UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

CS411 - Collaborative Filtering

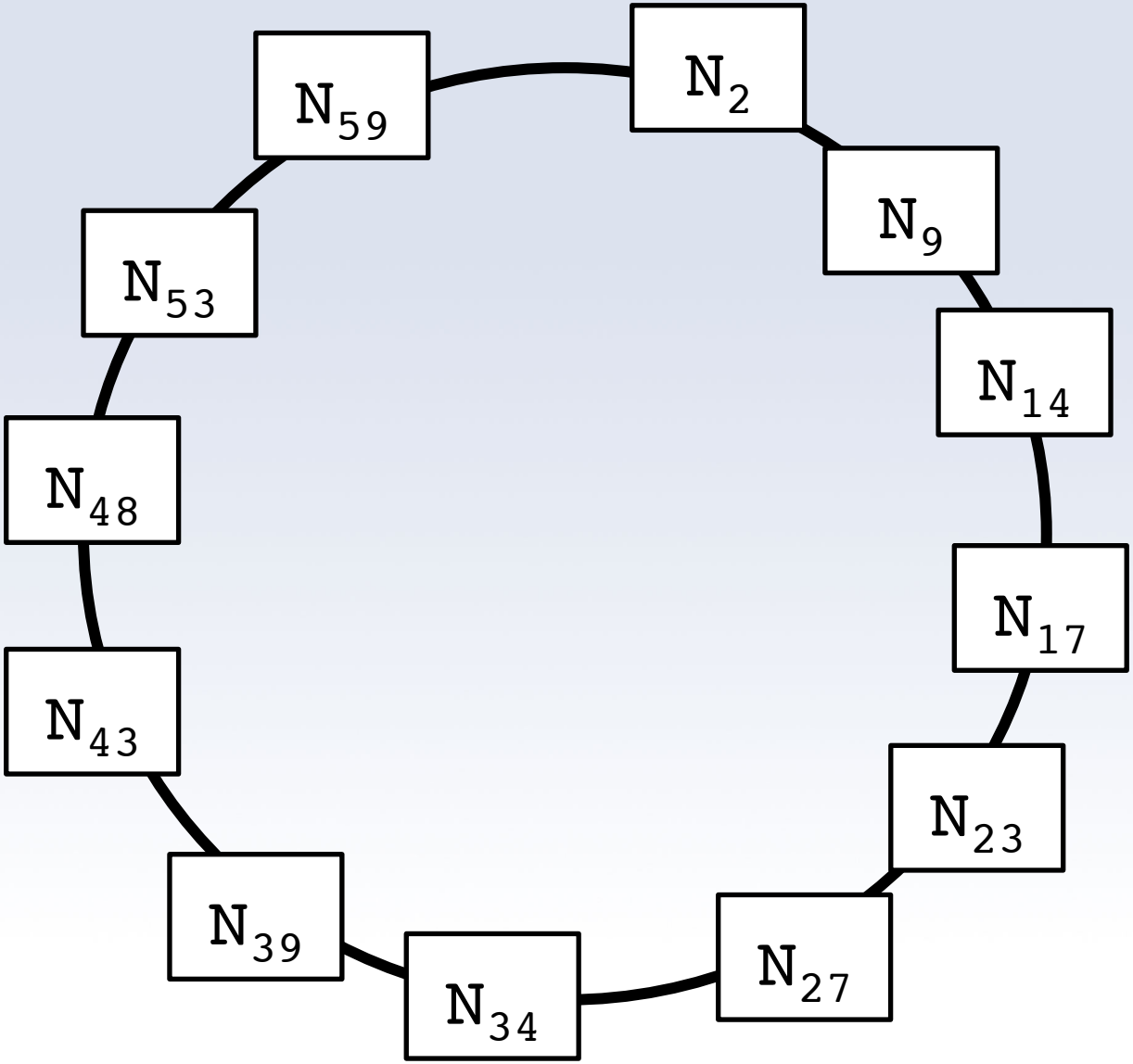
illinois.edu

Announcements

- HW4 due 05/01
- Last lecture
  - Exam review/project presentations
- Final Exam
  - Friday 05/03 at 8-11am
  - A-L Krannert Art Museum room 62
  - M-T Huff Hall room 112
  - U-Z Siebel Center 0216

Review

- What is a P2P network?
- What is NoSQL?
- What is a chord circle?
- What is a finger table?


N59
N2
N53
N9
N14
N48
N17
N43
N23
N39
N27
N34

Recommender Systems

- Goal: Make suggestions to the user
- Predict user preferences based on:
  - user's stated preferences/reviews
  - past user behavior
  - other users' behaviors
  - item properties

amazon
Join Prime
Shop by Department
People Who Bo
Trident AG-XT912-T DROID RAZR MAXX by...
site to store
$32.07

Based On: It's Real

25 songs ▾ 1.7 hours

	✓	Name	Time	Artist	Album	Genre
1	✓	It's Real	2:49	Real Estate	Days	Alternativ
2	✓	Cruel	3:35	St. Vincent	Strange Mercy	Alternativ
3	✓	Afternoon	4:10	Youth Lagoon	The Year Of Hibern…	Alternativ
4	✓	Amor Fati	4:26	Washed Out	Within And Without	Alternativ
5	✓	Kaputt	6:18	Destroyer	Kaputt	Pop/Rock
6	✓	Helicopter	4:58	Deerhunter	Halcyon Digest	Indie Roc
7	✓	Lazuli	5:02	Beach House	Bloom	Alternativ
8	✓	Chinatown	3:19	Wild Nothing	Gemini	Alternativ
9	✓	Origins	3:28	Tennis	Young & Old	Alternativ
					Father, Son, Holy…	Alternativ
					Before Today	Psychede
					Days	Alternativ
					WHOKILL	Indie Roc

match.com Member Sign In »
TAKE THE PERSONALITY TEST FROM CHEMISTRY.COM »
More Ways to Meet at Match.com!
Connect Online & at Local Events
I'm a Woman looking for a Man
Between ages 25 and 35
Near ZIP/Postal code VIEW PHOTOS »
NEW - Stir, Events by Match.com | Happy Hours, Cooking Classes & More!
last
Nazarete
ıal friends
Friend
lised
Fiol
ıl friends
Add Friend

Collaborative Filtering

- Focus on similarity of user behavior
- Examples
  - Amazon - items purchased by the same users
  - Pandora - songs played by the same users
  - Netflix - movies watched by the same users
  - Twitter - users followed by the same users
  - Facebook - users friends with same users

On the border of…

- Data Mining (CS412)
  - Finding simple rules or models to summarize massive datasets
- Information Retrieval (CS410)
  - Identifying relevant data in a large dataset
- Machine Learning (CS446)
  - Learning from data to make predictions

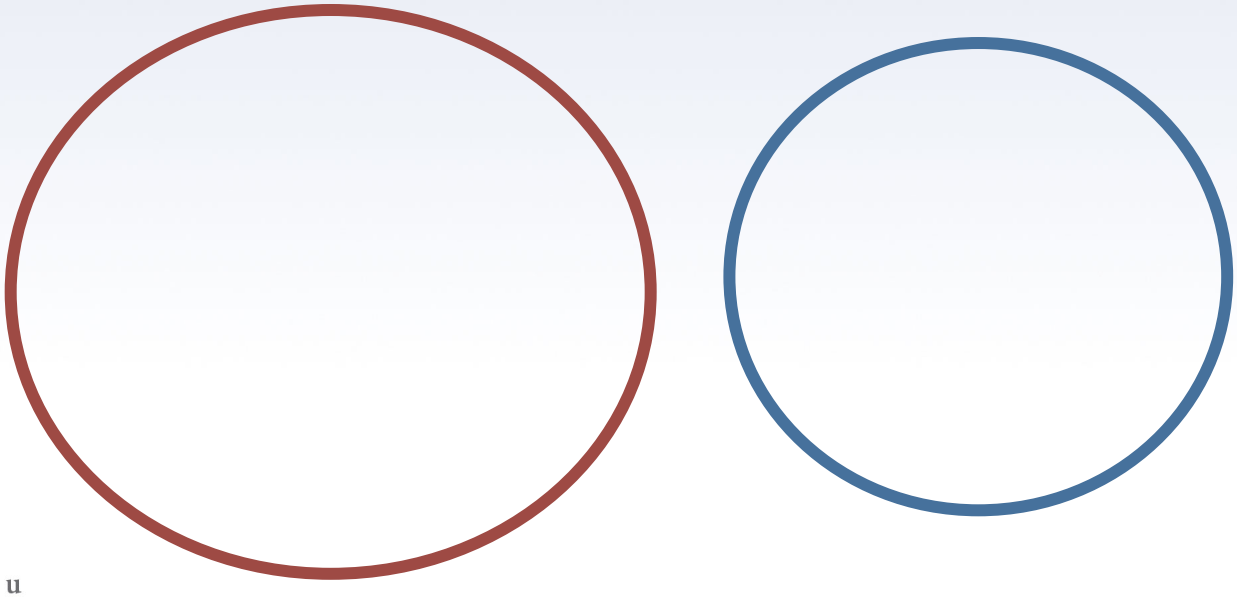
Running Example

- Netflix
  - Goal: identify similar movies
  - 7.1 million customers
  - 2.5 million movies in IMDB
  - 12 movies you actually want to watch

Jaccard Similarity

- Measure of similarity of two sets

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|}$$



Jaccard Similarity

“Inception”	S={1,4,6,9}
“Up”	T={1,2,3,5,7,8}
“Wall-E”	U={1,3,5,8,9}

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|} = ?$$

$$J(S,U) = ?$$

$$J(T,U) = ?$$

Our goal

- Calculate the Jaccard similarity for all pairs of movies in our database
- When a user watches a movie, we can recommend similar movies

Relational Model

- Could we solve this using an RDBMS?
- Data is probably in a SQL database

```
User(userId,name,address)
Movie(movieId,title,length)
Watched(userId,movieId)
```

Relational Model

```
SELECT COUNT(*)
FROM Watched AS W1, Watched AS W2
WHERE W1.userId=W2.userID AND
   (W1.movieId=m1 AND W2.movieId=m2);

SELECT COUNT(*)
FROM Watched AS W1, Watched AS W2
WHERE W1.userId=W2.userID AND
   (W1.movieId=m1 OR W2.movieId=m2);
```

Relational Model

- 7.1 million customers
- 2.5 million movies
- 3.1 trillion pairs of movies
- If every customer watched one movie…
  - Over 22 ***quintillion*** record accesses

Relational Model

- What if we tried the trick from HW1?

Compatibility

Person	Match	Score
Alex	Alex	10
Alex	Sam	5
Alex	Jesse	3
Alex	Terry	4
Sam	Alex	7
Sam	Sam	10
Sam	Jesse	9

Table

Person	Alex	Sam	Jesse	Terry
Alex	10	5	3	4
Sam	7	10	9	1
Jesse	3	8	10	3
Terry	2	3	4	10

Relational Model

	U1	U2	U3	U4	U5	U6	U7
M1	Yes	Yes	No	Yes	No	Yes	Yes
M2	No	Yes	No	No	No	No	No
M3	No	No	No	No	No	No	No
M4	No	Yes	No	No	Yes	No	Yes
M5	No	No	Yes	No	Yes	No	No
M6	No	No	No	Yes	No	Yes	No
M7	Yes	No	No	Yes	No	Yes	No
M8	Yes	No	No	Yes	No	Yes	Yes
M9	No	Yes	No	Yes	No	Yes	No
M10	No	No	Yes	No	No	No	Yes
M11	Yes	Yes	No	No	Yes	No	No

Relational Model

- 7.1 million customers
- 2.5 million movies
- 17.8 trillion entries in this table…
  - 16 terabytes of data!
  - AND we still have to compute similarity of 3.1 trillion pairs of movies

Curse of Dimensionality

- In high dimensional analysis problems
  - objects are sparse and dissimilar
    - most people haven't seen most movies
  - standard data structures and algorithms won't work
- Need to be more clever…

Sparse Representation

- For each movie, keep a list of users that watched it
- More like a NoSQL or MapReduce structure

Movie	Users
1	1,2,6,8
2	3
3	1,2,5,6,8
4	3,4,7,9
5	1,3,6
6	1,7
7	4,5
8	2,4
9	7,8
10	3

Minhash

- Jaccard similarity is slow to compute
  - Computing union AND intersection
  - Have to do this for ALL pairs
- Minhash is a fast way to estimate it
  - hash all items in sets
  - compare minimum hash value of two sets

Minhash

x	h(x)
1	3
2	4
3	5
4	7
5	6
6	9
7	8
8	1
9	2

"Inception" S={1,4,6,9}

"Up" T={1,2,3,5,7,8}

"Wall-E" U={1,3,5,8,9}

$$h_{min}(S) = \min(\{3, 7, 9, 2\}) = 2$$

$$h_{min}(T) = \min(\{3, 4, 5, 6, 8, 1\})$$

$$h_{min}(U) = ?$$

Minhash

- ***Amazing*** fact:
  - Probability minimum hashes are equal is ***exactly*** equal to the Jaccard similarity

$$P(h_{min}(S) = h_{min}(T)) = J(S,T)$$

How do they work?

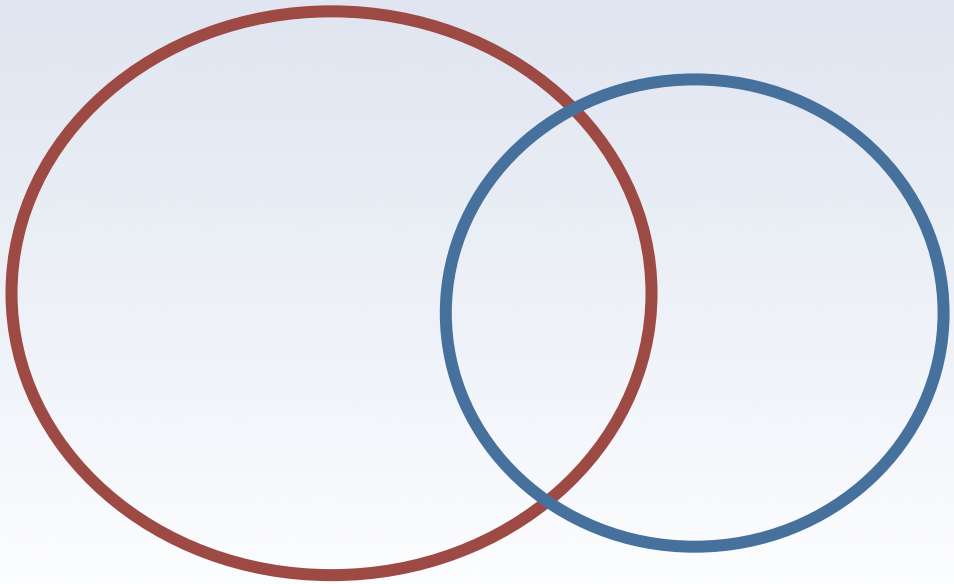
Minhash

- Computing similarity of “Up” and “Wall-E”
- Could do it like this…
- Or estimate it by sampling like this…

$$\frac{|S \cap T|}{|S \cup T|}$$

Minhash

- Like throwing darts at this diagram:



Minhash

- Hash creates a random permutation of all items
- Think about Netflix example:
  - Users lined up in a random order
  - Start at first person in line
  - Ask if they have seen either movie
  - Continue until we find someone who has

Minhash

- When we get to that person
  - Value is minhash of at least one movie
  - ***And*** a randomly sampled element of union
- If first user has seen both movies…
  - Value is minhash of both movies
  - ***And*** random sample is in intersection
- Probability they are equal: $\frac{|S \cap T|}{|S \cup T|}$

Minhash

- Fast estimate of Jaccard similarity
  - We need multiple samples for accuracy
  - Use *m* hash functions
- Simple to compute Minhash of a set:
  - Track *m* minima
  - For each item, compute *m* hashes, check if each one is current min
  - Return *m* minima

Minhash

- We call *m* minima our "signature"
  S={1,4,6,9}, U={1,3,5,8,9}

x	h1(x)
1	3
2	4
3	5
4	7
5	6
6	9
7	8
8	1
9	2

x	h2(x)
1	2
2	4
3	6
4	8
5	5
6	3
7	1
8	7
9	9

x	h3(x)
1	8
2	3
3	7
4	9
5	2
6	1
7	4
8	5
9	6

m=3

Signatures

S=[2,2,1]

U=[1,2,2]

J(S,U)=.29

Storing Minhash Signatures

- 2.5 million movies
- *m=10,000* signatures per movie
- Only 186 gigabytes!
- But…
  - We still have to compute similarity for all pairs of movies
  - 3.1 trillion pairs…

More cleverer

- We are only interested in items that are similar
- If two movies aren't similar, we'll ***never*** recommend them!

What if I told you…

Locality-Sensitive Hashing

- Hash function that maps similar items into the same bin with high probability
- We only have to compare pairs of items that are binned together
- Like hash join, but for similar sets

Locality-Sensitive Hashing

- Divide signature into *b* bands
  - bands contain *r* rows
- For each band, pick a hash function to map rows into *B* buckets

M1	M2	M3
3	3	1
4	1	2
1	1	3
2	3	3
3	3	2
1	1	1
1	1	2
2	2	3
3	3	4

Locality-Sensitive Hashing

0	1	2	3	4

0	1	2	3	4

0	1	2	3	4

M1	M2	M3
3	3	1
4	1	2
1	1	3
2	3	3
3	3	2
1	1	1
1	1	2
2	2	3
3	3	4

Locality-Sensitive Hashing

0	1	2	3	4
M2	M3		M1	

0	1	2	3	4

0	1	2	3	4

M1	M2	M3
3	3	1
4	1	2
1	1	3
2	3	3
3	3	2
1	1	1
1	1	2
2	2	3
3	3	4

Locality-Sensitive Hashing

0	1	2	3	4
M2	M3		M1	

0	1	2	3	4
	M1 M3	M2		

0	1	2	3	4
	M1 M2			M3

M1	M2	M3
3	3	1
4	1	2
1	1	3
2	3	3
3	3	2
1	1	1
1	1	2
2	2	3
3	3	4

Locality-Sensitive Hashing

- Now we compute similarity of pairs same bucket

Locality-Sensitive Hashing

0	1	2	3	4
M2	M3		M1	

0	1	2	3	4
	M1 M3	M2		

Sim(M1,M3)= 1/9

0	1	2	3	4
	M1 M2			M3

Sim(M1,M2)= 7/9

M1	M2	M3
3	3	1
4	1	2
1	1	3
2	3	3
3	3	2
1	1	1
1	1	2
2	2	3
3	3	4

Locality-Sensitive Hashing

- To avoid false positives, we should choose a large number of buckets *B*
- What is probability two items compared?
  - Assume no false positives
  - As a function of their actual similarity *s*
    - *s=J(S,T)*
  - And the number of rows *r* and bands *b*

Locality-Sensitive Hashing

- Probability items agree on all r rows:

$$s^r$$

- Probability items do not agree:

$$1 - s^r$$

- Probability items do not agree in b bands:

$$(1 - s^r)^b$$

Locality Sensitive Hashing

- Probability items do not agree in b bands:

$$(1 - s^r)^b$$

- Probability items agree on *at least* one band:

$$1 - ((1 - s^r)^b)$$

Example

- *m=10,000*
- *b=500*
- *r=20*
- *s=.5*

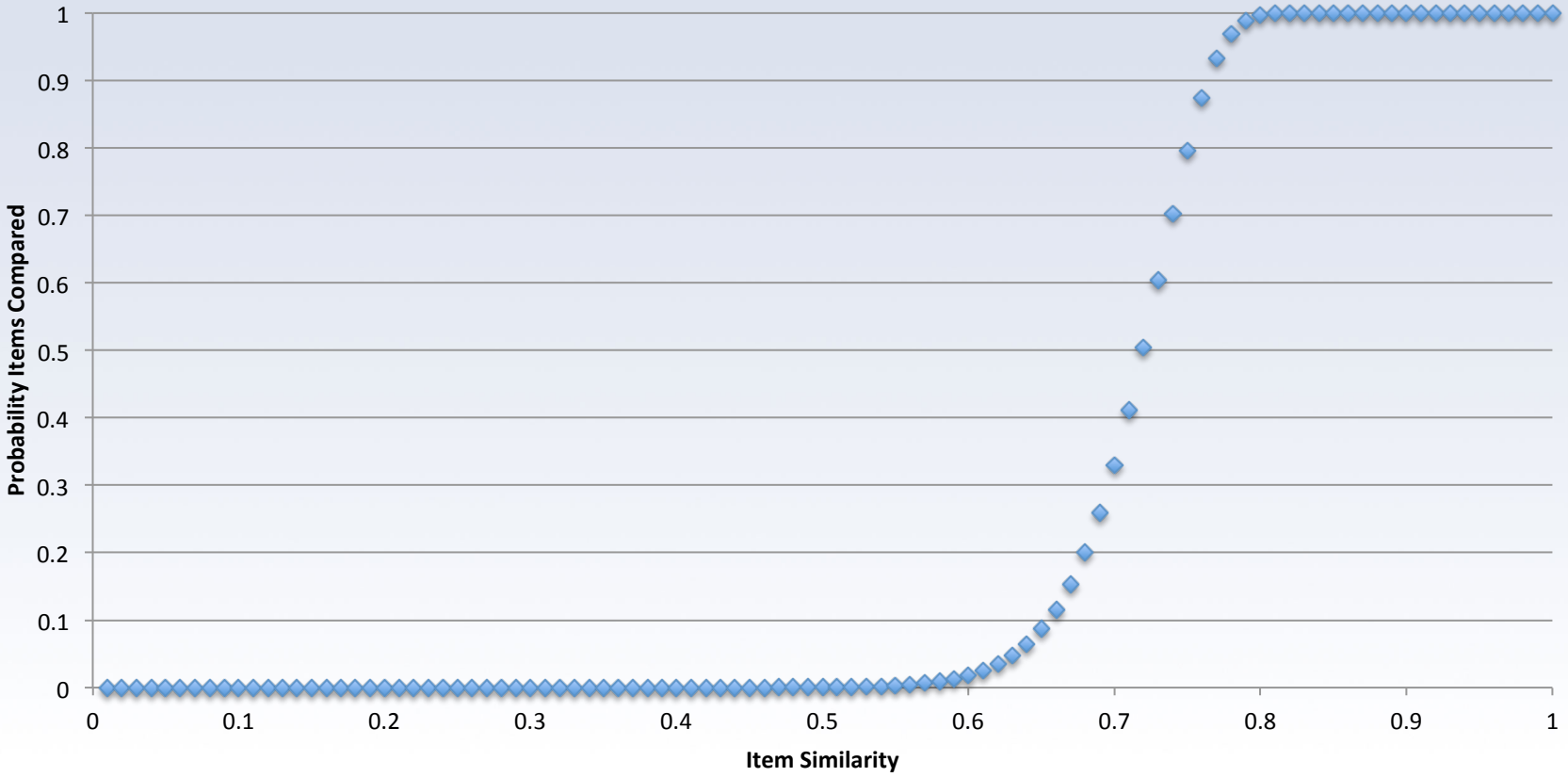
$$1 - ((1 - s^r)^b) = 1 - ((1 - .5^{20})^{500}) = .00048$$

Example

- *m=10,000*
- *b=500*
- *r=20*
- *s=.8*

$$1 - ((1 - s^r)^b) = 1 - ((1 - .8^{20})^{500}) = .997$$

Example


Probability Items Compared
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Item Similarity

Locality-Sensitive Hashing

- Only need to hash each of 2.5 million movies
- Of 3.1 trillion pairs of movies
  - most are not similar
  - small *s* implies very low chance to compare

Entire Process

1. Compute minhash signatures for all 2.5 million movies (186 Gb)
2. Process *b=500* bands in each signature, hash in to *B=10 trillion* buckets
   – Use extensible or linear hash tables (a few Gb)
3. For each non-empty bucket, compute actual similarities using full signatures

Summary

- Recommender Systems
- Collaborative Filtering
- Minhashing
- Locality-Sensitive Hashing