

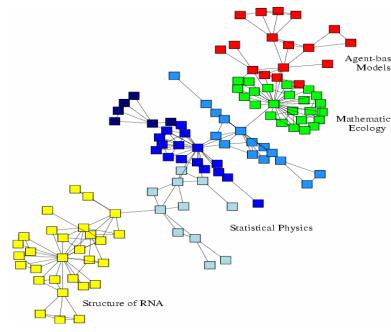
Personalised Data Analytics

Personalization with Recommender Systems

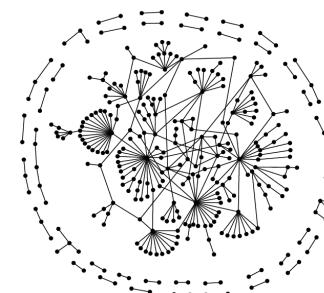
Last Week Recap: Many Data are Networks



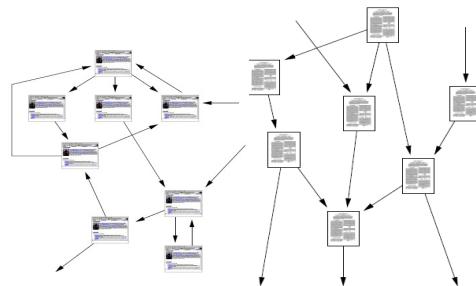
Social networks



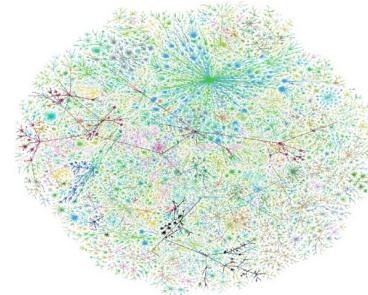
Economic networks



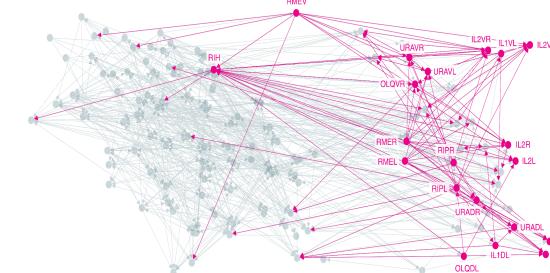
Biomedical networks



Information networks:
Web & citations



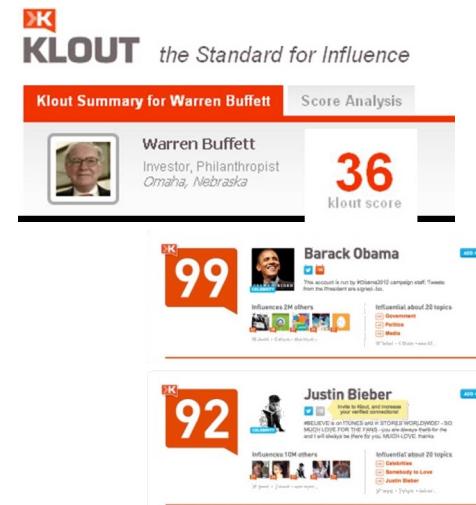
Internet



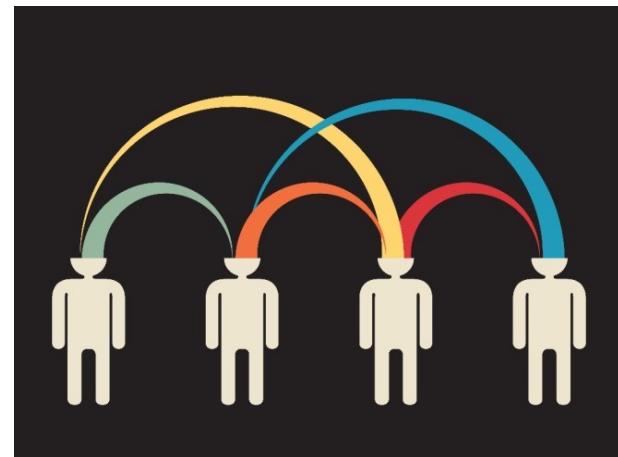
Networks of neurons

Last Week Recap: Centrality

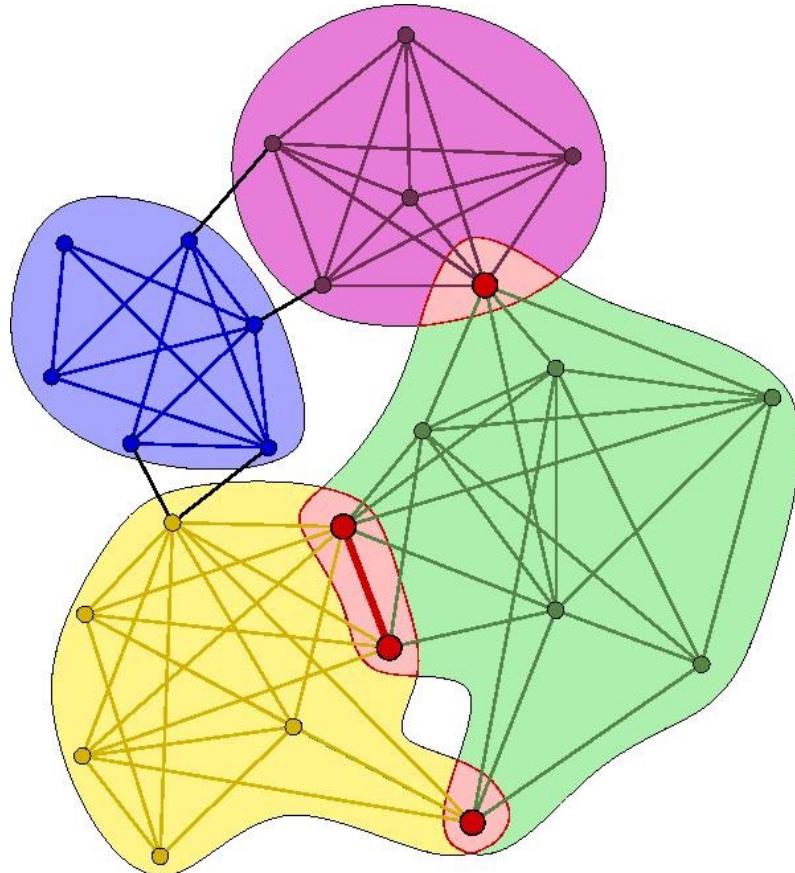
- ❖ What is centrality?
 - Centrality defines **how important** an actor is within a network



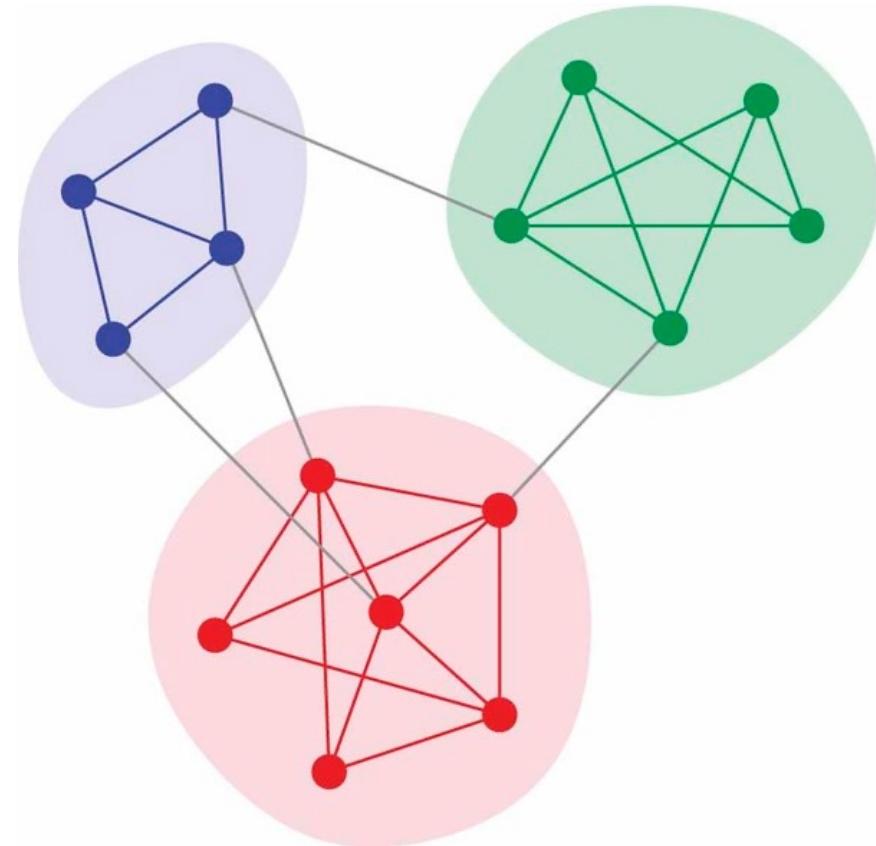
- ❖ Why centrality? a measure of **influence**
 - The act or power of producing an effect without apparent exertion of force or direct exercise of command



Last Week Recap: Community



Overlapping Communities



Disjoint Communities

Last Week Recap: Information Diffusion

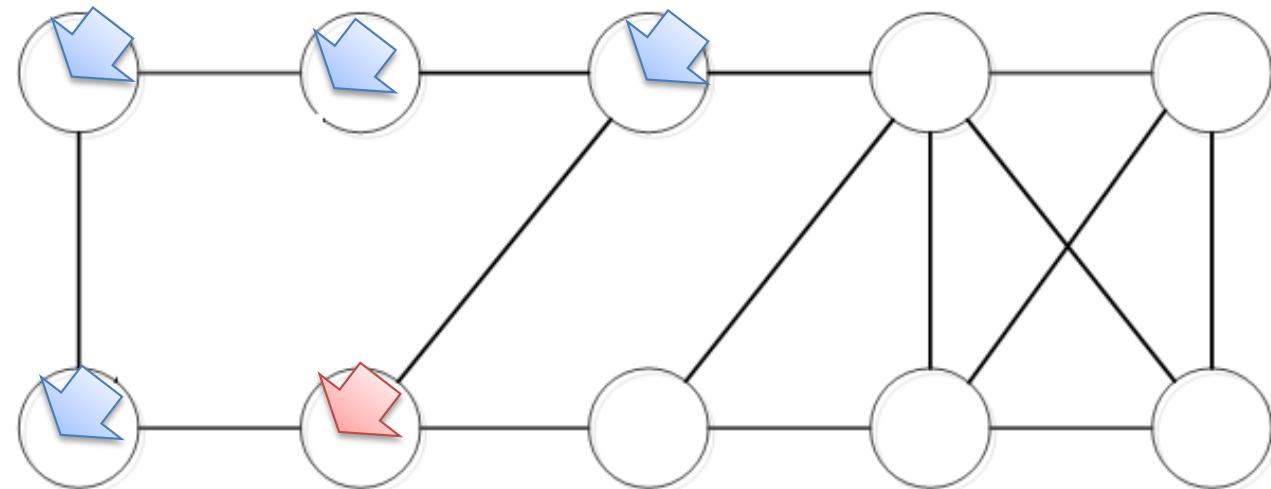


seed

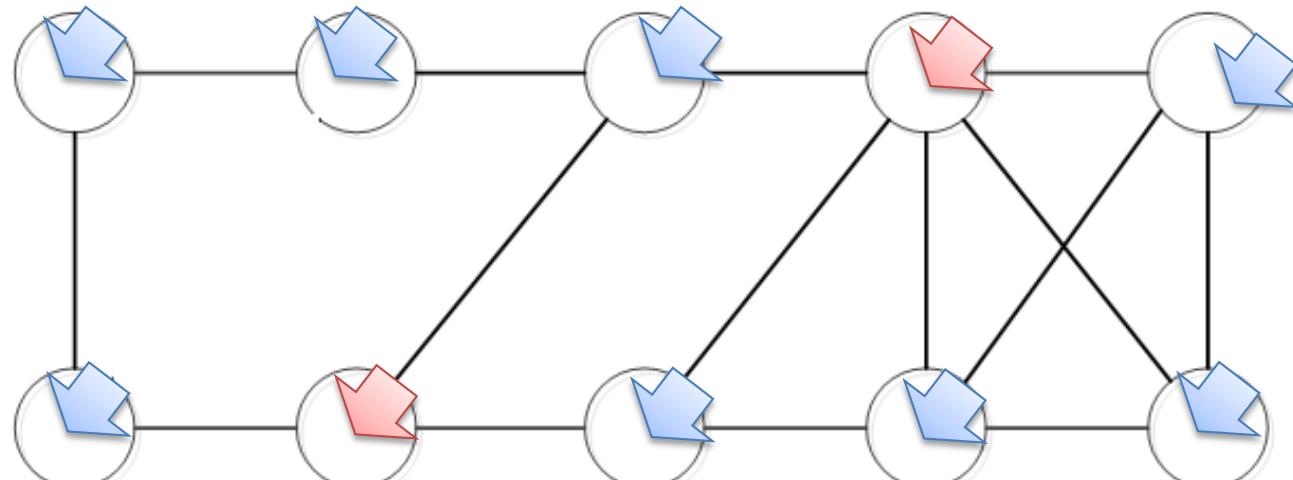


activated nodes

k=1 seed



k=2 seeds



Course structure

W1. Data Processing with Python

W2. Data Exploration with Python

W3. Data Modeling with Pytyhon

W4. Data Analytics for Timeseries

Holiday

W5-6-7. Data Analytics for Texts

W8. Data Analytics for Images

W9. Data Analytics for Graphs

W10-11. Data Analytics for Other Data

W12. Revision

Personalized Data Analytics

- I. Why Personalization?
- II. Recommender Systems
 - 1. Content-based Recommendation
 - 2. Rating-based Recommendation (Collaborative Filtering)
 - 3. Hybrid Recommendation
 - 4. Clustering-based Recommendation
- III. Performance measures

I. PERSONALIZATION

No Personalization

Stores



Bag



Sign In/Up

Online Exclusives **Home & Living** Tech Toys Womens Mens Kids & Baby Beauty Sport & Outdoor Catalogue

All Home & Living	Home & Living Latest Arrivals	>	Pets	>
Home by Category	> Home & Living Online Exclusives	>	Rugs	>
Home by Room	> Appliances	>	<u>Stationery & Office Supplies</u>	>
Home & Living Clearance	> Art & Craft	>	Storage & Organisation	>
Party, Cards & Wrap	> Bedding	>	Towels	>
Features	> Books	>		
	Dining	>		
	Furniture	>		
	Home Decor	>		
	Home & Car Maintenance	>		
	Independent Living	>		
	Lighting	>		

- All users see the **same** list of products
- Need to browse a lot to find **what they want**

Why Personalization?

Customers have higher than ever expectations.

75%

of consumers are more likely to buy from a retailer that recognizes them by name, recommends options based on past purchases, or knows their purchase history

Why Personalization?

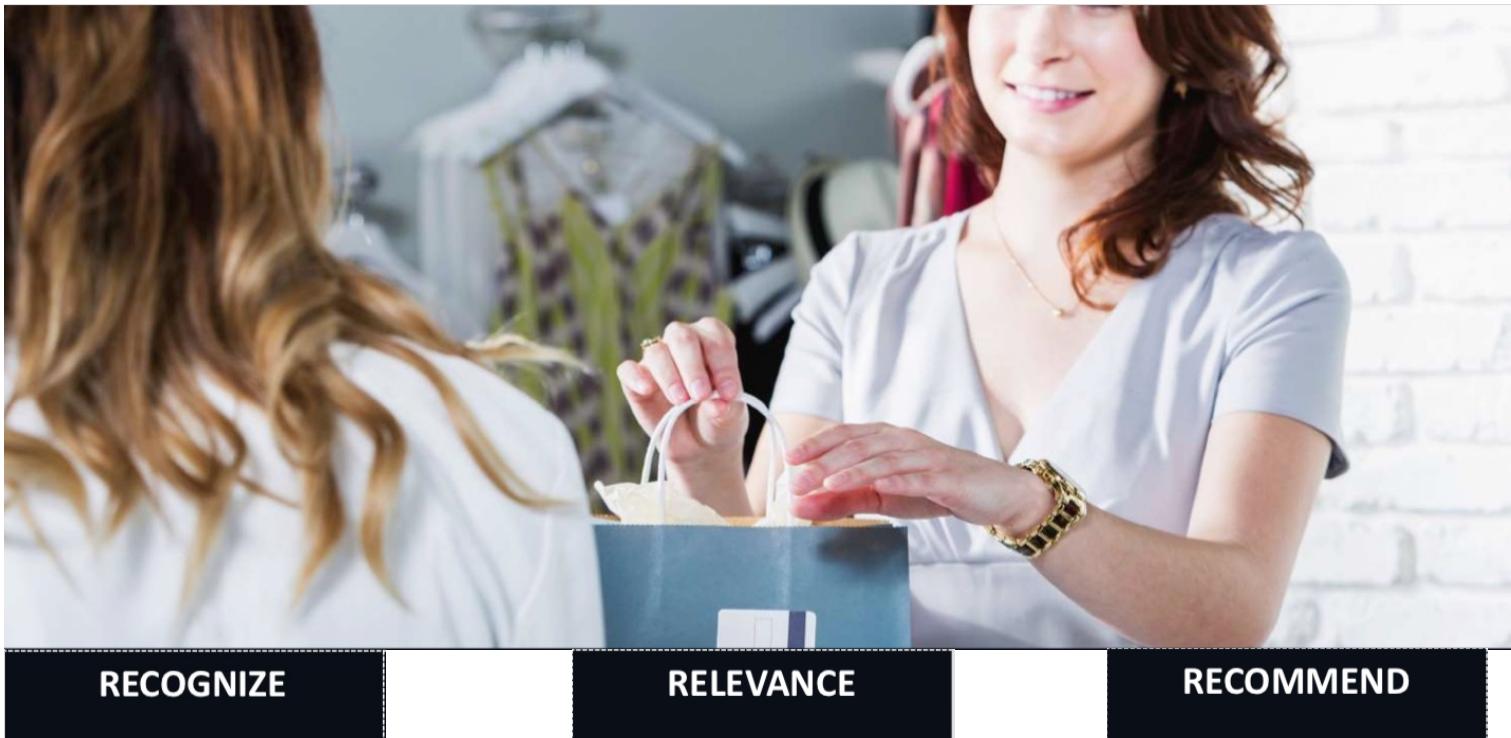


What is Personalization?

The goal of Personalization is to **use** data to make it **easier** for customers to **find & consume** what they want, how and when they want.



3 dimensions of Personalization



Personalization Algorithms

- ❖ **Recognize:** classification, clustering, community analysis etc.
- ❖ **Relevance:** information retrieval, search engine, information diffusion etc.
- ❖ **Recommend:** Recommender Systems

II. RECOMMENDER SYSTEMS

What is Recommendation?

- ❖ Which mobile phone should I buy?
- ❖ Where should I visit for my business trip?
- ❖ Whom should I follow on Twitter?
- ❖ Where should I invest my money?



- ❖ Which tour is the best for our class?



Recommender Systems - Applications

Book recommendation in Amazon

The screenshot shows a product page for 'Networks: An Introduction' by Mark Newman. It includes sections for 'Frequently Bought Together' and 'Customers Who Bought This Item Also Bought'. The 'Also Bought' section is circled in red.

Video clip recommendation in YouTube

The screenshot shows a YouTube search results page for a wildfire in Arizona. The 'Suggestions' sidebar is circled in red, showing recommended videos like 'Schultz Fire - Flagstaff, AZ - June 20, 2010' and 'Schultz Wildfire'.

Product Recommendation in ebay

The screenshot shows an eBay product page for a 3000mAh camera battery. It features a 'Recommendations for you' section with various products like Dr. Seuss books and AAA batteries.

Restaurant Recommendation in Yelp

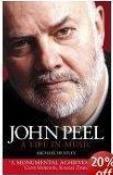
The screenshot shows a Yelp search results page for 'Tempo' in Tempe, AZ. It includes a map view showing the restaurant's location relative to other landmarks and roads.

recommendation = **personalized** prediction

The value of recommendations

- ❖ Netflix: 2/3 of the **movies** watched are recommended
- ❖ Google News: recommendations generate 38% more click through
- ❖ Amazon: 35% **sales** from recommendations
- ❖ Choicestream: 28% of the people would buy more **music** if they found what they liked.

John Peel: A Life in Music
Michael Heatley



List Price: £6.99
Our Price: £5.59 & eligible for **Free UK delivery** on orders over £15 with Super Saver Delivery. See [details & conditions](#).
You Save: £1.40 (20%)

Availability: usually dispatched within 24 hours.

27 Used & New from £1.60
[See larger photo](#)

Edition: Paperback

[More Product Details](#)

Perfect Partner
Buy **John Peel: A Life in Music** with **Margrave Of The Marshes** today!



Total List Price: £25.98
Buy Together Today: £16.98

[Buy both now](#)

Customers who bought this item also bought:

- [The Little Book of Wanking: The Definitive Guide to Man's Ultimate Relief](#); Paperback ~ Dick Palmer
- [\(Shag Yourself Slim\) The Most Enjoyable Way to Lose Weight](#); Paperback ~ Imah Goer
- [Grumpy Old Men, the Official Handbook](#); Hardcover ~ Stuart Prebble
- [The Little Book of Minge Topiary](#); Paperback ~ Michael O'Mara Books Ltd

READY TO BUY

[Add to Shopping Bag](#) or [sign in to turn on 1-Click](#)

MORE BUYING OPTIONS

22 New from **5 used** from ...
Have one to sell?

[Shopping with Guarantees](#)

[Add to Wish List](#)
(We'll set one up for you)

[View my Wish List](#)

Japan Halts US Beef Imports After Banned Meat Found (Update)

Bloomberg - 1 hour ago
Jan. 20 (Bloomberg) -- Japan stopped imports of beef from the US after inspectors found banned cattle parts in a shipment, disrupting trade that resumed last month following a two-year halt because of mad-cow disease.

Japan halts US beef imports due to fear of mad cow San Diego Union Tribune

US to probe beef shipment Japan San Jose Mercury News

Boston Globe - Guardian Unlimited - MarketWatch - CNN - [all 1,045 related »](#)

Recommended for aprice@gmail.com »

Serena in denial over her terminal decline

Guardian Unlimited - 7 hours ago - It was in Australia eight years ago that the Williams sisters were seen competing at the same grand slam for the first time.

International Herald Tribune - TennisReporters.net - Forbes - [all 319 related »](#)

2 dozen hurt in Tel Aviv bombing

San Francisco Chronicle - 20 hours ago - Jerusalem -- At least two dozen Israelis were wounded Thursday when a suicide bomber detonated explosives he was ...

Los Angeles Times - Detroit Free Press - San Jose Mercury News - [all 836 related »](#)

US plans to shift diplomats to developing countries

Boston Globe - Jan 19, 2006 - By Farah Stockman, Globe Staff | January 19, 2006. WASHINGTON -- Secretary of State Condoleezza Rice announced ...

International Herald Tribune - Sydney Morning Herald - Financial Times - [all 70 related »](#)

Phone Cancer Link Downplayed

Red Herring - all 170 related »

'American Idol' Gets a Little Mean

Ceres Courier - all 5/5 related »

Deadline to kill US journalist passes with no news

Khaleej Times - all 2,958 related »

From here, Oscar race goes inside Hollywood

Reuters - all 114 related »

NASA starry-eyed at comet's samples

Houston Chronicle - all 219 related »

REGION: Annan urges Iran to resume talks with EU

Daily Times - all 1,382 related »

In The News

Osama bin Laden - [Midnight Hour](#)

Albert Brooks - [Mustans Sally](#)

Tel Aviv - [Air Sahara](#)

Jet Airways - [Mehmet Ali Agca](#)

Wilson Pickett - [Jill Carroll](#)

Recommender Systems - Problem

- ❖ Estimate a utility function that automatically **predicts how a user will like an item**
 - Formally, a recommender system takes a set of users U and a set of items I and learns a function f such that: $f: U \times I \rightarrow R$
 - For each user $u \in U$, we want to choose an item $i \in I$ that maximize f .
- ❖ Based on:
 - **Past behavior**
 - **Relations to other users**
 - **Item similarity**
 - **Context**
 - ...

Types of Recommender Systems

1. Content-based recommendation:
 - Recommend **based on similarity** between user features and item features
2. Rating-based recommendation (Collaborative Filtering)
 - Recommend **based on rating** matrix
3. Clustering-based recommendation
 - Recommend **based on clusters of rating** matrix

1. Content-based recommendation

- ❖ Assumption: a **user's interest** should match the description of the items that the user should be recommended by the system.
 - The more similar the item's description to that of the user's interest, the more likely that the user finds the item's recommendation interesting.
- ❖ Goal: **find the similarity** between the user and all of the existing items is the core of this type of recommender systems

Content-based Recommendation: An Example

The screenshot shows the 'Edit Favorites' section of the Amazon.com website. At the top, there's a search bar with 'Amazon.com' and a dropdown, followed by a yellow flower icon labeled 'Find Gifts' and a 'Web Search' button. The main area is titled 'Edit Favorites' and has a sub-section 'Your Books Favorites'. It includes a 'Categories' section with checkboxes for 'Biographies & Memoirs', 'Business & Investing', 'Computers & Internet', 'Nonfiction', and 'Outdoors & Nature'. Below this is an 'Add to Your Favorites' section with checkboxes for various categories like Arts & Photography, Children's Books, Comics & Graphic Novels, etc.

Items recommended

User profile

The screenshot shows the 'Recommended For You' section for the 'Books' category. The top navigation bar is identical to the previous screenshot. The main content area is titled 'Recommended For You > Books'. It says 'These recommendations are based on items you own and more.' and lists 'view: All | New Releases | Coming Soon | More results'. The first recommendation is for the book 'The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture' by John Battelle, with a price of \$16.35 and a 'Used & new from \$10.95' link. The second recommendation is for 'Writing Successful Science Proposals' by Andrew J. Friedland, Carol L Folt, with a price of \$14.95. There are buttons for 'Add to cart' and 'Add to Wish List' for each item. A note at the bottom says 'Recommended because you purchased Amazonia and more (edit)'.

Content-based Recommendation: Information Retrieval Approach

- ❖ We represent user profiles and item descriptions by vectorizing them using a set of k keywords (e.g., using **TF-IDF**)

$$I_j = (i_{j,1}, i_{j,2}, \dots, i_{j,k})$$
$$U_i = (u_{i,1}, u_{i,2}, \dots, u_{i,k})$$

- ❖ Compute their (cosine) similarity

$$\text{sim}(U_i, I_j) = \text{cos}(U_i, I_j) = \frac{\sum_{l=1}^k u_{i,l} i_{j,l}}{\sqrt{\sum_{l=1}^k u_{i,l}^2} \sqrt{\sum_{l=1}^k i_{j,l}^2}}$$

- ❖ Recommend the top most similar items to the user

Algorithm Content-based recommendation

Require: User i 's Profile Information, Item descriptions for items $j \in \{1, 2, \dots, n\}$, k keywords, r number of recommendations.

- 1: **return** r recommended items.
 - 2: $U_i = (u_1, u_2, \dots, u_k)$ = user i 's profile vector;
 - 3: $\{I_j\}_{j=1}^n = \{(i_{j,1}, i_{j,2}, \dots, i_{j,k})\}$ = item j 's description vector $\}_{j=1}^n$;
 - 4: $s_{i,j} = \text{sim}(U_i, I_j), 1 \leq j \leq n$;
 - 5: Return top r items with maximum similarity $s_{i,j}$.
-

Content-based RecSys: Summary

- (+) Not depend on other users
- (+) Lower calculation cost
- (+) Can recommend niche items that very few other users are interested in

- (-) Hard to collect description of item, need hand-engineered features
- (-) Limited ability to expand on the users' existing interests.

2. Rating-based Recommendation

- ❖ AKA Collaborative filtering: recommend items by only **users' past behavior**
- ❖ Advantage: we don't need to have additional information about the users or content of the items
 - Users' **rating or purchase history** is the only information that is needed to work
- ❖ Input: Rating matrix
 - Users rate (rank) items (purchased, watched)
 - Explicit ratings: entered by a user directly
 - Implicit ratings: inferred from other user behavior
 - e.g. the amount of time users spent on a webpage
 - e.g. the number of times users listen to a song

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	
Love at last	5	5	0	6	
Romance forever	5	?	?	0	
Cute puppies of love	?	4	0	?	
Nonstop car chases	0	0	5	4	
Swords vs. karate	0	0	5	?	

2.1. Nearest-Neighbour Approach

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

User-based

- ❖ Find similar users to me and recommend what they liked
- ❖ Why?: users with similar **previous** ratings for items are likely to rate future items similarly

Item-based

- ❖ Find similar items to those that I have previously liked
- ❖ Why?: Items that have received similar ratings **previously** from users are likely to receive similar ratings from future users

Nearest-Neighbour Approach (Cont'd)

User-based

1. Weigh all **users** with respect to their similarity with the current **user**
2. Select a subset of the **users** (e.g. top-k neighbors) as recommenders
3. Predict the rating of the user for specific items using neighbors' ratings for the same (or similar) **users**
4. Recommend items with the highest predicted rank

Item-based

1. Weigh all **items** with respect to their similarity with the current **item**
2. Select a subset of the **items** (e.g. top-k neighbors) as recommenders
3. Predict the rating of the user for specific items using neighbors' ratings for the same (or similar) **items**
4. Recommend items with the highest predicted rank

User-based vs. Item-based CF

- ❖ Finds the **most similar users** to the current user
- ❖ Cosine Similarity:

$$\begin{aligned} sim(u_1, u_2) &= \frac{u_1 \cdot u_2}{\|u_1\| \cdot \|u_2\|} \\ &= \frac{\sum_i r_{1,i} r_{2,i}}{\sqrt{\sum_i r_{1,i}^2} \sqrt{\sum_i r_{2,i}^2}} \end{aligned}$$

- ❖ Pearson Similarity (Correlation Coefficient):

$$\begin{aligned} sim(u_1, u_2) &= \frac{\sum_i (r_{1,i} - \bar{r}_{u_1})(r_{2,i} - \bar{r}_{u_2})}{\sqrt{\sum_i (r_{1,i} - \bar{r}_{u_1})^2} \sqrt{\sum_i (r_{2,i} - \bar{r}_{u_2})^2}} \end{aligned}$$

- ❖ Finds the **most similar items** to the current item
- ❖ Cosine Similarity:

$$\begin{aligned} sim(i_1, i_2) &= \frac{i_1 \cdot i_2}{\|i_1\| \cdot \|i_2\|} \\ &= \frac{\sum_u r_{u,1} r_{u,2}}{\sqrt{\sum_u r_{u,1}^2} \sqrt{\sum_u r_{u,2}^2}} \end{aligned}$$

- ❖ Pearson Similarity (Correlation Coefficient):

$$\begin{aligned} sim(i_1, i_2) &= \frac{\sum_u (r_{u,1} - \bar{r}_{i_1})(r_{u,2} - \bar{r}_{i_2})}{\sqrt{\sum_u (r_{u,1} - \bar{r}_{i_1})^2} \sqrt{\sum_u (r_{u,2} - \bar{r}_{i_2})^2}} \end{aligned}$$

User-based vs. Item-based CF

❖ Update the ratings:

$$r_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u, v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} sim(u, v)}$$

$r_{u,i}$: predicted rating of user u for item i

\bar{r}_u : user u's mean rating

\bar{r}_v : user v's mean rating

$r_{v,i}$: observed rating of user v for item i

❖ Update the ratings:

$$r_{u,i} = \bar{r}_i + \frac{\sum_{j \in N(i)} sim(i, j)(r_{u,j} - \bar{r}_j)}{\sum_{j \in N(i)} sim(i, j)}$$

\bar{r}_i : item i's mean rating

\bar{r}_j : item j's mean rating

User-based CF, Example

	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

Predict Jane's rating
for Aladdin

1- Calculate average ratings

$$\bar{r}_{John} = \frac{3 + 3 + 0 + 3}{4} = 2.25$$

$$\bar{r}_{Joe} = \frac{5 + 4 + 0 + 2}{4} = 2.75$$

$$\bar{r}_{Jill} = \frac{1 + 2 + 4 + 2}{4} = 2.25$$

$$\bar{r}_{Jane} = \frac{3 + 1 + 0}{3} = 1.33$$

$$\bar{r}_{Jorge} = \frac{2 + 2 + 0 + 1}{4} = 1.25$$

2- Calculate user-user similarity

$$sim(Jane, John) = \frac{3 \times 3 + 1 \times 3 + 0 \times 3}{\sqrt{10} \sqrt{27}} = 0.73$$

$$sim(Jane, Joe) = \frac{3 \times 5 + 1 \times 0 + 0 \times 2}{\sqrt{10} \sqrt{29}} = 0.88$$

$$sim(Jane, Jill) = \frac{3 \times 1 + 1 \times 4 + 0 \times 2}{\sqrt{10} \sqrt{21}} = 0.48$$

$$sim(Jane, Jorge) = \frac{3 \times 2 + 1 \times 0 + 0 \times 1}{\sqrt{10} \sqrt{5}} = 0.84$$

User-based CF, Example (cont'd)

3- Calculate Jane's rating for Aladdin, Assume that neighborhood size = 2

$$\begin{aligned} r_{Jane,Aladdin} &= \bar{r}_{Jane} + \frac{sim(Jane, Joe)(r_{Joe,Aladdin} - \bar{r}_{Joe})}{sim(Jane, Joe) + sim(Jane, Jorge)} \\ &\quad + \frac{sim(Jane, Jorge)(r_{Jorge,Aladdin} - \bar{r}_{Jorge})}{sim(Jane, Joe) + sim(Jane, Jorge)} \\ &= 1.33 + \frac{0.88(4 - 2.75) + 0.84(2 - 1.25)}{0.88 + 0.84} = 2.33 \end{aligned}$$

Item-based CF, Example

1- Calculate average ratings

$$\bar{r}_{Lion\ King} = \frac{3 + 5 + 1 + 3 + 2}{5} = 2.8$$

$$\bar{r}_{Aladdin} = \frac{0 + 4 + 2 + 2}{4} = 2.$$

$$\bar{r}_{Mulan} = \frac{3 + 0 + 4 + 1 + 0}{5} = 1.6$$

$$\bar{r}_{Anastasia} = \frac{3 + 2 + 2 + 0 + 1}{5} = 1.6$$

2- Calculate item-item similarity

$$sim(Aladdin, Lion\ King) = \frac{0 \times 3 + 4 \times 5 + 2 \times 1 + 2 \times 2}{\sqrt{24} \sqrt{39}} = 0.84$$

$$sim(Aladdin, Mulan) = \frac{0 \times 3 + 4 \times 0 + 2 \times 4 + 2 \times 0}{\sqrt{24} \sqrt{25}} = 0.32$$

$$sim(Aladdin, Anastasia) = \frac{0 \times 3 + 4 \times 2 + 2 \times 2 + 2 \times 1}{\sqrt{24} \sqrt{18}} = 0.67$$

3- Calculate Jane's rating for Aladdin, Assume that neighborhood size = 2

$$\begin{aligned} r_{Jane, Aladdin} &= \bar{r}_{Aladdin} + \frac{sim(Aladdin, Lion\ King)(r_{Jane, Lion\ King} - \bar{r}_{Lion\ King})}{sim(Aladdin, Lion\ King) + sim(Aladdin, Anastasia)} \\ &\quad + \frac{sim(Aladdin, Anastasia)(r_{Jane, Anastasia} - \bar{r}_{Anastasia})}{sim(Aladdin, Lion\ King) + sim(Aladdin, Anastasia)} \\ &= 2 + \frac{0.84(3 - 2.8) + 0.67(0 - 1.6)}{0.84 + 0.67} = 1.40 \end{aligned}$$

2.2. Rating-based Recommendation: Embedding/Latent Approach (OPTIONAL)

- ❖ Assume each item has a feature vector of d dimensions
 - But don't know what these features are
- ❖ Similarly, each user also has a preference vector of d dimensions
 - Each user has different preferences on these features

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

2.2.1. Dual Optimization Method

Minimizing $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ simultaneously

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) =$$

$$\frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

Want:

$$\min_{\substack{x^{(1)}, \dots, x^{(n_m)} \\ \theta^{(1)}, \dots, \theta^{(n_u)}}} J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$$

Dual Optimization Method (cont'd)

Iterative Algorithm:

1. Initialize $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$ to small random values.
2. Minimize $J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization algorithm). E.g. for every

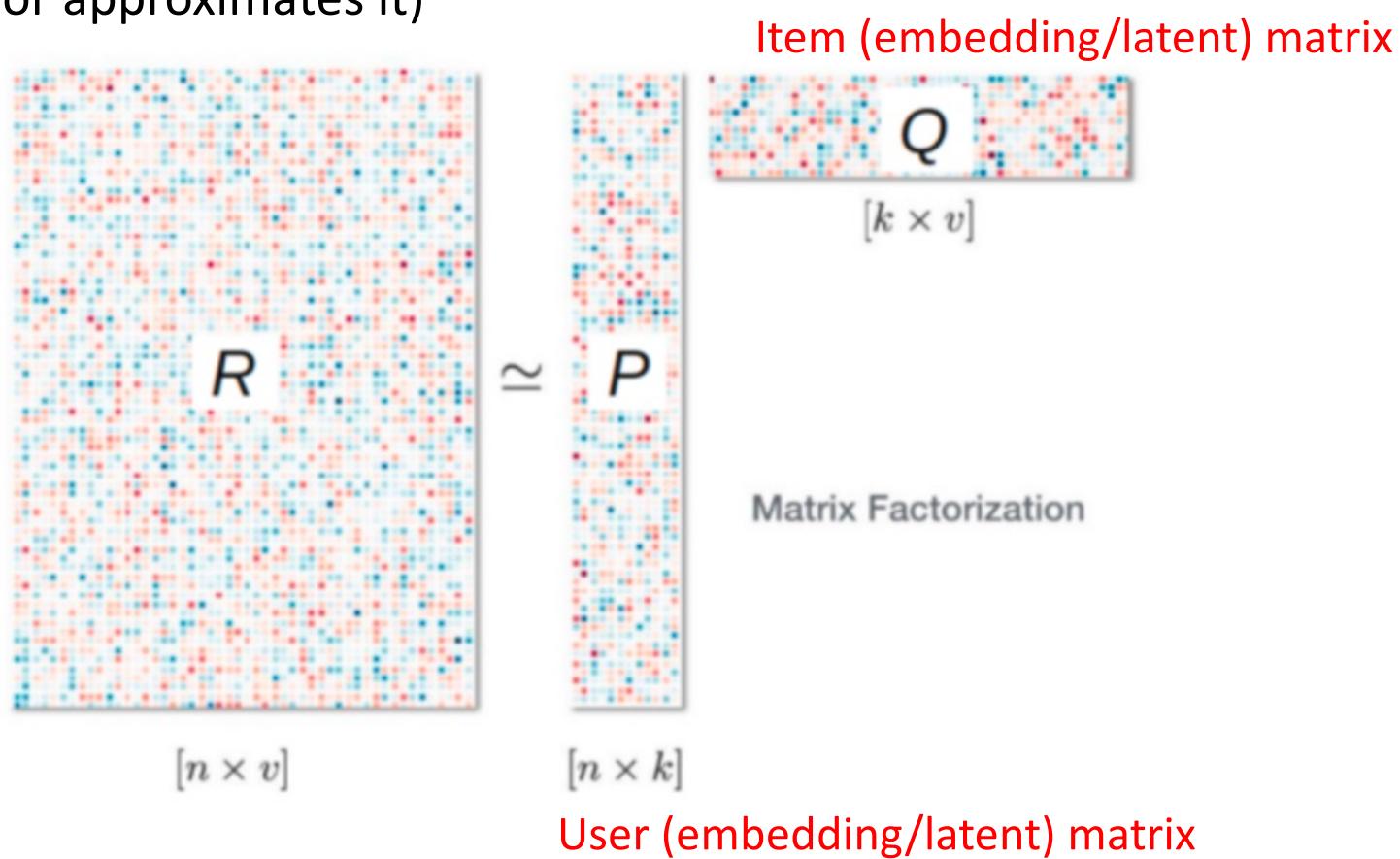
$j = 1, \dots, n_u, i = 1, \dots, n_m$:

$$x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) \theta_k^{(j)} + \lambda x_k^{(i)} \right)$$
$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right)$$

3. For a user with parameters θ and a movie with (learned) features x , predict a star rating of $\theta^T x$

2.2.2. Matrix factorization method

- ❖ Find two smaller matrices P and Q such that the ratings matrix equals to $P \cdot Q$ (or approximates it)



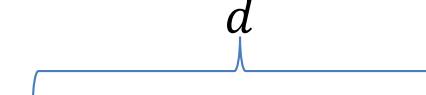
How to compute? SVD, VD++, Deep-Learning, ... (sorry, too many to cover)

Collaborative Filtering: Summary

- (+) Don't need domain knowledge because the embeddings are automatically learned
- (+) The model can help users discover new interests

- (-) Cold-start problem: new item or new user?
- (-) Sparsity problem: too few ratings to infer anything
- (-) Hard to include side features

3. Hybrid Recommendation: Rating + Content (OPTIONAL)



Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

$x^{(i)}$

- ❖ For each user j , learn a parameter vector $\theta^{(j)} \in R^d$. Predict that user j will rate movie i with $\theta^{(j)} \odot x^{(i)}$ stars

Regression Approach: Notations

$r(i, j) = 1$ if user j has rated movie i (0 otherwise)

$y^{(i,j)}$ = rating by user j on movie i (if defined)

$\theta^{(j)}$ = parameter vector for user j → User preference for each movie feature

$x^{(i)}$ = feature vector for movie i

For user j , movie i , predicted rating: $(\theta^{(j)})^T(x^{(i)})$

$m^{(j)}$ = no. of movies rated by user j

To learn $\theta^{(j)}$:

Regularized Mean Squared Error

Optimization objective:

To learn $\theta^{(j)}$ (parameter for user j):

$$\min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^n (\theta_k^{(j)})^2$$

To learn $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(n_u)}$:

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

Model Training

Optimization algorithm:

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

Gradient descent update:

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} \quad (\text{for } k = 0)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right) \quad (\text{for } k \neq 0)$$

4. Clustering-based Recommendation

❖ Limitations of rating-based recommendation:

- Hard to **scale** with large data
- Bad with **sparse** rating matrix
- Bad with **diversity** of users and items

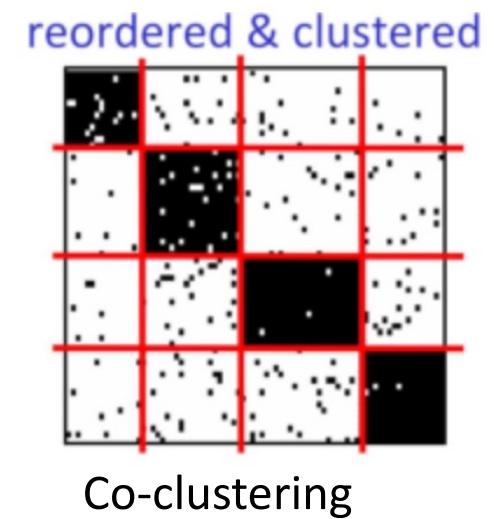
		Items						
		X		X				
			X	X				
Users		X			X	X		
			X		X	X		
				X		X	X	
				X			X	X
				X	X	X		

Clustering-based Recommendation

- ❖ Solution: **clustering** the data before-hand
 - Clustering based on ratings: k-means [*], etc.
 - **One-dimension clustering**: user clustering, item clustering
 - **Co-clustering**
- ❖ Then, perform rating-based recommendation on each cluster

		Items				
		Cluster 1		Cluster 2		
Users	Cluster 1	x	x			
		x	x	x	x	
Users	Cluster 2	x		x	x	x
		x	x	x	x	x
User clustering		Item clustering				

		Items				
		Cluster 1		Cluster 2		
Users	Cluster 1	x	x			
		x	x	x	x	
Users	Cluster 2	x		x	x	x
		x	x	x	x	x
User clustering		Item clustering				



[*] Al Mamunur Rashid, Shyong K. Lam, George Karypis, and John Riedl. "ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm." *Proceeding of WebKDD 2006* (2006).

Clustering-based Recommendation – How does it work?

- ❖ Any customer that shall be classified as a member of **CLUSTER** will receive recommendations based on preferences of the group:
 - Book 2 will be highly recommended to Customer F
 - Book 6 will also be recommended to some extent
- ❖ Pros:
 - Overcome the sparse data problem
 - Capture latent similarities between users and items
- ❖ Cons:
 - Recommendations (per cluster) maybe less relevant than collaborative filtering (per individual)

	Book1	Book2	Book3	Book4	Book5	Book6
CustomerA	X			X		
CustomerB		X	X		X	
CustomerC		X	X			
CustomerD		X				X
CustomerE	X				X	
CustomerF			X		X	

III. PERFORMANCE MEASURES

RecSys - Performance Measures

- ❖ Qualitative measures:
 - **User Satisfaction** (e.g. questionnaire)
- ❖ Quantitative measures:
 - If ground truth is not available:
 - Add-on sales
 - Click-through rates
 - Number of products purchased
 - If ground truth is available:
 - **Predictive accuracy:**
 - The ratio of predicted ratings being the true user ratings?
 - Rank accuracy

Predictive accuracy

- ❖ **Mean Absolute Error (*MAE*)**. The average absolute deviation between a predicted rating (p) and the user's true rating (r)

➤ $NMAE = MAE / (r_{max} - r_{min})$

$$MAE = \frac{\sum_{ij} |\hat{r}_{ij} - r_{ij}|}{n}$$

- ❖ **Root Mean Square Error (*RMSE*)**. Similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n} \sum_{i,j} (\hat{r}_{ij} - r_{ij})^2}$$

Evaluation Example

<i>Item</i>	<i>Predicted Rating</i>	<i>True Rating</i>
1	1	3
2	2	5
3	3	3
4	4	2
5	4	1

$$MAE = \frac{|1 - 3| + |2 - 5| + |3 - 3| + |4 - 2| + |4 - 1|}{5} = 2$$

$$NMAE = \frac{MAE}{5 - 1} = 0.5$$

$$\begin{aligned} RMSE &= \sqrt{\frac{(1 - 3)^2 + (2 - 5)^2 + (3 - 3)^2 + (4 - 2)^2 + (4 - 1)^2}{5}} \\ &= 2.28 \end{aligned}$$

Rank Accuracy

❖ Kendall's τ

- **Compares concordant** the items of the recommended ranking list **against the ground truth** ranking list
 - If the two orders are consistent, it is concordant
 - E.g., for top 4 items in ranking list, there are $4 \times 3 / 2 = 6$ pairs

$$\tau = \frac{c-d}{\binom{n}{2}}$$

- c is the number of concordants
- d is the number of discordants

Ranking Accuracy: Example

- ❖ Consider a set of four items $I = \{i_1, i_2, i_3, i_4\}$ for which the predicted and true rankings are as follows

	<i>Predicted Rank</i>	<i>True Rank</i>
i_1	1	1
i_2	2	4
i_3	3	2
i_4	4	3

Pair of items and their status

{**concordant**/**discordant**} are

(i_1, i_2) : concordant

(i_1, i_3) : concordant

(i_1, i_4) : concordant

(i_2, i_3) : discordant

(i_2, i_4) : discordant

(i_3, i_4) : concordant

$$\tau = \frac{4 - 2}{6} = 0.33$$

RecSys – Further Challenges

- ❖ Cold-Start Problem
 - Recommender systems use **historical data** or information provided by the user to recommend items, products, etc.
 - When user join sites, they still haven't bought any product, or they have no history.
 - It is hard to infer what they are going to like when they start on a site.
- ❖ Data Sparsity
 - When historical or prior information is **insufficient**.
 - Unlike the cold start problem, this is in the system as a whole and is not specific to an individual.
- ❖ Attacks:
 - **Push Attack:** pushing ratings up by making fake users
 - **Nuke attack:** DDoS attacks, stop the whole recommendation systems
- ❖ Explanation
 - Recommender systems often recommend items with **no explanation** on why these items are recommended

Cold-Start Solutions

- ❖ Use Popular Products
- ❖ Use Cookies ...

The screenshot shows the Amazon.com.au homepage with several features highlighted:

- Navigation Bar:** Includes the Amazon logo, "Hello Select your address", a search bar, and account links ("Hello, Sign in Account & Lists", "Returns & Orders", "Cart" with 0 items).
- Header Links:** "All", "Best Sellers" (highlighted with a red box), "Customer Service", "Prime", "Today's Deals" (highlighted with a red box), "Fashion", "Music", "Books", "Kindle Books", "New Releases", and a promotional banner for "Stream Movies and TV Shows with Prime".
- Hero Section:** A large banner with the text "*Alexa, play the news*" and images of Echo Show smart speakers.
- Deals Section:** A grid of four cards:
 - Prime deal: Sennheiser headphones:** An image of a woman wearing headphones.
 - Trending deals:** An image of a black adjustable weight bench.
 - Top Deal:** An image of an LG UHD Monitor displaying a car driving at night.
 - Sign in for your best experience:** A call-to-action button: "Sign in securely".
- Amazon Music Section:** A red banner with the text "amazon music" and "PLAY MUSIC, FREE".

Summary

- I. Why Personalization?
- II. Recommender Systems
 - 1. Content-based Recommendation
 - 2. Rating-based Recommendation (Collaborative Filtering)
 - 3. Hybrid Recommendation
 - 4. Clustering-based Recommendation
- III. Performance measures

References

- [1] https://en.wikipedia.org/wiki/Recommender_system
- [2] [https://en.wikipedia.org/wiki/Matrix_factorization_\(recommender_systems\)](https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems))