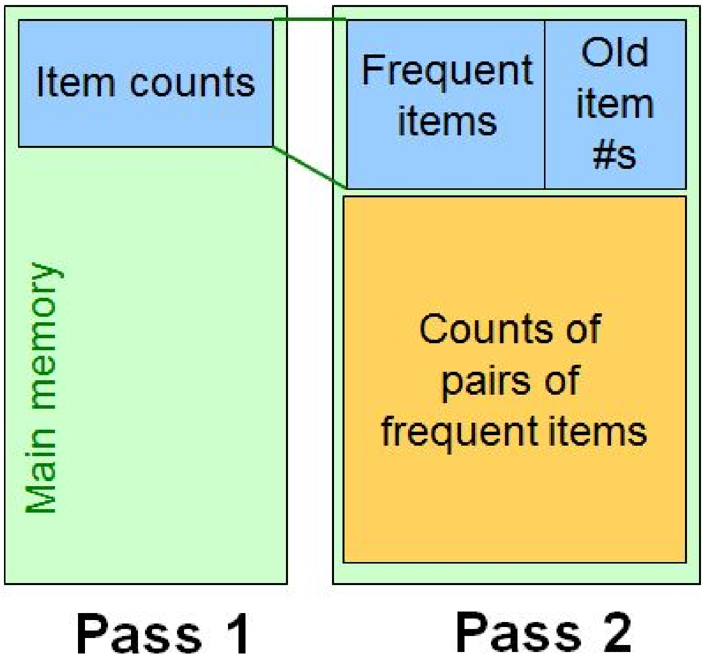
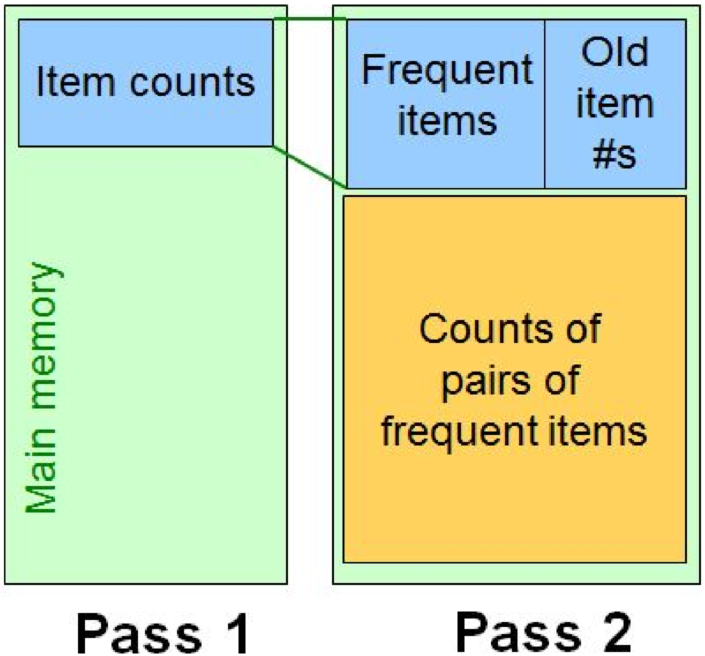
Recommendation Systems

Stony Brook University
CSE545, Fall 2016

From Frequent to Recommended



From Frequent to Recommended



Similar idea, but slightly different question:

- Frequent items: Which items belong together?
- Recommendation Systems:
  - What other item will this **user** like? (based on previously liked items)
  - How much will user like item X?

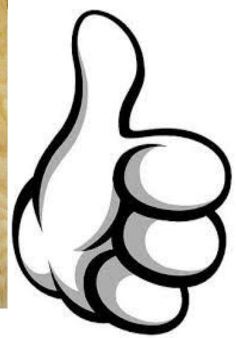
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

From Frequent to Recommended

Similar idea, but slightly different question:

- Frequent items: Which items belong together?
- Recommendation Systems:
  - What other item will this **user** like? (based on previously liked items)
  - How much will user like item X?

From Frequent to Recommended



?

Similar idea, but slightly different question:

- Frequent items: Which items belong together?

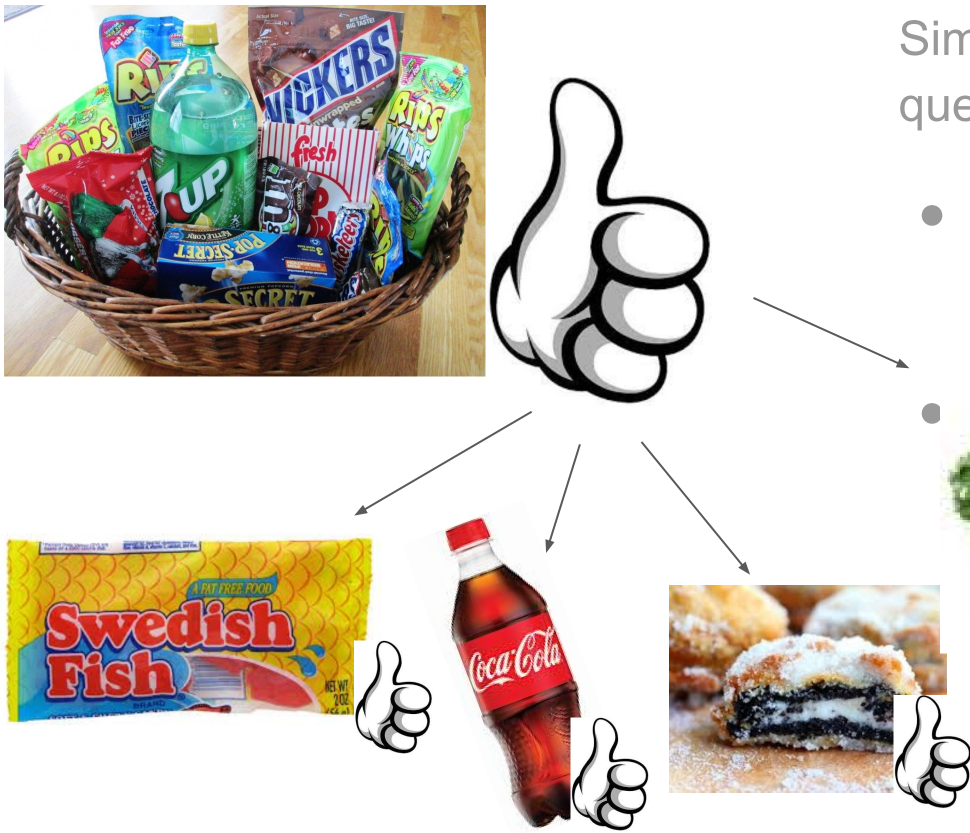
- Recommendation Systems:
  - What other item will this **user** like? (based on previously liked items)
  - How much will user like item X?

From Frequent to Recommended

Similar idea, but slightly different question:

- Frequent items: Which items belong together?
- Recommendation Systems:
  - What other item will this **user** like? (based on previously liked items)
  - How much will user like item X?

From Frequent to Recommended



Similar idea, but slightly different question:

- Frequent items: Which items belong

- Systems:
  ɪm will this **user**
  ɪn previously liked
  - H uch will user like item X?

From Frequent to Recommended

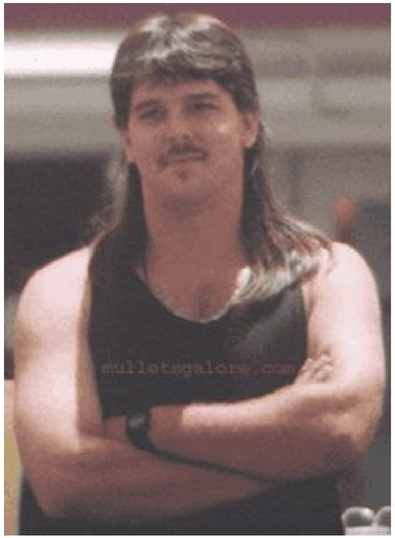
Past User Ratings

Recommendation Systems

Why Big Data?

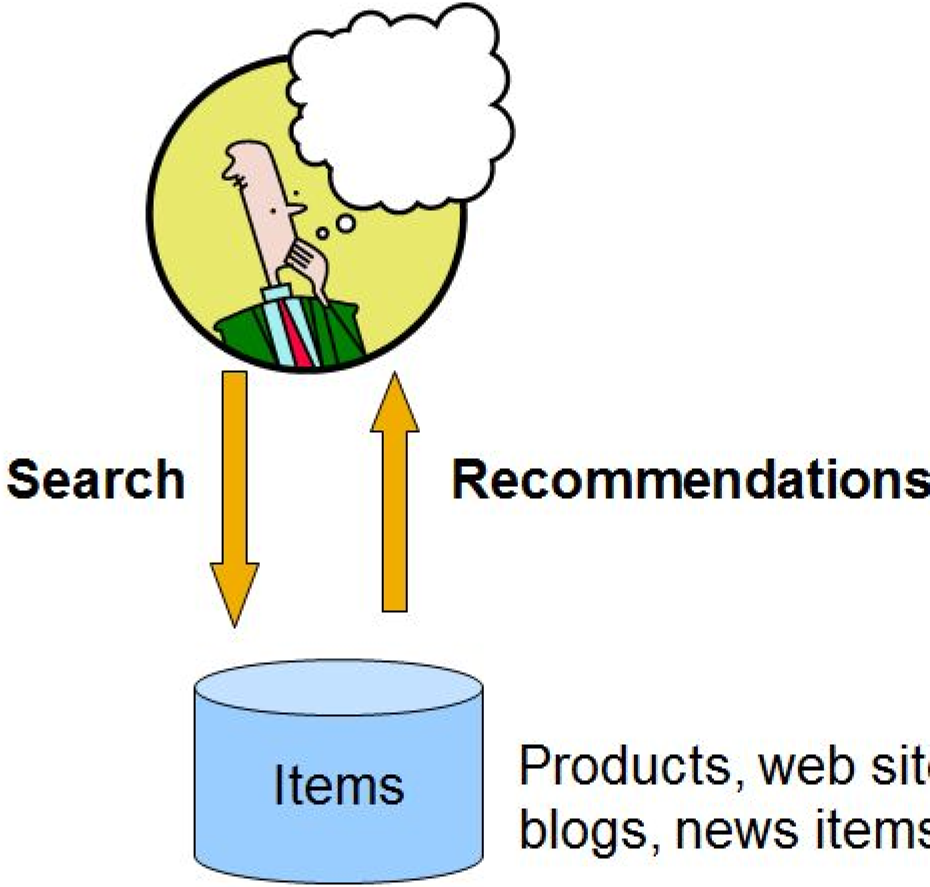
- Data with many potential features (and sometimes observations)
- An application of techniques for finding similar items
  - Locality sensitive hashing
  - Clustering / dimensionality reduction

Recommendation System: Example



- **Customer X**
  - Buys Metallica CD
  - Buys Megadeth CD

- **Customer Y**
  - Does search on Metallica
  - Recommender system suggests Megadeth from data collected about customer **X**



Products, web sites, blogs, news items, …

Examples:

amazon.com
PANDORA
StumbleUpon
del.icio.us
NETFLIX
movielens
helping you find the *right* movies
last.fm
the social music revolution
Google News
YouTube
XBOX LIVE

Enabled by Web Shopping

- Does Wal-Mart have everything you need?

Enabled by Web Shopping

- Does Wal-Mart have everything you need?

(thelongtail.com)

Enabled by Web Shopping

- Does Wal-Mart have everything you need?

- A lot of products are only of interest to a small population (i.e. "long-tail products").
- However, most people buy many products that are from the long-tail.

- Web shopping enables more choices
  - Harder to search
  - Recommendation engines to the rescue

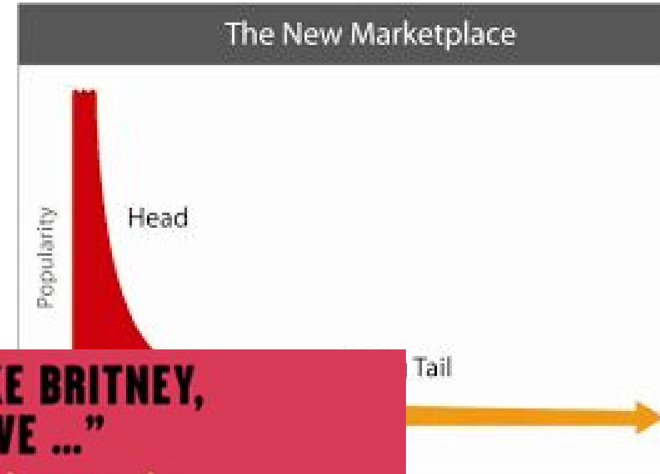
(thelongtail.com)

Enabled by Web Shopping

- Does Wal-Mart have everything you need?

- A lot of products are only of interest to a small population (i.e. "long-tail products").
- However, most people buy many products that are fro

- Web shopp
  - Harder t
  - Recomm



A Model for Recommendation Systems

Given: *users*, *items, utility matrix*

A Model for Recommendation Systems

Given: *users*, *items*, *utility matrix*

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4	5	3		3
B	5			4	2
C			5	2	

A Model for Recommendation Systems

Given: *users*, *items*, *utility matrix*

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4	5	3		3
B	5			4	2
C	?	?	5	2	?

Recommendation Systems

Problems to tackle:

1. Gathering ratings

2. Extrapolate unknown ratings
   a. Explicit: based on user ratings and reviews
      (problem: only a few users engage in such tasks)
   b. Implicit: Learn from actions (e.g. purchases, clicks)
      (problem: hard to learn low ratings)

3. Evaluation

Recommendation Systems

Problems to tackle:

1. Gathering ratings

2. Extrapolate unknown ratings
   a. Explicit: based on user ratings and reviews
      (problem: only a few users engage in such tasks)
   b. Implicit: Learn from actions (e.g. purchases, clicks)
      (problem: hard to learn low ratings)

3. Evaluation

Recommendation Systems

Problems to tackle:

1. Gathering ratings
2. Extrapolate unknown ratings
   a. Explicit: based on user ratings and reviews (problem: only a few users engage in such tasks)
   b. Implicit: Learn from actions (e.g. purchases, clicks) (problem: hard to learn low ratings)
3. Evaluation

Common Approaches

1. Content-based
2. Collaborative
3. Latent Factor

Recommendation Systems

Problems to tackle:

1. Gathering ratings
2. Extrapolate unknown ratings
   a. Explicit: based on user ratings and reviews (problem: only a few users engage in such tasks)
   b. Implicit: Learn from actions (e.g. purchases, clicks) (problem: hard to learn low ratings)
3. Evaluation

Common Approaches

1. Content-based
2. Collaborative
3. Latent Factor

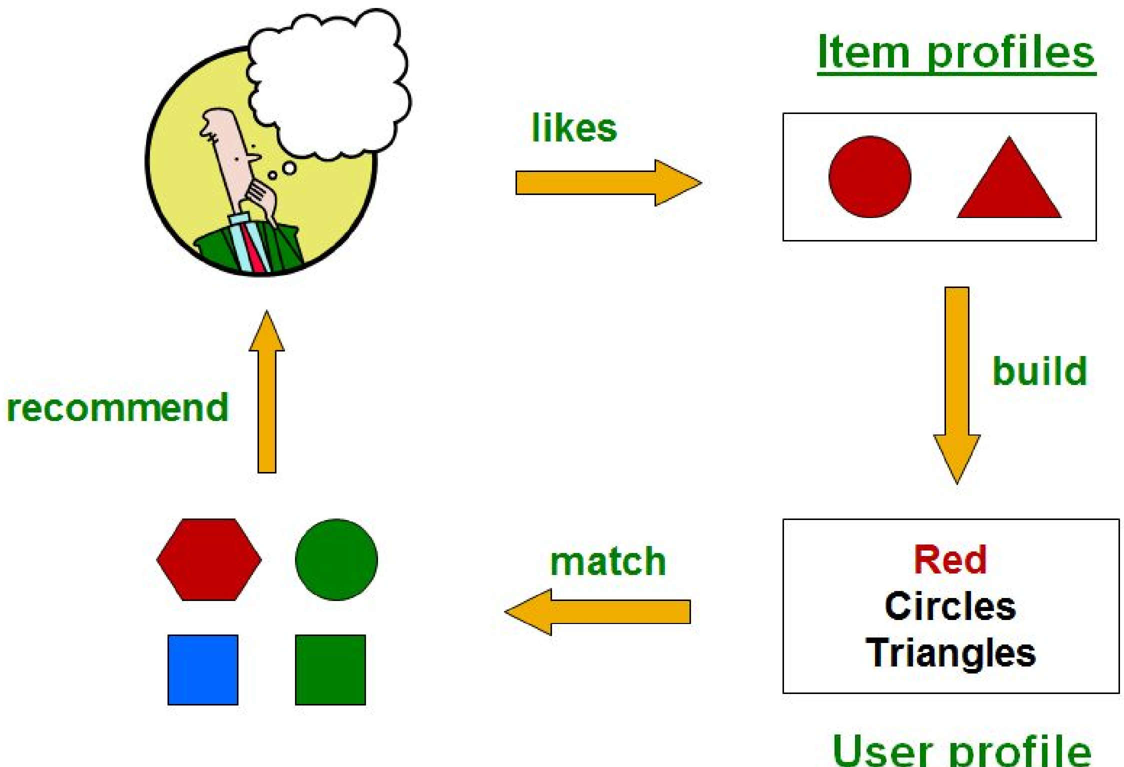
Key Challenge:
New users have no ratings or history
(a cold-start)

Content-based Rec Systems

Based on similarity of items to past items that they have rated.

Content-based Rec Systems

Based on similarity of items to past items that they have rated.



J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Content-based Rec Systems

Based on similarity of items to past items that they have rated.

1. `Build profiles of items (set of features);` examples:

   shows: producer, actors, theme, review

   people: friends, posts

pick words with tf-idf

Content-based Rec Systems

Based on similarity of items to past items that they have rated.

1. `Build profiles of items (set of features);` examples:
   shows: producer, actors, theme, review
   people: friends, posts
   pick words with tf-idf
2. `Construct user profile from item profiles;` approach:
   average all item profiles
   variation: weight by difference from their average

Content-based Rec Systems

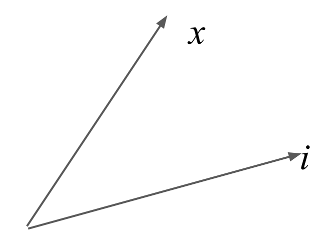
Based on similarity of items to past items that they have rated.

1. `Build profiles of items (set of features);` examples:
    - *shows:* producer, actors, theme, review
    - *people:* friends, posts

   pick words with tf-idf

2. `Construct user profile from item profiles;` approach:
    - average all item profiles
    - variation: weight by difference from their average
3. `Predict ratings for new items;` approach:

$$utility(user, i) = \cos(x, i) = \frac{x \cdot i}{||x|| \cdot ||i||}$$



Why Content Based?

- Only need users history
- Captures unique tastes
- Can recommend new items
- Can provide explanations

Why Content Based?

- Only need users history
- Captures unique tastes
- Can recommend new items
- Can provide explanations

- Need good features
- New users don’t have history
- Doesn’t venture “outside the box” (Overspecialized)

Why Content Based?

- Only need users history
- Captures unique tastes
- Can recommend new items
- Can provide explanations

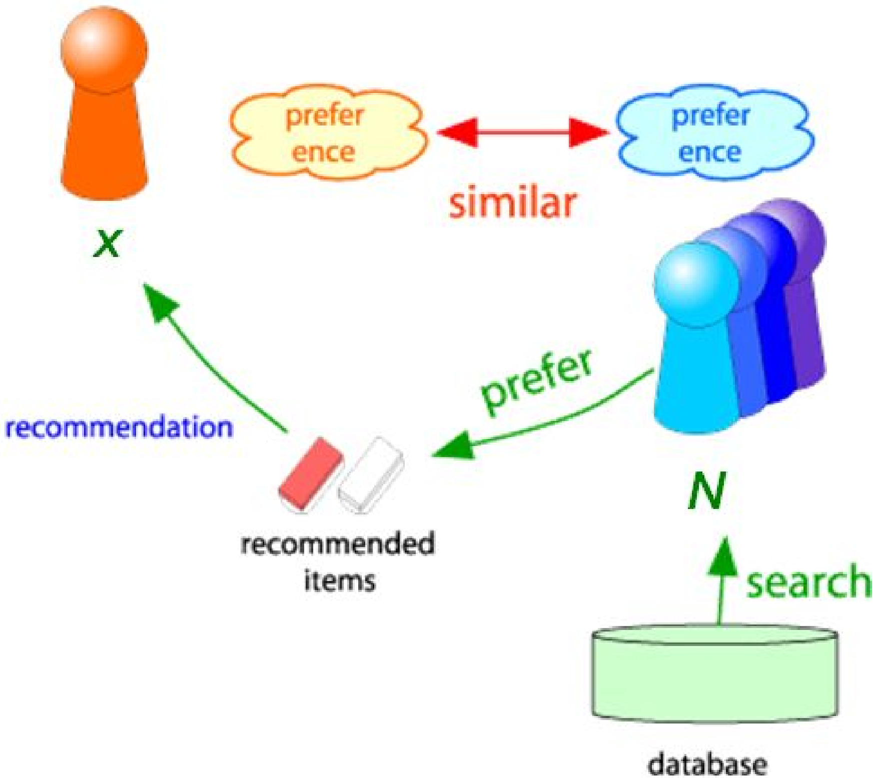
- Need good features
- New users don’t have history
- Doesn’t venture “outside the box” (Overspecialized)

(not exploiting other users judgments)

Collaborative Filtering Rec Systems

(not exploiting other users judgments)

Collaborative Filtering Rec Systems



(not exploiting other users judgments)

Collaborative Filtering Rec Systems

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4	5	2		3
B	5			4	2
C			5	2	

Collaborative Filtering Rec Systems

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4	5	2		3
B	5			4	2
C			5	2	

1. Find Similarity
   (need to handle missing values) : subtract user’s mean

Collaborative Filtering Rec Systems

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4 => 0.5	5 => 1.5	2 => -1.5	=> 0	3 => -0.5
B	5			4	2
C			5	2	

Given user, *x*, item, *i*

1. Find neighborhood, *N* -- set of *k* users most similar to *x* who have also rated *i*

    Find similarity between all users (using cosine sim)

    (need to handle missing values) : subtract user's mean

Collaborative Filtering Rec Systems

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4 => 0.5	5 => 1.5	2 => -1.5	=> 0	3 => -0.5
B	5			4	2
C			5	2	

Given user, *x*, item, *i*

1. Find neighborhood, *N* -- set of *k* users most similar to *x* who have also rated *i*
   Find similarity between all users (using cosine sim)
   (need to handle missing values) : subtract user's mean
2. Predict utility (rating); options:
   a. take average
   b. weight average by similarity

$$utility(x, i) = \frac{\sum_{y \in N} Sim(x,y) \cdot utility(y,i)}{\sum_{y \in N} Sim(x,y)}$$

Collaborative Filtering Rec Systems

"User-User collaborative filtering"

Given user, x, item, i

1. Find neighborhood, N -- set of k users most similar to x who have also rated i
   Find similarity between all users
   (need to handle missing values) : subtract user's mean
2. Predict utility (rating); options:
   a. take average
   b. weight average by similarity

Collaborative Filtering Rec Systems

"User-User collaborative filtering"

Item-Item:

Flip rows/columns of utility matrix and use same methods.

user	Game of Thrones	Fargo	Ballers	Silicon Valley	Walking Dead
A	4	5	2		3
B	5			4	2
C			5	2	

CF: Example

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	

- unknown rating   - rating between 1 to 5

CF: Example

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	

- estimate rating of movie **1** by user **5**

CF: Example

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		**1.00**
2			5	4			4			2	1	3	**-0.18**
3	2	4		1	2		3		4	3	5		**0.41**
4		2	4		5			4			2		**-0.10**
5			4	3	4	2					2	5	**-0.31**
6	1		3		3			2			4		**0.59**

Neighbor selection:
Identify movies similar to movie 1, **rated by user 5**

Here we use Pearson correlation as similarity:
1) Subtract mean rating m_i from each movie i
$m_1 = (1+3+5+5+4)/5 = 3.6$
row 1: $[-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]$
2) Compute cosine similarities between rows

Same as cosine sim when substracting the mean

CF: Example

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Compute similarity weights:

$s_{1,3}=0.41, s_{1,6}=0.59$

CF: Example

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

***utility*(1, 5) = (0.41*2 + 0.59*3) / (0.41+0.59)**

$$\frac{\sum_{y \in N} Sim(x,y) \cdot utility(y,i)}{\sum_{y \in N} Sim(x,y)}$$

Item-Item v User-User

- Item-item often works better than user-user

  Users tend to be more different than each other than items are from each other.
  (e.g. user A likes jazz + rock, user B likes classical + rock,
  but user-A may still have same rock preferences as B;
  Users span genres but items usually do not)