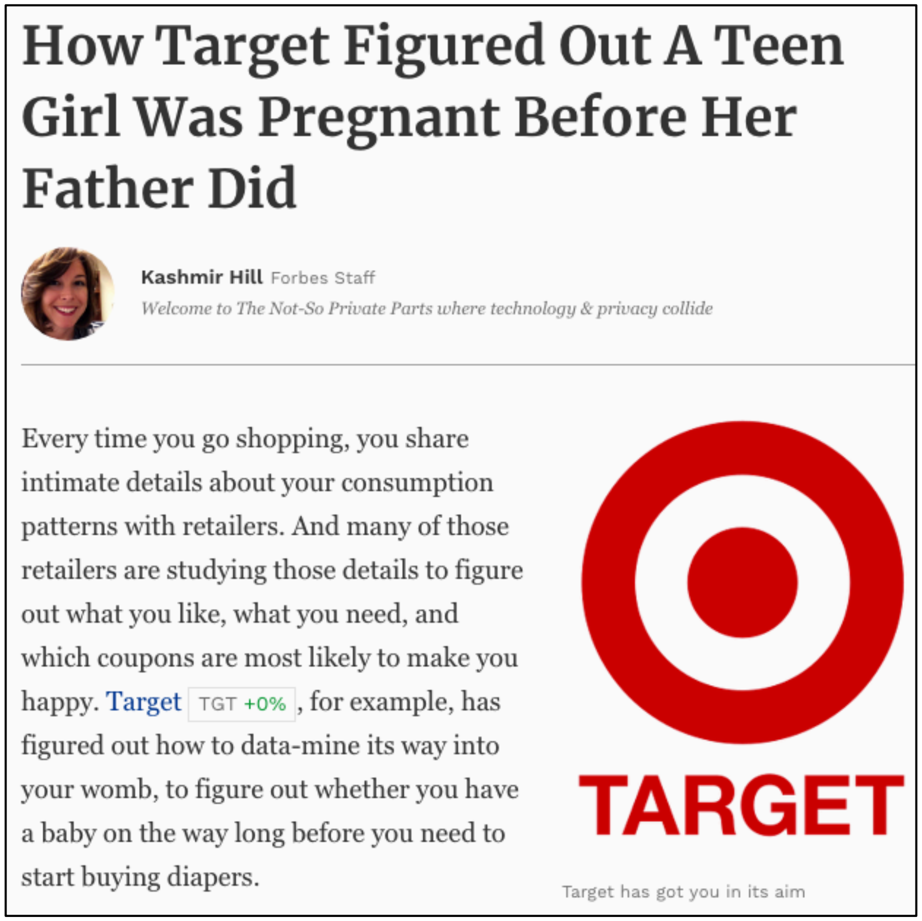
- *Hybrid filtering:* Combine user-based and content-based filtering
- Netflix Example:
  - Recommendations based on similar users; AND
  - Recommendations based on shows user has rated highly

Agenda

- Introduction
- User-based collaborative filtering
- Content-based filtering
- **Historical examples**

Target and pregnancy prediction

1. Predict pregnancy
2. Recommend pregnancy related products
3. Father angry when teen daughter receives ads
4. Father finds out daughter is pregnant



How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

Kashmir Hill Forbes Staff
Welcome to The Not-So Private Parts where technology & privacy collide

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target TGT +0%, for example, has figured out how to data-mine its way into your womb, to figure out whether you have a baby on the way long before you need to start buying diapers.

Target has got you in its aim

Target and pregnancy prediction

- Features were products
  - Unscented lotion, calcium, magnesium and zinc supplements
- Sent ads through the mail
- Story from 2012

Netflix prize

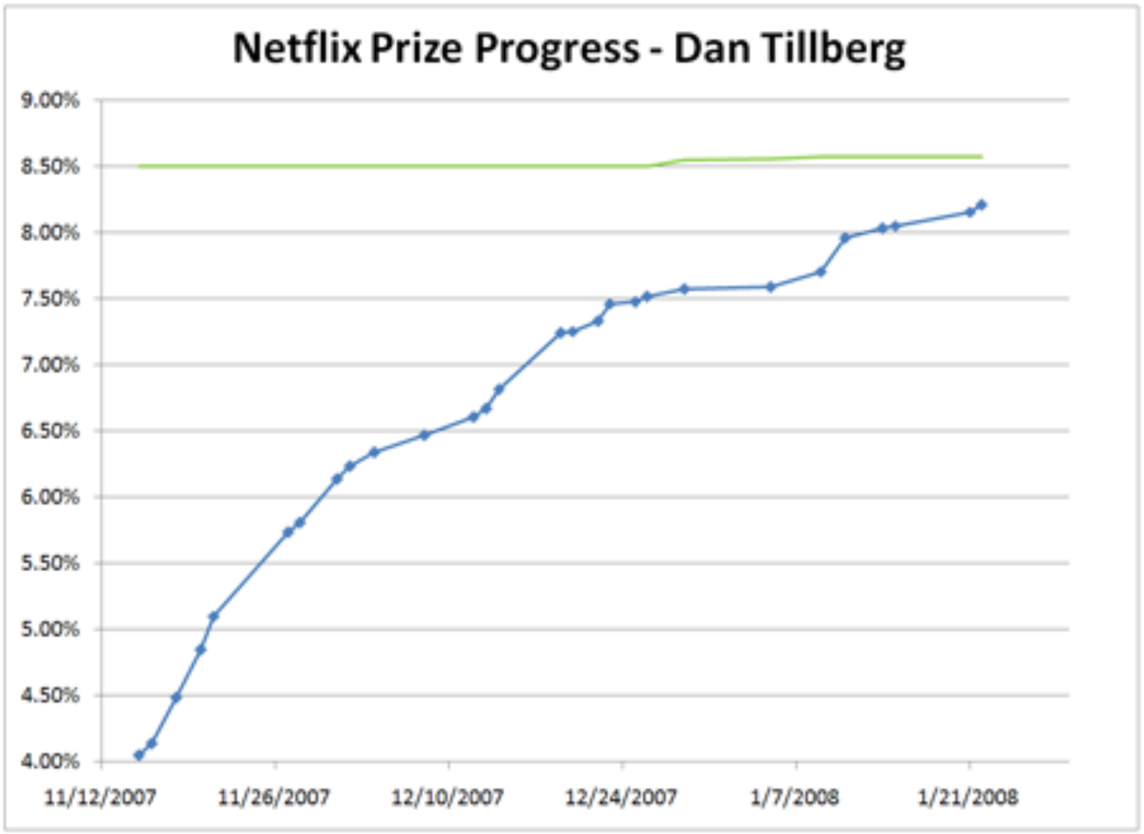
- Announced October 2, 2006
- $1M to whoever could improve NetFlix's own recommender by 10%
- Data from Netflix in the form:
  - <user, movie, date, grade>
  - Where grade is 1...5
  - That's it
- Training data: 100M examples from ~500k users on ~18k movies
- Ideas for how you might do this?

Netflix prize

- How did they do it?
- Among many ideas:
  - Use IMDB for director, genre, etc.
  - Some movies' grades change when reviewers grade them in a clump, suggesting a long time has passed since the movie was viewed
  - User grades depend on day of week

Netflix prize

- Dan Tillberg's progress…
- Dropped out in 2008 while in 5th place



Netflix prize

- Won on September 21, 2009

Summary

- User-based collaborative filtering: recommend items enjoyed by *users who are similar to you*

- Content-based filtering: recommend items *similar to items you have enjoyed in the past*