THE UNIVERSITY OF MELBOURNE

COMP20008 Elements of Data Processing

Semester 1 2020

Recommender Systems

Plan today

MELBOURNE

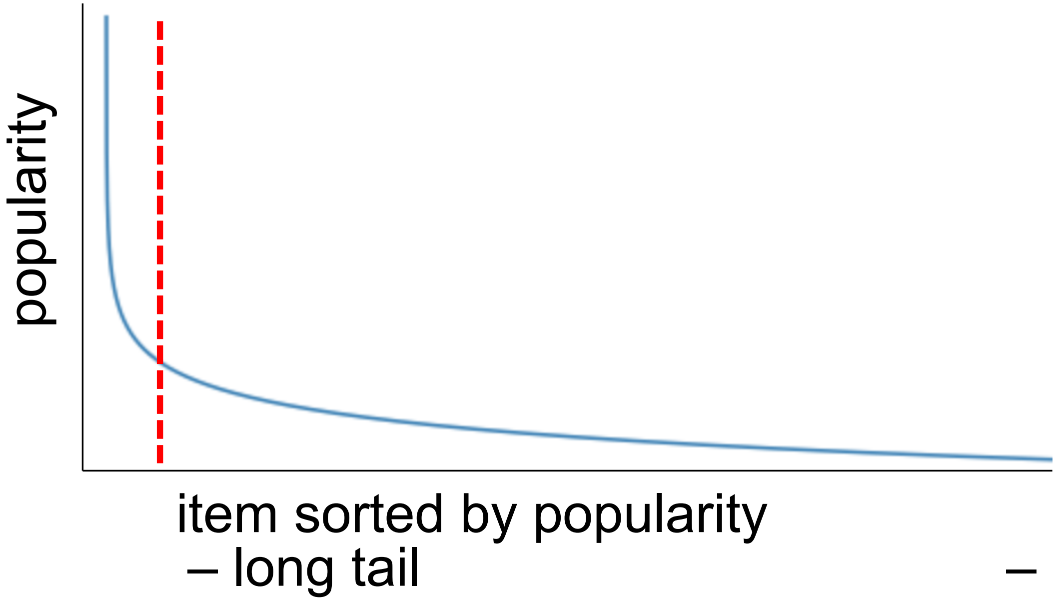
- Why recommender systems?
- Recommender systems and collaborative filtering
  - Popularity based
  - Collaborative filtering – memory based
    - Item-Item
    - User-User

- Friday's lecture
  - Content-based and model-based collaborative filtering
  - System evaluation

Why recommender systems?

MELBOURNE

- Scarcity to Abundance
- Internet changed shopping behaviours
- Online business is heavily dependent on recommender systems.



Why recommender systems?

MELBOURNE

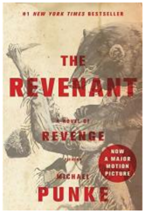
- The Long Tail by **Chris Anderson:** "*In 1988, a British mountain climber named Joe Simpson wrote a book called '**Touching the Void'**, a harrowing account of near death in the Peruvian Andes. It got good reviews but, only a modest success, it was soon forgotten. Then, a decade later, a strange thing happened. Jon Krakauer wrote '**Into Thin Air'**, another book about a mountain-climbing tragedy, which became a publishing sensation. Suddenly **Touching the Void** started to sell again*".

- "A lot of times, people don't know what they want until you show it to them" – Steve Jobs

Recommender systems – examples

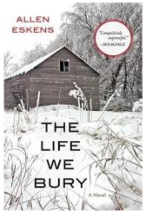
MELBOURNE

- LinkedIn
- Facebook
- Twitter
- Youtube
- Netfix
- Amazon

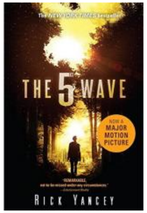


The Revenant: A Novel of Revenge
› Michael Punke
1,250
Paperback
$9.52

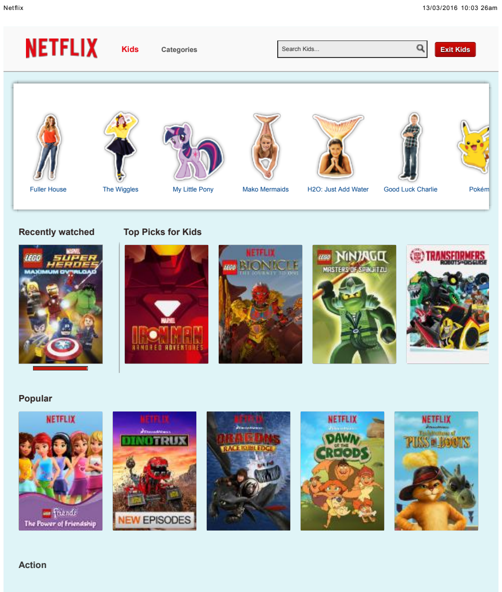
Ready Player One: A Novel
› Ernest Cline
9,210
Paperback
$8.37



The Life We Bury
› Allen Eskens
1,896
Paperback
$8.75



The 5th Wave: The First Book of the 5th Wave Series
› Rick Yancey
2,006
Paperback
$6.70



☆ ☆ ☆ ☆ ☆

Recommender systems

MELBOURNE

- "75% of what people watch is from some sort of recommendation" (Netflix)
- "If I have 3 million customers on the web, I should have 3 million stores on the web." (Amazon CEO)

Movie recommender systems

- finding best **matched** movies,
- reducing search times and frustration.

Users	Titanic	Batman	Inception	Superman	The Martian	Jurassic World
Harry	3	2	-	-	1	-
Ming	-	-	1	2	-	-
Peter	1	-	-	3	2	1

Recommender systems – How it works

MELBOURNE

- An online system where many users interact with many items.
- Each user has a profile
- User rate items
  - Explicitly: give a score
  - Implicitly: web usage mining: Time spent on viewing the item, etc.
- System does the rest, How?

Popularity based recommendation

MELBOURNE

- Show popular items.
- Which item is popular?
- Simple but not personalised.

Collaborative filtering

MELBOURNE

- Collaborative Filtering: Making predictions about a user's missing data according to the **collective** behaviour of many other users
  - Look at users' collective behavior (e.g. ratings)
  - Active user history
  - Combine!

- Item-based collaborative filtering (Item-Item)
- User-based collaborative filtering (User-User)

Collaborative filtering – A framework

MELBOURNE

I: n-items

U: m-users

R	i_1	i_2	$\cdots$	i_j	$\cdots$	i_n
u_1	3	4	2			1
u_2	2					5
$\vdots$	1		2		1	
u_i	3		4	r_{ij}**?**		
$\vdots$				2		3
u_m		5	3	4	3	

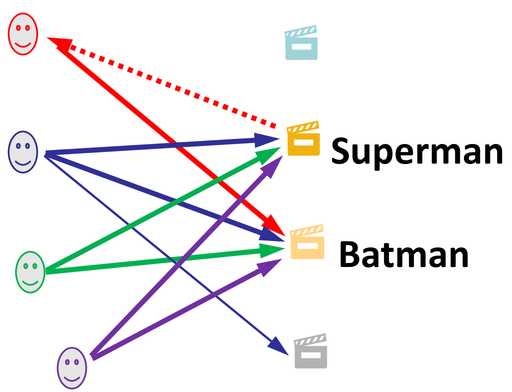
$f\colon U \times I \rightarrow R$

- Given
  - A set of m users $\boldsymbol{U}$ and a set of n Items $\boldsymbol{I}$
  - A $m \times n$ Interaction Matrix or Rating Matrix $\boldsymbol{R}$
- Find unknown ratings r_{ij}

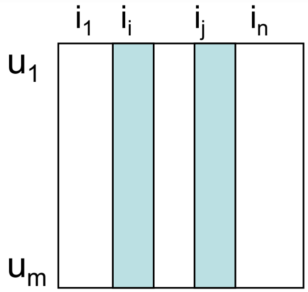
Item-based method: Intuition

People like things similar to other things they like

- Search for similarities among **items**
  - Many users like both *Batman* and *Superman*→ the two movies are similar.
  - Similarity is collective similarity in ratings by many users.
- Recommend items similar to those rated by the target user.
  - Superman and Batman are similar
  - If Peter liked *Batman* then recommend *Superman* to Peter.



Item based collaborative filtering



Three questions to address:

- How to measure item similarities?
- How to find similar items?
- How to combine ratings of these items?

Q1: Measure item-similarity

MELBOURNE

Example similarity between item i_i and item i_j:

- Euclidean distance with mean imputation
  - Imputation with their mean values
    - $mean(i_i) = \frac{3+3+2}{3} = 2.7$
    - $mean(i_j) = 3.75$

$$\begin{array}{cc} i_i & i_j \\ \end{array} \quad \begin{array}{cc} i_i & i_j \\ \end{array}$$

$$\begin{bmatrix} 3 & - \\ - & 3.5 \\ - & 3 \\ 3 & 3.5 \\ 2 & 4 \\ - & 4 \\ - & 4.5 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 3.75 \\ 2.7 & 3.5 \\ 2.7 & 3 \\ 3 & 3.5 \\ 2 & 4 \\ 2.7 & 4 \\ 2.7 & 4.5 \end{bmatrix}$$

  - Similarity score based on Euclidean distance

$$\boldsymbol{sim(i_i, i_j) = \frac{1}{1+d(i_j,i_i)}} \text{ where } \boldsymbol{d(i_i, i_j) = \sqrt{\sum_{k=1}^{m}(r_{ki} - r_{kj})^2}}$$

- $d(i_i, i_j) = 3.24$

$$= \sqrt{(3-3.75)^2+(2.7-3.5)^2+(2.7-3)^2+(3-3.5)^2+(2-4)^2+(2.7-4)^2+(2.7-4.5)^2}$$

- $sim(i_i, i_j) = \frac{1}{1+3.24} = 0.24$

Q2: How to find similar items?

MELBOURNE

- We have an answer to Q1, for item $x1$, we have the similarities between it and other items:

$sim(x1, x2)$	$sim(x1, x3)$	$sim(x1, x4)$	$sim(x1, x5)$	$sim(x1, x6)$
0.48	0.4	0.20	0.33	0.35

- The target user $\boldsymbol{a}$ has rated some items:

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$
a	?	4	-	5	3	3

- Choose a number $\boldsymbol{k}$, find k-most similar items to $x1$ for user a
- Let $k = 3$, which 3 items?
  - Items $x2$, $x6$, and $x5$
  - scores 0.48, 0.35, and 0.33.

x2, x3, x6, x5, x4

x3 ↓ can't use

since user hasn't rate this

Q3: How to combine ratings of similar items?

- Predict the rating of item $x1$ for user a
- From Q1 and Q2, we get:
  - For user a, the 3 (k = 3) most similar items to $x1$: $x2$, $x6$, $x5$

$sim(x1, x2)$	$sim(x1, x3)$	$Sim(x1, x4)$	$sim(x1, x5)$	$sim(x1, x6)$
0.48	0.4	0.20	**0.33**	**0.35**

  - The ratings of these 3 items by user a: 4, 3.5, 3

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$
a	?	**4**	-	5	**3**	**3**

- Rating = weighted average over the ratings of the 3 most similar items
  - $r_{a,x1} = \frac{0.48 \times 4 + 0.33 \times 3 + 0.35 \times 3}{0.48 + 0.33 + 0.35} = 3.41$

Item-based Collaborative Filtering – Algorithm

MELBOURNE

- Phase 1 – For each item j,
  - Compute similarities between j and other items. $\theta(n^2)$

    similarity: e.g. Euclidean distance with mean imputation.
  - Batch, Off-line → calculate similarity in advance
- Phase 2 – Predict rating of item *j* by user a based on the k-most similar items (among items rated by a)
  - Predicted rating = weighted average over the ratings of the k-most similar items.

$$r_{aj} = \frac{\sum_{i \in k-similar-items} sim(i,j) \times r_{ai}}{\sum_{i \in k-similar-items} sim(i,j)}$$

  - Online

Item-based filtering – Practice example

- Predict r_{aj} $(a = Tim; \quad j = Inception)$

Users	Titanic	Batman	Inception	Superman	The Martian	Jurassic World
Michelle	2.5		3	3.5	2.5	3
Tom	3	3.5		5	3	3.5
Lao	2.5	3		3.5		4
Chan		3.5	3	4	2.5	
Mary		4	2	3	2	3
Tim	3	4	$\boxed{r_{aj}?}$	5	3.5	3
John		4.5		4	1	

Phase – 1 offline: similarities between Inception and other movies

sim(Inception, Titanic)	sim(Inception, Batman)	sim(Inception, Superman)	sim(Inception, The Martian)	sim(Inception, Jurassic World)
0.48 (d=1.08)	0.24 (d=3.24)	0.20 (d=3.89)	0.33 (d=2.05)	0.34 (d=1.97)

choose k=3

Item-based filtering – Practice example cont.

- Phase – 2 online:
  - select 3-most similar items (k=3) w.r.t. (Tim, Inception)

sim(Inception, Titanic)	sim(Inception, Batman)	sim(Inception, Superman)	sim(Inception, The Martian)	sim(Inception, Jurassic World)
0.48	0.24	0.20	**0.33**	**0.34**

Users	Titanic	Batman	Inception	Superman	The Martian	Jurassic World
Michelle	2.5		3	3.5	2.5	3
Tom	3	3.5		5	3	3.5
Lao	2.5	3		3.5		4
Chan		3.5	3	4	2.5	
Mary		4	2	3	2	3
Tim	3	4	?	5	3.5	3
John		4.5		4	1	

  - weighted avg over the ratings of the 3-most similar items
  - $$r_{aj} = \frac{0.48\times3 + 0.33\times3.5 + 0.34\times3}{0.48 + 0.33 + 0.34} = 3.14$$

Item-based collaborative filtering Summary

MELBOURNE

- Item similarities computation is off-line → make online comparison efficient
- So, efficient at runtime.
- Developed by Amazon, suited for situations #users >> #items

- What do we do with new items?

1. randomly select people to rate
2. or change to other method

   eg. content-based

User-based collaborative filtering: Intuition

MELBOURNE

People like things liked by other people with similar taste

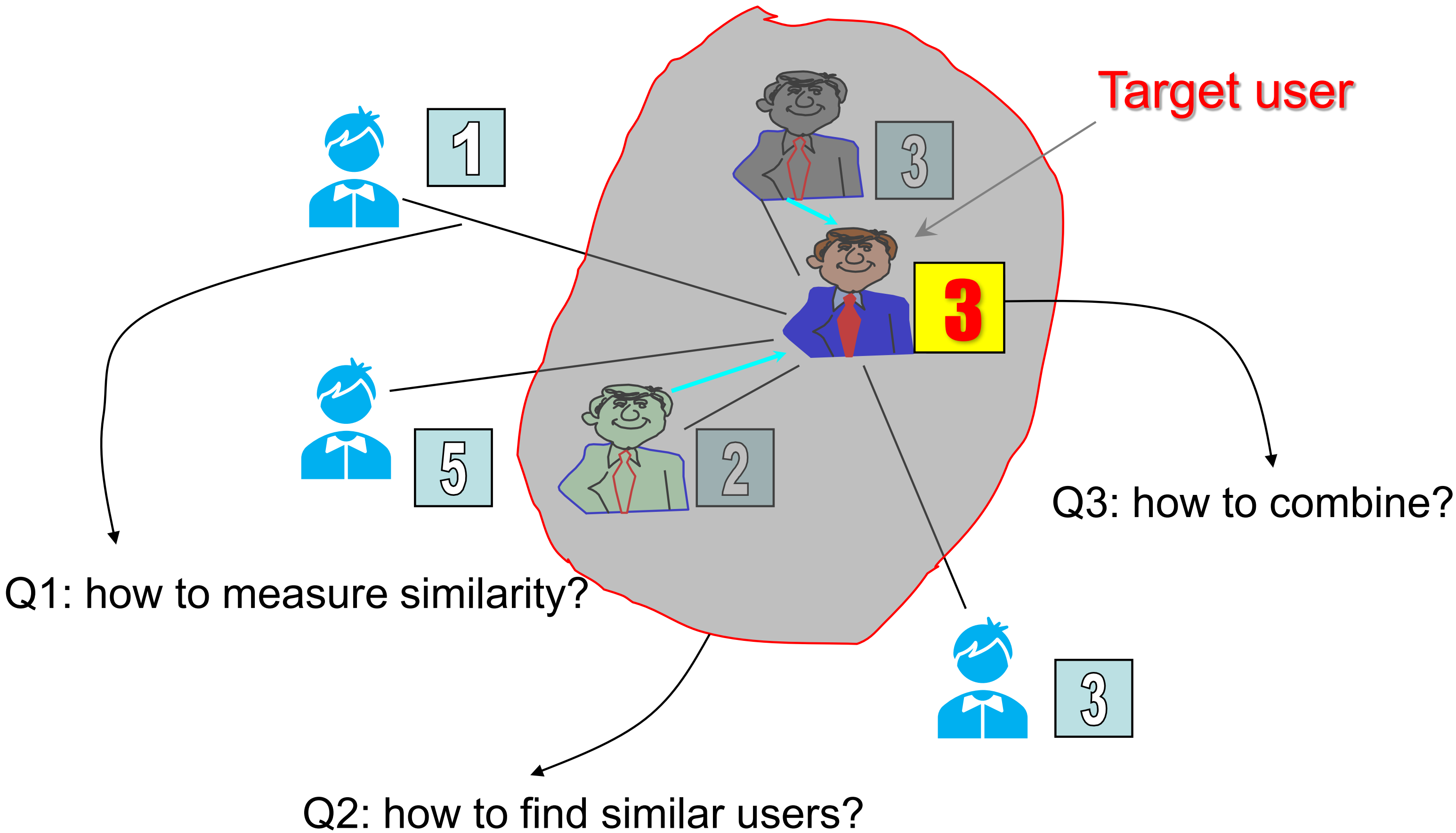
- Search for similarities among **users**
  - Two users Jane and Bob tend to like same movies; they have similar taste in movies.
- Recommend items like by users similar to the target user.
  - Jane and Bob have similar rating behaviours (taste),
  - If Jane liked *Batman* then

    recommend *Batman* to Bob.
- Mathematically similar to Item-based methods.

transpose the matrix

	use 1	use 2	user 3
item 1			
2			
3			
:			

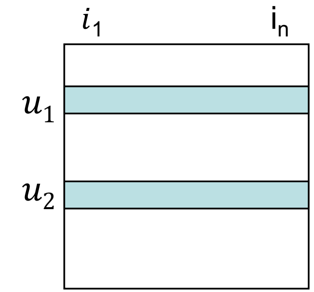
User-based method



Q1: How to measure similarity between users

- Euclidean distance with mean imputation

u_1	17	18.1 -	20	18	17	18.5
u_2	8	14.1 -	14.1 -	17	14	17.5



- $sim(u_1, u_2) = \frac{1}{1+d(u_1,u_2)} = 0.08$

$d(u_1, u_2) = 11.9 =$
$\sqrt{(17-8)^2 + (18.1-14.1)^2 + (20-14.1)^2 + (18-17)^2 + (17-14)^2 + (18.5-17.5)^2}$

  - Compute mean value for user1's missing values (18.1)
  - Compute mean value for user2's missing values (14.1)
  - Compute Euclidean distance between resulting rows
  - Convert the distance into a similarity (high similarity for low distance, low similarity for high distance)

User-based: Q2: How to find similar users? Q3: How to combine ratings?

- Selecting similar users and making prediction

- With respect to user a and item j:
  - Choose $\boldsymbol{k}$ most similar users **who have rated item $\boldsymbol{j}$**.
  - Prediction of rating is weighted average of the ratings of item j from the top-k similar users.

User-based method

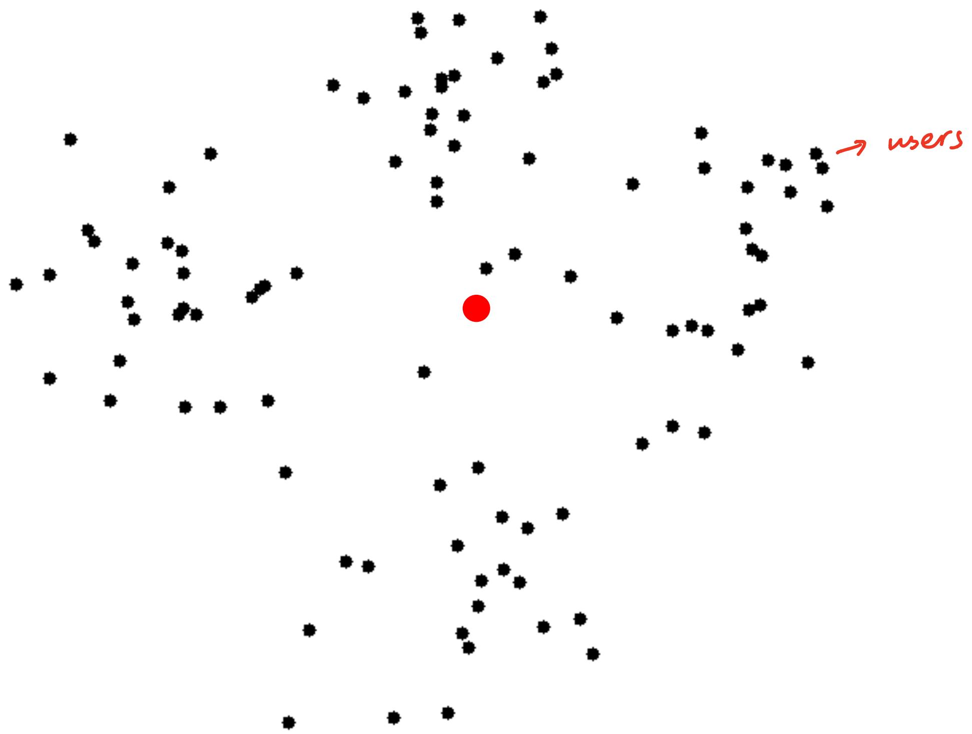
reason
① you can have more information to items than to users

- Mathematically similar to Item-based method.
- However:
  - Item-based performs better in many practical cases: movies, books, etc.
  - User preference is dynamic; relatively static for item based
    - High update frequency of offline-calculated information
  - Sparsity problem with user based method.

    if you have a new user, have no way to compare with other user

- No recommendation for new users
- Scalability issues
  - As the number of users increase, more costly to find similar users.
  - Offline clustering of users

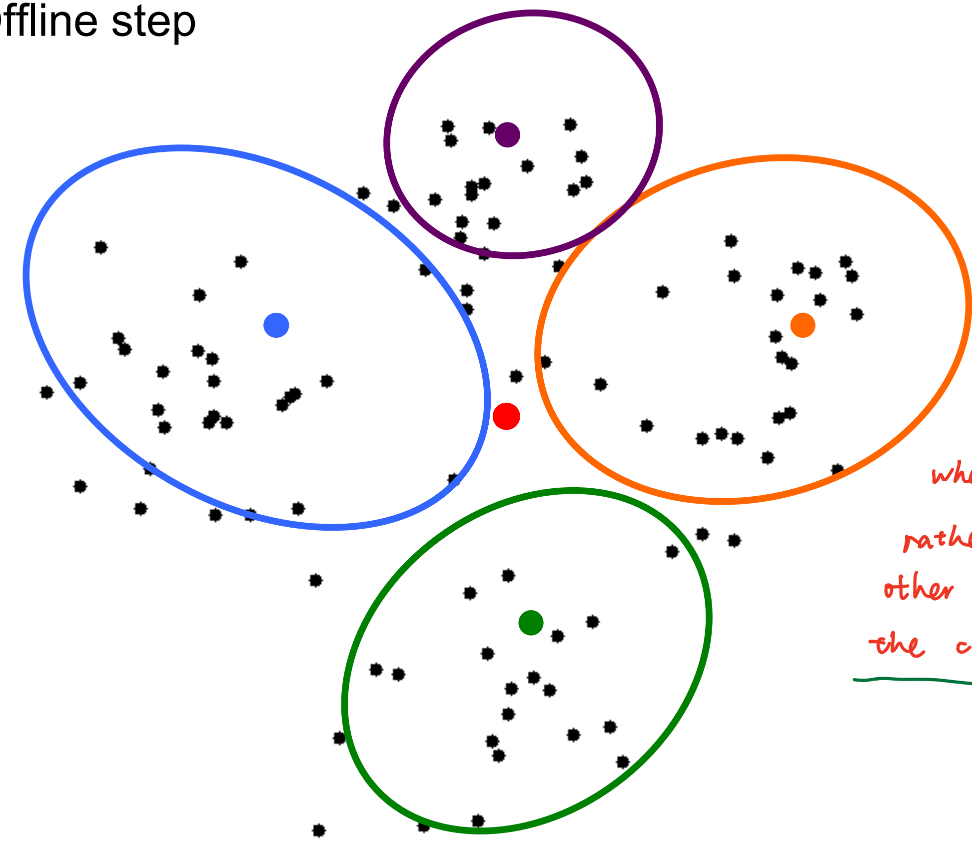
Scale-up search of k-similar users



Scale up search of k-similar users

MELBOURNE

- Offline step



when new user come rather than compare with other user, compare with the cluster center

Options for Q1: Similarity metrics

MELBOURNE

- Item-item: Considers the similar items
- User-user: Considers the similar users

- We looked at Euclidean distance based similarity.

- The other two popular similarity measures are
  - Cosine similarity and
  - Pearson Correlation (centered cosine similarity).

Cosine similarity

- Cosine similarity is a measure of similarity between two vectors X, Y.
  - a dot product between two vectors X, Y.
  - X, Y: 2 vectors of ratings by user x and user y
  - X, Y: 2 vectors of ratings of item x and item y

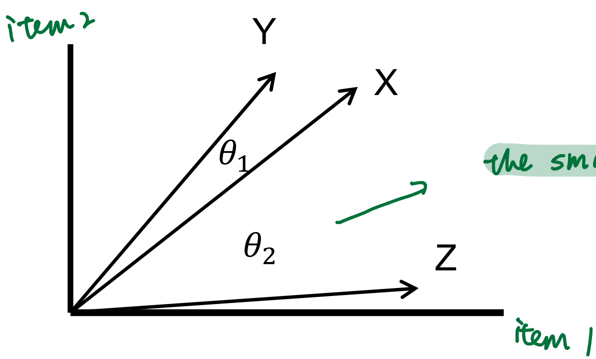
Impute 0 rather than mean

why [?]

- $$cos(X,Y) = \frac{X \cdot Y}{(\|X\| \cdot \|Y\|)} = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} \times \sqrt{\sum_{i=1}^{n} {y_i}^2}}$$

- $cos(X,Y) > cos(X,Z)$

X Y is more similar



the smaller the angle, more similar of two users

Centred cosine similarity

MELBOURNE

someone will rate between [5, 10]
someone will rate between [0, 7] → different people, different thought to a movie

- Cosine similarity
  - Missing values in vectors are imputed with the value 0

normalise by subtracting the mean

  - Issue: 0 has very different meanings in different vector context
    - Two users, one is tough and one is easy
    - Two items having higher and lower ratings.
    - Misleading results
- let X_{nrom} be normalised values of X and Y_{nrom} normalised Y
- $centred\_cos(X, Y) = cos(X_{norm}, Y_{norm}) =$

$$\frac{\sum_{i=1}^{n}(X_i - \bar{x})\times(Y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Pearson correlation

- Centred cosine similarity is Pearson correlation.

THE UNIVERSITY OF MELBOURNE

Summary

- We learnt:
  - Popularity based.
  - Item-based and user-based collaborative filtering (Gen 1)
    - Simple but reasonably powerful.
    - Achieves some level of personalisation.
  - Different measurements of similarities.
  - Some limitations with these approaches.
    - Cold start problem – new items/users
    - Scalability issues → particularly with user-based