Intro to Recommendations Systems

Data Science and Machine Learning Workshop' 2017

Habib University

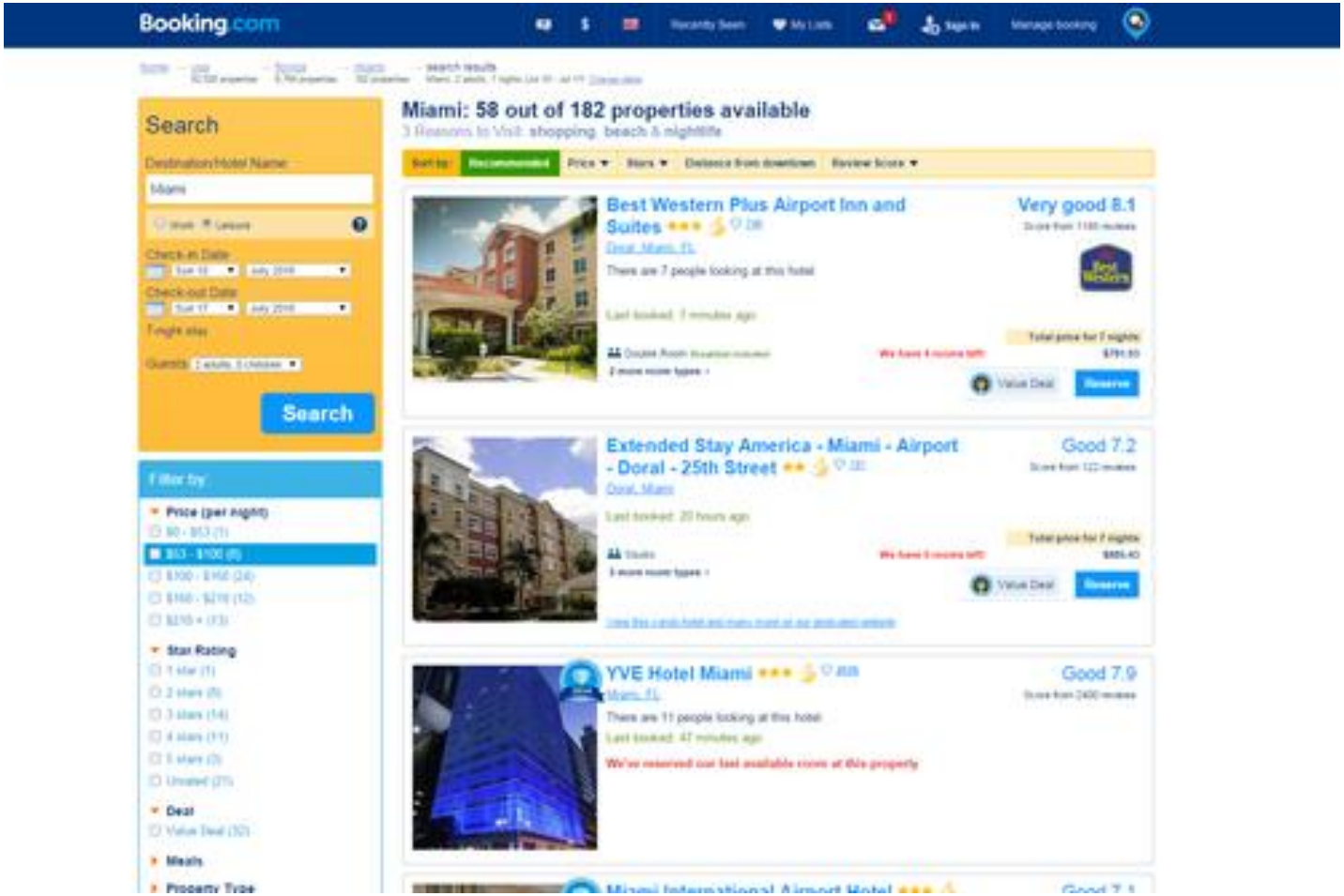
Acknowledgement

- Slides of this lecture have been taken from following online resources:
  - https://www.slideshare.net/stanleywanguni/overview-of-recommender-system
  - http://katbailey.github.io/post/matrix-factorization-with-tensorflow/

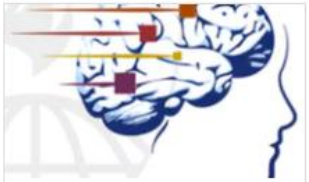
Amazon


Grant, Welcome to Your Amazon.com (If you're not Grant Ingersoll, click here.)
Today's Recommendations For You
Here's a daily sample of items recommended for you. Click here to see all recommendations.
Data Mining
LOOK INSIDE!
PYTHON
IN A NUTSHELL
LOOK INSIDE!
Principles of Data Mining (A...
by David J....
(17) $52.00
Python in a Nutshell, Secon...
by Alex Mart...
(40) $26.39
Introductory Statistics wit...
by Peter Dal...
(20) $48.56

Booking.com


Booking.com
Search
Destination/Hotel Name:
Miami
Check-in Date:
Check-out Date:
Search
Miami: 58 out of 182 properties available
Best Western Plus Airport Inn and Suites
Very good 8.1
Extended Stay America - Miami - Airport - Doral - 25th Street
Good 7.2
YVE Hotel Miami
Good 7.9

Coursera

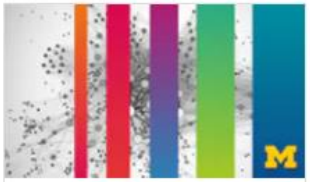


Johns Hopkins University

Data Science

University of Michigan

Python for Everybody



University of Michigan

Applied Data Science with Python

Duke University

Excel to MySQL: Analytic Techniques for Business

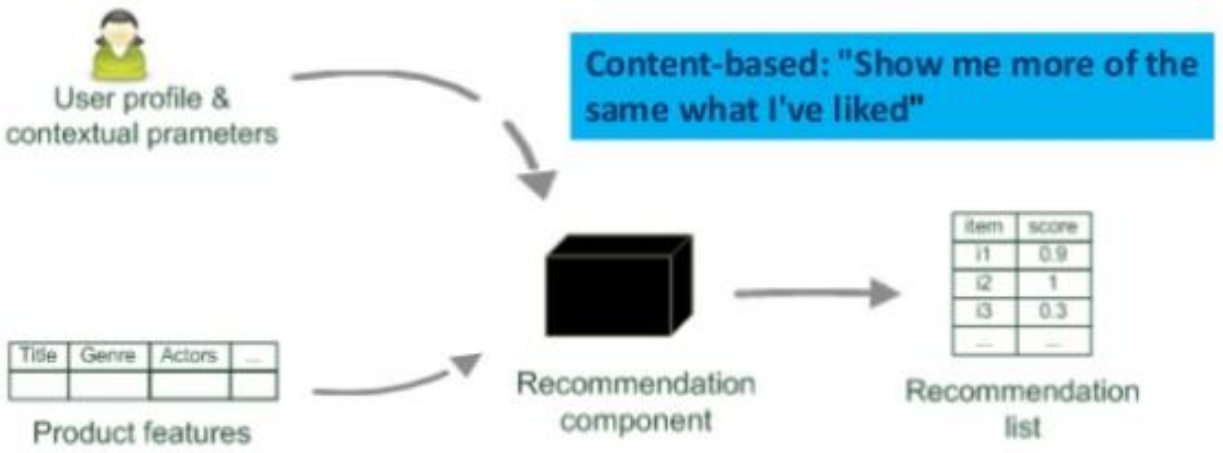
Duke University

Java Programming and Software Engineering Fundamentals

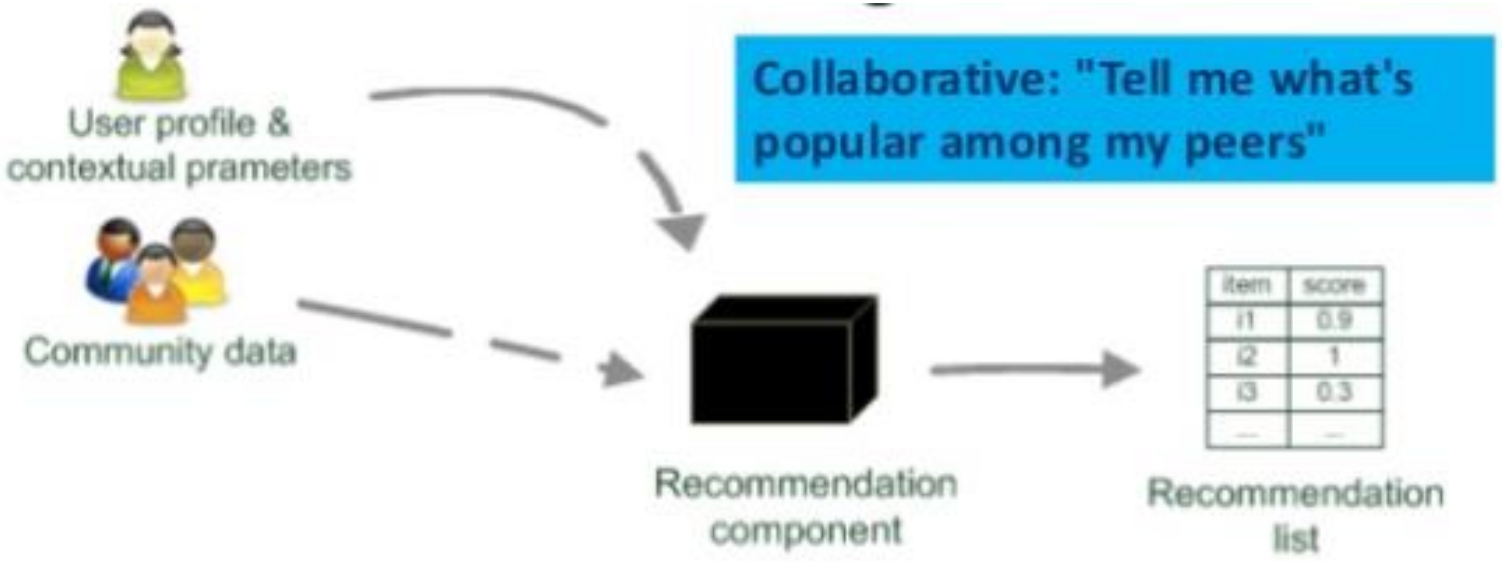
Data

- Item Data
- User Data
- Sales Data
- Ranking Data (Implicit/Explicit)

Content Based Recommendations



Collaborative Filtering (CF)



Collaborative Filtering

User Based Collaborative Filtering

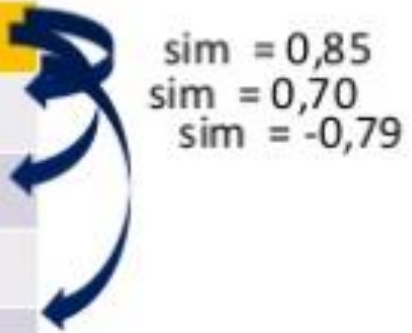
- what people with similar tastes seem to like

Item Based Collaborative Filtering

- what other items are similar to what user liked

Nearest Neighbor CF

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1



Item Based CF

Basic idea:

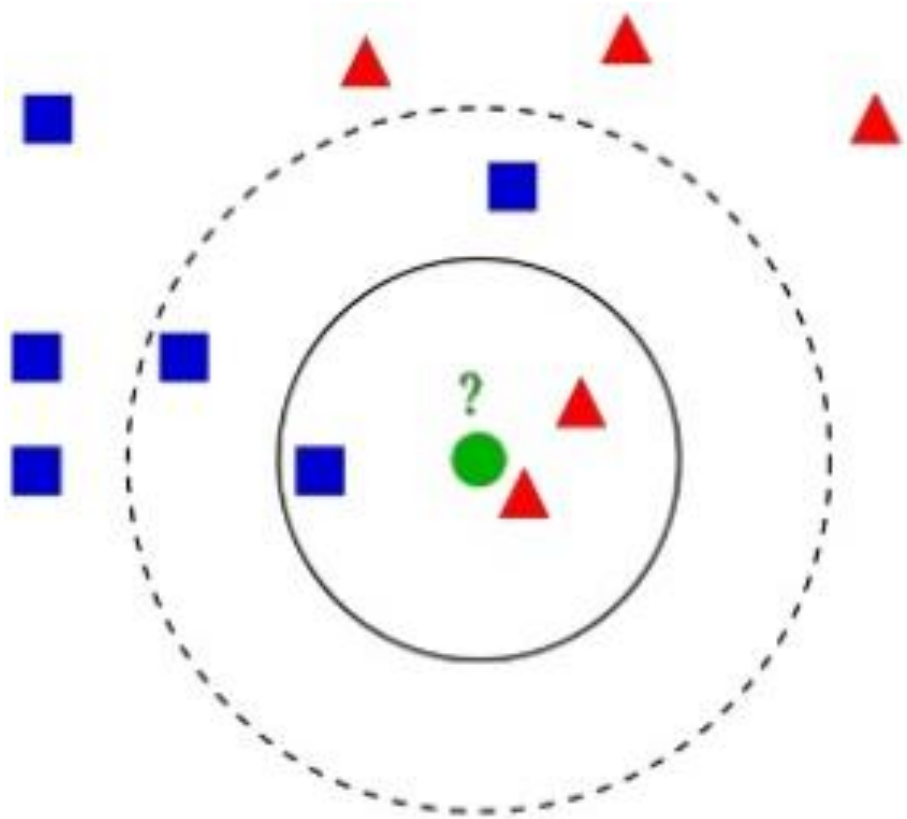
- **Use the similarity between items (and not users) to make predictions**

Example:

- **Look for items that are similar to Item5**
- **Take Alice's ratings for these items to predict the rating for Item5**

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

K- Nearest Neighbor



Measuring Similarity

- Jaccard coefficient:

$$sim(a,b) = \frac{(1+1)}{(1+1+1)+(1+1+1+1)-(1+1)}$$

- Cosine similarity:

$$sim(a,b) = \cos(\vec{a},\vec{b}) = \frac{\vec{a}\bullet\vec{b}}{\|\vec{a}\|_2 * \|\vec{b}\|_2} = \frac{(1*1+0.5*1)}{\sqrt{(1^2+0.5^2+1^2)*(1^2+0.5^2+1^2+1^2)}}$$

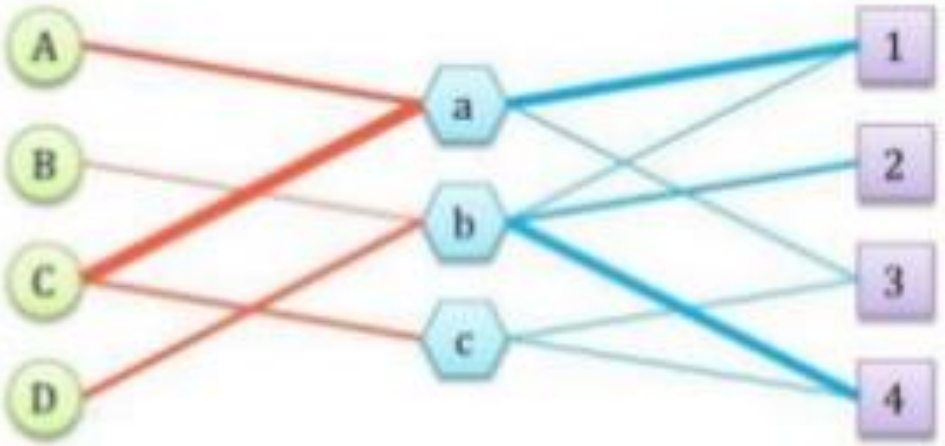
- Pearson Correlation:

$$corr(a,b) = \frac{\sum_i (r_{ai}-\overline{r_a})(r_{bi}-\overline{r_b})}{\sqrt{\sum_i (r_{ai}-\overline{r_a})^2 \sum_i (r_{bi}-\overline{r_b})^2}} = \frac{m\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{m\sum a_i^2 - (\sum a_i)^2}\sqrt{m\sum b_i^2 - (\sum b_i)^2}}$$

$$= \frac{match\_cols * Dotprod(a,b) - sum(a) * sum(b)}{\sqrt{match\_cols * sum(a^2) - (sum(a))^2}\sqrt{match\_cols * sum(b^2) - (sum(b))^2}}$$

Latent Factor Model

- Users and items are connect by latent features.



User-Item Rating

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0?	3	0?	3	0?
User 2	4	0?	0?	2	0?
User 3	0?	0?	3	0?	0?
User 4	3	0?	4	0?	3
User 5	4	3	0?	4	0?

Matrix Factorization

	Feature 1	Feature 2
User 1	?	?
User 2	?	?
User 3	?	?
User 4	?	?
User 5	?	?

X

	Item 1	Item 2	Item 3	Item 4	Item 5
Feature 1	?	?	?	?	?
Feature 2	?	?	?	?	?

=

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0?	3	0?	3	0?
User 2	4	0?	0?	2	0?
User 3	0?	0?	3	0?	0?
User 4	3	0?	4	0?	3
User 5	4	3	0?	4	0?

http://katbailey.github.io/post/matrix-factorization-with-tensorflow/

Libraries for Recommendations

- Python
  - GraphLab
  - Crab.
  - Surprise
  - Python Recsys
  - MRec
- Java
  - Mahout
-

Challenges

- Sparsity
- Synonymy
- Scalability
- Gray Sheep
  - refers to the users whose opinions do not consistently agree or disagree with any group of people
- Privacy