CS-4053 Recommender System

Fall 2023

Lecture 2: Collaborative Filtering

Course Instructor: Syed Zain Ul Hassan

National University of Computer and Emerging Sciences, Karachi

Email: zain.hassan@nu.edu.pk

Terminologies

- List of ***m*** users and a list of ***n*** items
- Each user has a list of items with associated opinion (rating). This opinion can be:
  - **Explicit** *e.g. a 3-star rating for an app on Google Play Store*
  - **Implicit** *e.g. purchasing (or not) purchasing a certain product*

- An **Active User** for whom the CF prediction task is performed
- A **Metric** for measuring similarity between users
- A **Method** for selecting subset consisting of closest neighbors

Collaborative Filtering (CF)

- **Basic idea**
  - *Use "similarities" to recommend items to the active user*
- **Background**
  - *(Used to be) The most prominent approach for recommendations*
  - *well-understood, various algorithms and variations exist*
  - *applicable in various domains (movies, e-commerce, songs, ...)*
- **Approach**
  - **User-based CF:** *Find users most similar to me and recommend to me what they liked*
  - **Item-based CF:** *Recommend to me an item that is similar to the ones I frequently like*

Collaborative Filtering (CF)

- **Input**
  - A matrix of user-item ratings

2		2	4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

- **Output**
  - A (numerical) prediction indicating **to what degree** the **active user** will **like** or dislike an **item**
  - A list of top-N recommended items

User-based Collaborative Filtering: Basic Steps

- Given an **active user X** and an item *i* not yet seen by **X**:
  - Find a set of users (peers/"nearest neighbors") who liked the same items as **X** in the past and who have rated item *i*
  - Use, e.g. the average of their ratings to predict if **X** will like item *i*
  - Do this for all items **X** has not seen and recommend the best-rated
- The idea is to find *k* users who are the nearest neighbors
- Also known as **user-based nearest neighbor collaborative filtering**

User-based Collaborative Filtering: Example

- Consider the following matrix of users and their ratings for items (**User 1** is the **active user** in this example)

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

User-based Collaborative Filtering: Example

- Consider the following matrix of users and their ratings for items (**User 1** is the **active user** in this example)

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

- Predict the rating of **User 1** for **Item 5** (assuming other users provided explicit ratings for items)

User-based Collaborative Filtering: Example

Some issues:

- *How do we measure similarity?*
- *How many neighbors should we consider?*
- *How do we generate a prediction from the neighbors' ratings?*

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

Measuring User Similarity

- When we compute similarity, we are going to calculate it as a measure of *"anti-distance“*

- Generally speaking, similarity is the inverse of distance:

$$\boldsymbol{Similarity = 1 - Distance}$$

- **Some similarity measures:**
  - *Euclidean*
  - *Jaccard*
  - *Cosine*
  - *Adjusted cosine*
  - *Raw cosine*
  - *Pearson correlation*

  and more…

Measuring User Similarity

- When we compute similarity, we are going to calculate it as a measure of *"anti-distance"*

- **Some similarity measures:**
  - **Euclidean distance** *(the simplest one)*

    <u>Example</u>:

    Consider two vectors **v1 = (3, 10)** and **v2 = (7, 13)**

    The Euclidean or straight-line distance between them is given by:

$$\mathbf{d} = \sqrt{(\boldsymbol{x_2} - \boldsymbol{x_1})^2 + (\boldsymbol{y_2} - \boldsymbol{y_1})^2}$$

$$\mathbf{d} = \sqrt{(\mathbf{7} - \mathbf{3})^2 + (\mathbf{13} - \mathbf{10})^2} = \mathbf{5}$$

Measuring User Similarity: **Example**

- Find **Euclidean distance** between **User 1** and **all other users**

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

Measuring User Similarity: Example

- Find **Euclidean distance** between **User 1** and **all other users**
- We find ***k*** nearest neighbors of **User 1**

	Item 1	Item 2	Item 3	Item 4	Item 5	Euclidean Distance with User 1	Similarity with User 1
User 1	5	3	4	4	?	0	1
User 2	3	1	2	3	3	≈ 3.60	-2.6
User 3	4	3	4	3	5	≈ 1.41	-0.41
User 4	3	3	1	5	4	≈ 3.74	-2.74
User 5	1	5	5	2	1	≈ 5	-4

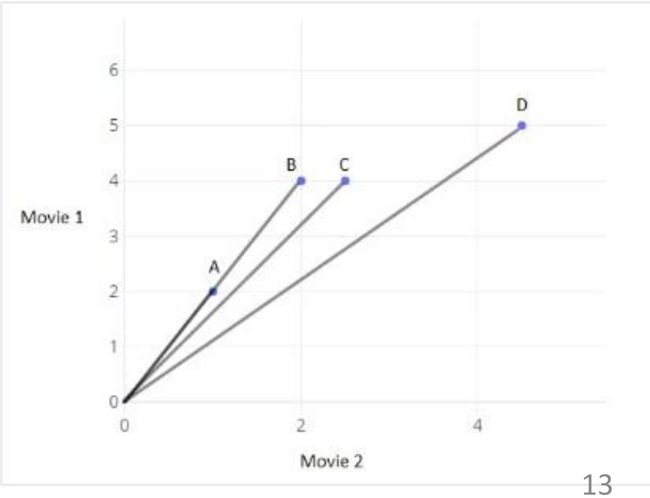
- For ***k* = 2**, the nearest neighbors of **User 1** are **User 2** and **User 3**

Measuring User Similarity

- At times, Euclidean or Manhattan distance cannot correctly detect patterns between our data points

- **Cosine distance (or similarity)** is another measure that can be used

$$CosineSim = Cos(\theta)$$



Measuring User Similarity

- To measure Cosine similarity between users **A** and **B**:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Measuring User Similarity: Example

- Find **Cosine similarity** between **User 1** and **all other users**
- We then find ***k*** nearest neighbors of **User 1** the same way we did for Euclidean distance (similarity)

	Item 1	Item 2	Item 3	Item 4	Item 5	Cosine Distance with User 1	Similarity with User 1
User 1	5	3	4	4	?	0	1
User 2	3	1	2	3	3		
User 3	4	3	4	3	5		
User 4	3	3	1	5	4		
User 5	1	5	5	2	1		

Measuring User Similarity: Example

- The **Cosine similarity** between **User 1** and **User 2** can be calculated as:

$$\boldsymbol{Cosine(U1, U2)} = \frac{(5*3+3*1+4*2+4*3)}{\sqrt{5^2+3^2+4^2+4^2}\,.\sqrt{3^2+1^2+2^2+3^2}} = \mathbf{0.97}$$

	Item 1	Item 2	Item 3	Item 4	Item 5	Cosine Similarity with User 1
User 1	5	3	4	4	?	1
User 2	3	1	2	3	3	≈ 0.97
User 3	4	3	4	3	5	
User 4	3	3	1	5	4	
User 5	1	5	5	2	1	

Measuring User Similarity

- We can also measure similarity between users with **Pearson Correlation Coefficient *(r)***

- It measures both magnitude and orientation between data points

- The strength and relationship is given by a number between **-1** and **1**
  - *-1 means strong negative correlation*
  - *0 means no correlation*
  - *1 means strong positive correlation*

Measuring User Similarity

- To measure Pearson Correlation Coefficient *(r)* between **x** and **y**:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Measuring User Similarity: Example

- The **Pearson Correlation** between **User 1** and **User 2** is calculated as:

$$r(U1, U2) = \frac{(5-4)*(3-2.4)+(3-4)*(1-2.4)+(4-4)*(2-2.4)+(4-4)*(3-2.4)}{\sqrt{1^2+(-1)^2+0^2+0^2}\,.\,\sqrt{0.6^2+(-1.4)^2+(-0.4)^2+0.6^2}} = 0.85$$

	Item 1	Item 2	Item 3	Item 4	Item 5	Mean	Pearson Correlation Similarity with User 1
User 1	5	3	4	4	?	4	1
User 2	3	1	2	3	3	2.4	≈ 0.85
User 3	4	3	4	3	5	3.8	
User 4	3	3	1	5	4	3.2	
User 5	1	5	5	2	1	2.8	

Measuring User Similarity: Example

- In a similar way, we find Pearson Correlation similarity between **User 1** and **all other users**

- For ***k* = 2**, the nearest neighbors of **User 1** are **User 2** and **User 4**

	Item 1	Item 2	Item 3	Item 4	Item 5	Mean	Pearson Correlation Similarity with User 1
User 1	5	3	4	4	?	4	1
User 2	3	1	2	3	3	2.4	≈ 0.85
User 3	4	3	4	3	5	3.8	≈ 0
User 4	3	3	1	5	4	3.2	≈ 0.70
User 5	1	5	5	2	1	2.8	≈ -0.76

Now what?

- Now that we have found the users most similar to the active users we can use them to predict our rating for active user

- There can be various prediction functions *e.g.*

$$R_U = (\sum_{u=1}^{n} R_u)/n$$

User-based Collaborative Filtering: Prediction

- The final predicted rating of **Item 5** for **User 1** is given by:

$$R_{15} = \frac{(3*0.85)+(4*0.70)}{|0.85|+|0.70|} = 3.45$$

$$R_{15} \approx 3$$

User-based Collaborative Filtering: Example

- Based on Pearson Correlation Coefficient and k = 2 nearest neighbors, the predicted rating of **User 1** is 3.45 ≈ **3**

	Item 1	Item 2	Item 3	Item 4	Item 5	Mean	Pearson Correlation Similarity with User 1
User 1	5	3	4	4	3	4	1
User 2	3	1	2	3	3	2.4	≈ 0.85
User 3	4	3	4	3	5	3.8	≈ 0
User 4	3	3	1	5	4	3.2	≈ 0.70
User 5	1	5	5	2	1	2.8	≈ -0.76

Pearson Correlation Coefficient: Issues

- Underlying assumption is that users dislike what they rated below average
- This is not true in practice (we rate only what we liked or highly disliked)
- The correlation *flattens* in case of uniformly distributed ratings

Deviation from average rating on shared items

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2} + \varepsilon}$$

!!! Will be zero in case of uniform rating !!!

User-based Collaborative Filtering: Prediction

- Does the prediction function used in the previous example always provides correct relative ordering of the predicted ratings?
  - *Maybe not*

- We need a prediction function that is *mean-centered*

- Let’s understand the issue using another example

User-based CF: Another Example

- The given table contains *user-user* similarity computation for **5** users and **6** items

- Let us consider **User 3** as **active user** for whom we have to predict ratings for unseen **Item 1** and **Item 6** and recommend the top-rated item from these two

User-based CF: **Example**

- The given table contains *user-user* similarity computation for **5** users and **6** items

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Mean	Pearson Correlation Similarity with User 3
User 1	7	6	7	4	5	4	5.5	0.894
User 2	6	7	?	4	3	4	4.8	0.939
User 3	?	3	3	1	1	?	2	1
User 4	1	2	2	3	3	4	2.5	-1
User 5	1	?	1	2	3	3	2	-0.817

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** is given by:

$$R_{31} = \frac{(7*0.894)+(6*0.939)}{|0.894| + |0.939|} \approx 6.49$$

$$R_{36} = \frac{(4*0.894)+(4*0.939)}{|0.894| + |0.939|} = 4$$

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** is given by:

$$R_{31} \approx 6.49 \text{ (we can round off } R_{31} \text{ to 6)}$$

$$R_{36} = 4$$

- We will recommend **Item 1** to the **User 3**

- Also observe that based on these ratings, we can conclude that **User 3** like **Item 1** and **Item 6** more than they like any other item
  - *Is that assumption really correct?*

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** is given by:

$$R_{31} \approx 6.49 \quad \textit{(we can round off } R_{31} \textit{ to 6)}$$

$$R_{36} = 4$$

- We will recommend **Item 1** to the **User 3**

- Also observe that based on these ratings, we can conclude that **User 3** like **Item 1** and **Item 6** more than they like any other item
  - *This appears to be an incorrect assumption based on the correlation*

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** is given by:

$$R_{31} \approx 6.49$$ *(we can round off R_{31} to 6)*

$$R_{36} = 4$$

- We will recommend **Item 1** to the **User 3**

- Also observe that based on these ratings, we can conclude that **User 3** like **Item 1** and **Item 6** more than they like any other item
  - *This appears to be an incorrect assumption based on the correlation*
  - ***Solution**: Using a mean-centered prediction function to remove bias*

User-based Collaborative Filtering: Prediction

- Let's use a different prediction function that is **mean-centered** in order to remove bias:

$$\boldsymbol{R_U} = \overline{\boldsymbol{r_a}} + \frac{\sum_{\boldsymbol{b} \in \boldsymbol{N}} \boldsymbol{sim}(\boldsymbol{a}, \boldsymbol{b}) * (\boldsymbol{r_{b,p}} - \overline{\boldsymbol{r_b}})}{\sum_{\boldsymbol{b} \in \boldsymbol{N}} \boldsymbol{sim}(\boldsymbol{a}, \boldsymbol{b})}$$

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** using mean-centered prediction function are:

$$R_{31} = 2 + \frac{(1.5*0.894)+(1.2*0.939)}{|0.894|+|0.939|} \approx 3.35$$

$$R_{36} = 2 + \frac{(-1.5*0.894)+(-0.8*0.939)}{|0.894|+|0.939|} \approx 0.86$$

User-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **Item 6** for **User 3** using mean-centered prediction function are:

$$R_{31} \approx 3.35$$ *(we can round off R_{31} to 3)*

$$R_{36} \approx 0.86$$ *(we can round off R_{36} to 1)*

- **Observation**
  - ***Item 3*** *still appears to be the most liked item by* ***User 3***
  - *But* ***Item 6*** *is now clearly the least liked item by* ***User 3***

Item-based Collaborative Filtering

- Basic idea is the same as user-based neighborhood based prediction *except* that we use the similarity between items (and not users) to predict the rating

- Item-based Collaborative Filtering is relatively more stable

Item-based Collaborative Filtering

- Basic idea is the same as user-based neighborhood based prediction *except* that we use the similarity between items (and not users) to predict the rating

- Item-based Collaborative Filtering is relatively more stable

  "Things don't change as much as people do."
  — Made-up quote

Item-based Collaborative Filtering: Example

- For User 3, we need to predict ratings for Item 1 and Item 6

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Mean
User 1	7	6	7	4	5	4	5.5
User 2	6	7	?	4	3	4	4.8
User 3	?	3	3	1	1	?	2
User 4	1	2	2	3	3	4	2.5
User 5	1	?	1	2	3	3	2

Item-based Collaborative Filtering: Example

- Although we can use any similarity measures discussed previously but we are going to use Adjusted Cosine similarity for this example
  - *It is Cosine similarity that is mean-adjusted*

$$\boldsymbol{sim(\vec{a},\vec{b}) = \frac{\sum_{u\in U}(r_{u,a}-\overline{r_u})(r_{u,b}-\overline{r_u})}{\sqrt{\sum_{u\in U}(r_{u,a}-\overline{r_u})^2}\sqrt{\sum_{u\in U}(r_{u,b}-\overline{r_u})^2}}}$$

Item-based Collaborative Filtering: Example

- For **User 3**, we need to predict ratings for **Item 1** and **Item 6**:

Adj Cosine(I1, I3) = $\frac{(1.5*1.5)+(-1.5*-0.5)+(-1*-1)}{\sqrt{1.5^2+(-1.5)^2+(-1)^2}\,.\,\sqrt{1.5^2+(-0.5)^2+(-1)^2}}$ **= 0.912**

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	1.5	0.5	1.5	-1.5	0.5	-1.5
User 2	1.2	2.2	?	-0.8	-1.8	-0.8
User 3	?	1	1	-1	-1	?
User 4	-1.5	-0.5	-0.5	0.5	0.5	1.5
User 5	-1	?	-1	0	1	1

Item-based Collaborative Filtering: Example

- For **User 3**, we need to predict ratings for **Item 1** and **Item 6**:

Adj Cosine(I1, I3) = $\frac{(\mathbf{1.5*1.5})+(\mathbf{-1.5*-0.5})+(\mathbf{-1*-1})}{\sqrt{\mathbf{1.5}^2+(\mathbf{-1.5})^2+(\mathbf{-1})^2}\,.\,\sqrt{\mathbf{1.5}^2+(\mathbf{-0.5})^2+(\mathbf{-1})^2}}$ **= 0.912**

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	1.5	0.5	1.5	-1.5	0.5	-1.5
User 2	1.2	2.2	?	-0.8	-1.8	-0.8
User 3	?	1	1	-1	-1	?
User 4	-1.5	-0.5	-0.5	0.5	0.5	1.5
User 5	-1	?	-1	0	1	1

- In the same manner, calculate similarity of I_1 with **all other items**

Item-based Collaborative Filtering: Prediction

- The final predicted ratings of **Item 1** for **User 3** is given by:

$$\mathbf{R_{31}} = \frac{(3*0.735)+(3*0.912)}{|0.735|+|0.912|} = \mathbf{3}$$

Note: We assume 0.735 to be the similarity score between Item 1 and the second most similar item (Item 2). You can calculate it yourself and verify

Item-based Collaborative Filtering: Exercise

- **Task 1:** What will be the predicted rating for **Item 6** of **User 3**?

- **Task 2:** Predict all the missing ratings and find the top (unseen) item that can be recommended to each user

Collaborative Filtering vs Classification

- **Collaborative Filtering** vs **Classification**
  - *Unlike classification, there is no distinction between dependent and independent variables in collaborative filtering*

- **Collaborative Filtering** is similar to **missing value analysis** but with a much larger matrix

Improving CF: Significance Weighting

- The reliability of any similarity function ***sim(u, v)*** between two users **u** and **v** is often affected by the number of common ratings between **u** and **v** *i.e.* **($I_u \cap I_v$)**

- When the two users have only a small number of ratings in common, the similarity function ***sim(u, v)*** should include a discount factor to **de-emphasize** the importance of that particular user pair

- This method is referred to as **Significance Weighting**

- The discount factor kicks in when the number of common ratings between the two users is less than a particular threshold β

Improving CF: Significance Weighting

- The discount similarity **DiscountSim(u, v)** is given by:

$$DiscountSim(u, v) = Sim(u, v).\frac{min\{(I_u \cap Iv), \beta\}}{\beta}$$

*where **$I_u \cap I_v$** is the number of common ratings between users **u** and **v**,*

***Sim(u,v)** is the original similarity score (using any measure) and **β** is our threshold value*

User-based CF: Pros and Cons

Pros

- Provides more diverse recommendations
- Is a better choice if no. of users is much smaller than no. of items (which is common in practice)

Cons

- It is generally not stable as user preferences change rather quickly
- Cannot provide in-depth analysis on individual user

Item-based CF: Pros and Cons

Pros

- Provides more accurate recommendations in general
- Is a better choice unless the no. of items are much larger than the no. of users
- Is more stable
- Can provide better in-depth analysis on individual users

Cons

- Is prone to shilling attacks
  - *A malicious user running campaign to degrade some particular item on purpose*
- Provides much less diversity than user-based collaborative filtering

Collaborative Filtering: Issues

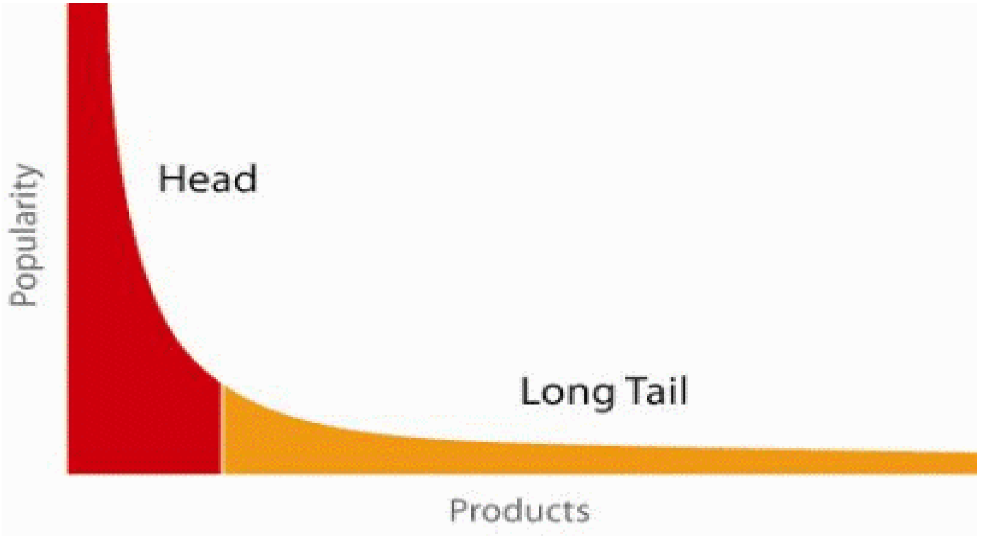
- **Serendipity**
  - *Expand the user's taste into neighboring areas*
    __Basic Idea:__ At times, it's good to recommend something different to the user

Collaborative Filtering: Issues

- **Cold Start**
  - *Using collaborative filtering without any initial data is very difficult*
    Basic Idea: *Ratings are not available for a newly launched website/store*

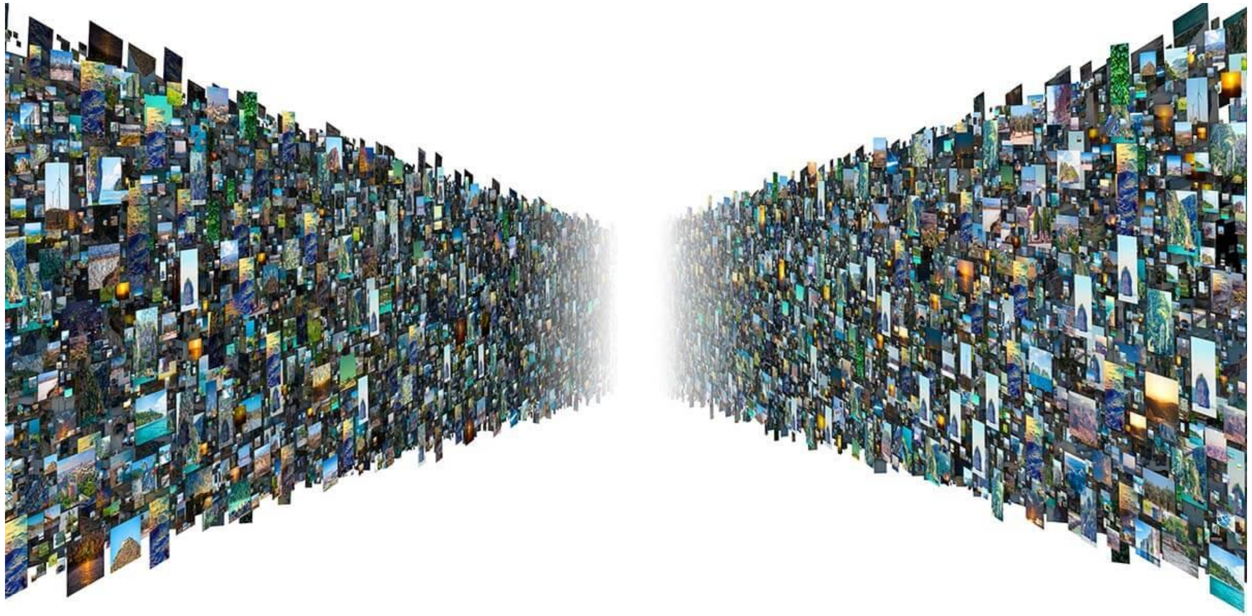
Collaborative Filtering: Issues

- **Long Tail**
  - *In practice, very few (relatively) popular items would be the ones rated by the users*
    Basic Idea: *A large number of items will be unrated hence cannot be recommended easily*
    Sparsity: *Long Tail can often lead to sparsity i.e. not having enough data to make prediction*



Collaborative Filtering: Issues

- **Scaling**
  - *Collaborative filtering requires a lot of computational operations*
    Basic Idea: *For Amazon, the number of items and users can be in millions*
    Possible Solution*: Use offline training i.e. don't throw away pre-computed similarities*



Memory-based vs Model-based

- Recommender Systems can either be **Memory-based** or **Model-based**

- **Memory-based** systems use entire data every time a rating is to be predicted
  - *User-based Collaborative Filtering*

- **Model-based** systems use the data once to create a model and can make a new prediction without using the entire data again
  - *Item-based Collaborative Filtering*
  - *Content-based Recommender System*