

Teaching Session | RecSys

Tamojit Maiti

Masters in Applied Statistics and Operations Research, ISI Kolkata

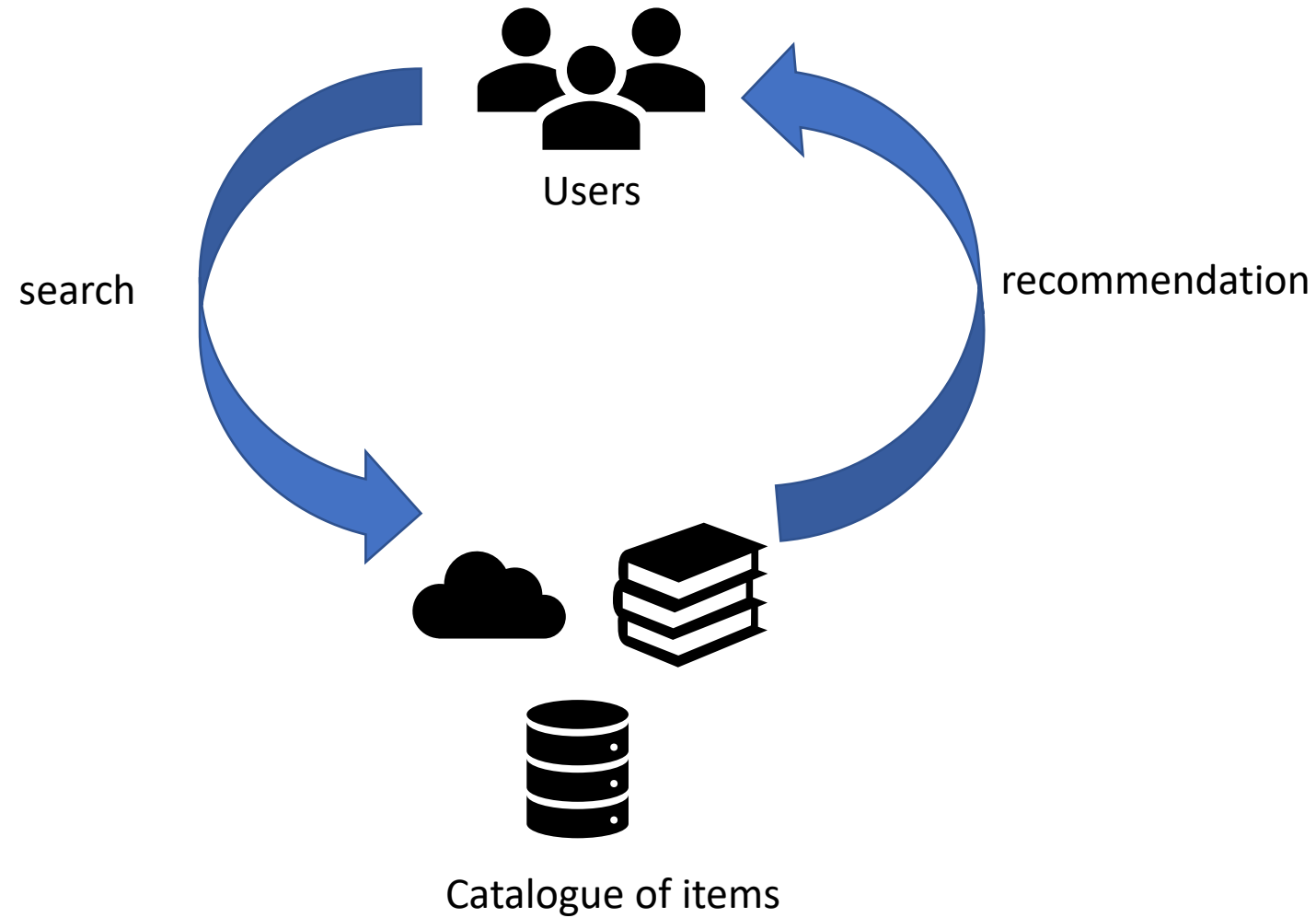
Data Scientist at Sixt R&D, previously at Rapido & AB InBev

Agenda

- Introduction
 - Why
 - Formal Math
- Collaborative Filtering
 - Theory
 - User-User
 - Item-Item
 - Drawbacks
- Latent Factor Model
 - Theory
 - Drawbacks
- Content Based Filtering
 - Theory
 - Caveats

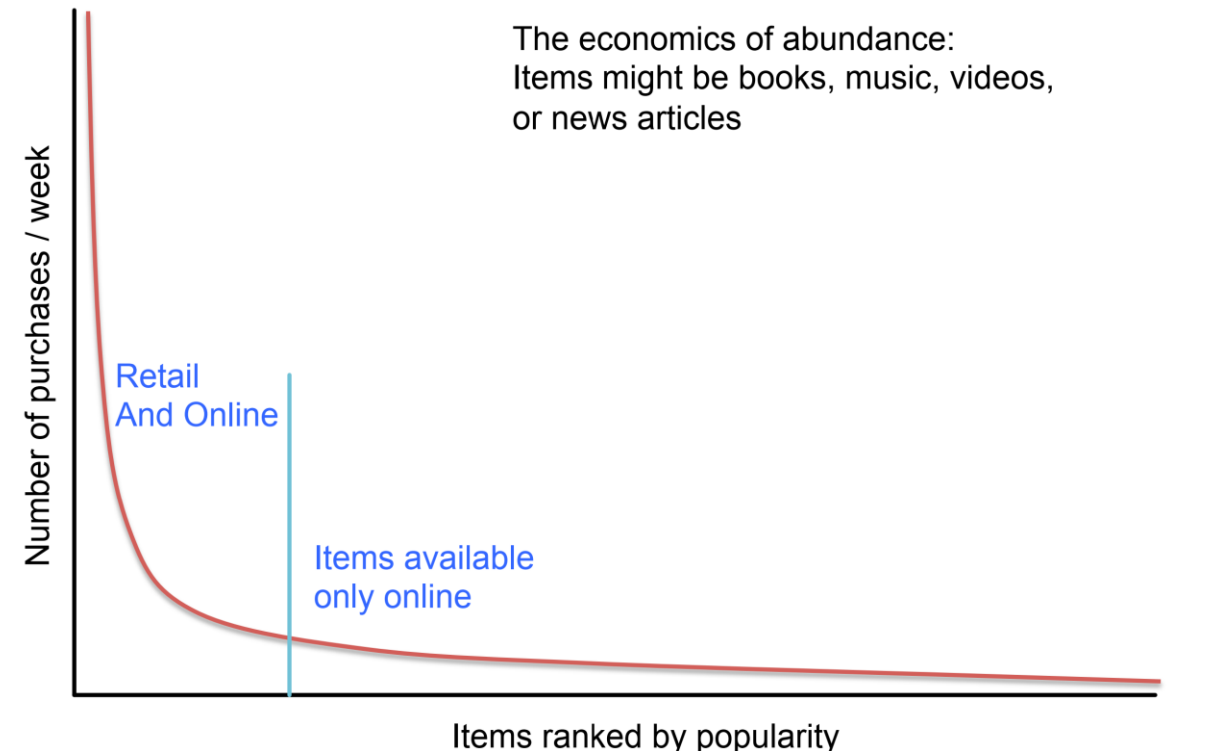
Introduction to RecSys

Introduction to RecSys | What



Introduction to RecSys | Why

- Brick and mortar stores did not need large scale recommendation systems
 - Lesser number of unique products
 - Lesser number of unique customer preferences
 - High cost of storage of inventory
- The internet brought with itself twin abilities
 - Low cost of dissemination of information of items
 - Larger access to number of unique customer preferences



Introduction to RecSys | Formal Math

C : Set of all customers

S : Set of all items

R : Set of ratings

u : Utility function (often matrix) that maps each combination of customer-item to the set of ratings R

That is

$$u : C \times S \rightarrow R$$

Introduction to RecSys | Formal Math

Utility Matrix

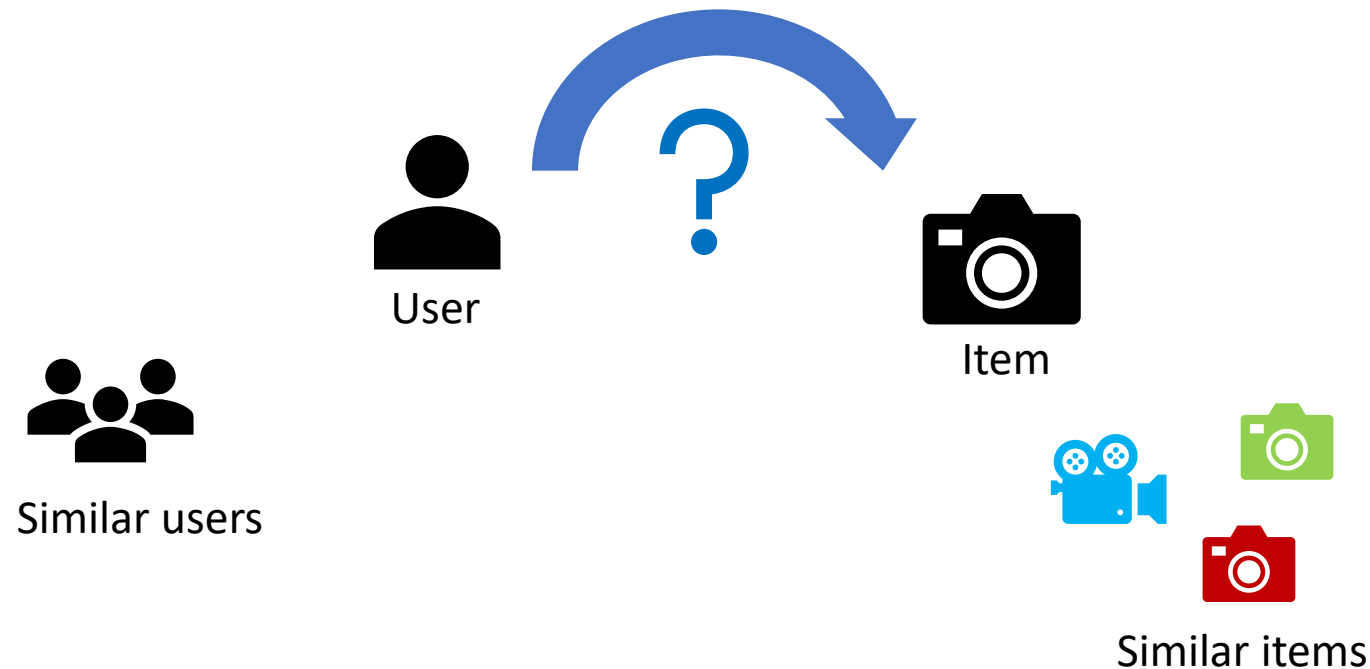
	M1	M2	M3	M4	M5
Ali	3	1	1	3	1
Brad	1	2	4	1	3
Carol	3	1	1	3	1
Dany	4	3	5	4	4

- Usually is sparse, contains many missing values
- Rows are not completely independent, groups of users have similar preferences
- Columns are not completely independent, groups of items share similar attributes
- Missing values can have important implication

Collaborative Filtering

Collaborative Filtering | Theory

- A user's affinity for a particular item depends on
 - Aggregate affinity towards that item by a similar group of users (user – user)
 - Aggregate affinity towards a similar set of items by all users (item - item)



Collaborative Filtering | Measuring Similarity

- We can define similarity between items by using various distance measures
 - Jaccard Index
 - Takes into account presence or absence of ratings
 - Does NOT take magnitude of ratings into account
 - Cosine Similarity
 - Takes into account magnitude of ratings into account
 - Personal rating biases can cause misleading results
 - Mean Adjusted Cosine Similarity
 - Handles biases in ratings, for example consistent high or low raters
- The next step involves selecting top N similar entities, either users or items

Collaborative Filtering | User - User

- Given a utility matrix $U = \{u_{ij}\}$, we want to find u_{pq} , that is, the rating of product q as rated by user p
- We find N users most similar to user p by using the vector of ratings r_p for user p and calculating the mean centred cosine distance with other users' ratings vector
- We can also find top N similar users to user p by using unsupervised ML models such as K Nearest Neighbours
- The top N similar users' ratings for product q are taken and aggregated
- There are different ways to aggregate, simple average or similarity weighted average
- The aggregation forms the estimate of the rating of the product q as rated by user p

Collaborative Filtering | Item - Item

What is the rating of movie 1 by user 5?

		Users												Movies	S(1,m)
		1	2	3	4	5	6	7	8	9	10	11	12		
1	1		3			5			5		4				1
2			5	4				4			2	1	3		-0.178
3	2	4		1	2			3		4	3	5			0.414
4		2	4		5				4			2			-0.101
5			4	3	4	2						2	5		-0.312
6	1		3		3			2				4			0.587

$$\frac{2+3}{2} = 2.5$$

$$\frac{2 \times 0.414 + 3 \times 0.587}{0.414 + 0.587} = 2.586$$

Collaborative Filtering | User - User

What is the rating of movie 1 by user 5?

		Users											
Movies		1	2	3	4	5	6	7	8	9	10	11	12
	1	1		3			5			5		4	
	2			5	4			4			2	1	3
	3	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	6	1		3		3			2			4	
S(5,u)						1							

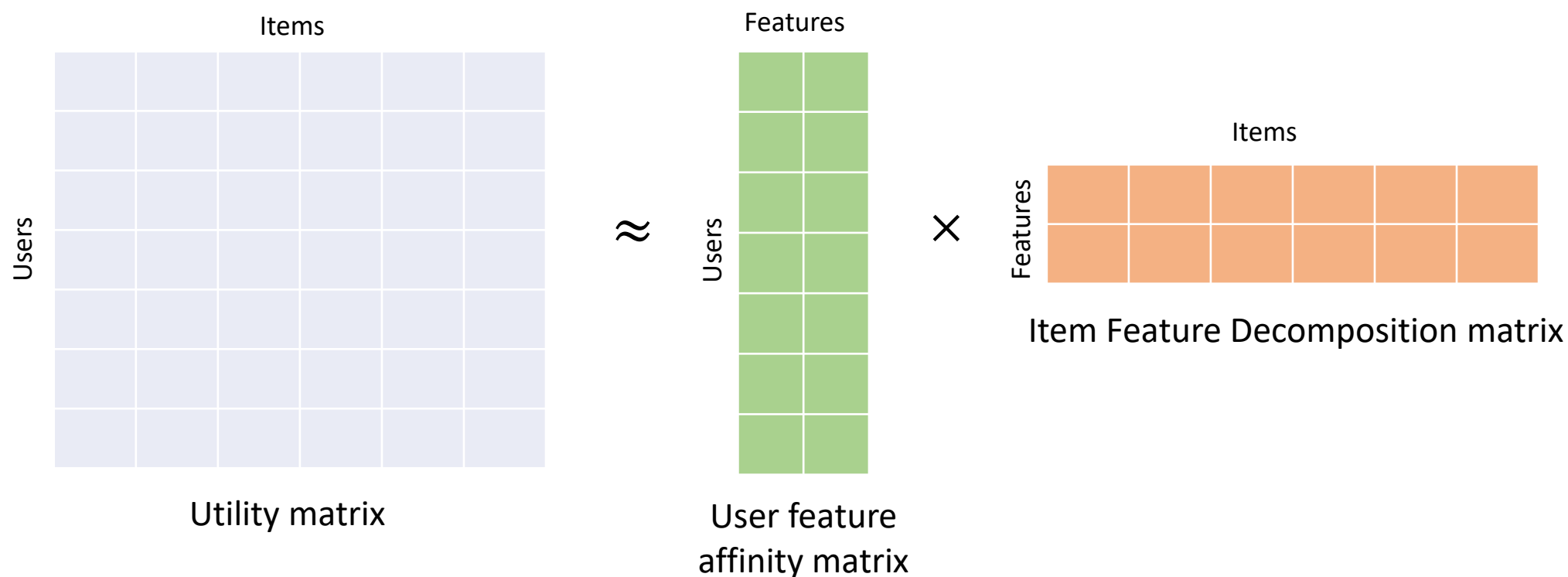
Collaborative Filtering | Caveats

- In theory, user-user and item-item approaches should be complimentary to one another
- In practice, we see that item-item collaborative filtering is much more powerful
- Items are more 'simpler and invariant' than users
 - Items' attributes do not change with time
 - Users' attributes and preferences change with time

Latent Factor Models

Latent Factor Models | Theory

- The utility matrix is a result of interaction of users with items
- Each item can be decomposed into several attributes
- Each person has a particular affinity towards each of the attributes



Latent Factor Models | Theory

- We wish to decompose the utility matrix as a product of two separate matrices
- The decomposed product should approximate the original utility matrix as much as possible
- We choose an appropriate number of features based on
 - Domain knowledge
 - Reconstruction error threshold
- Each item is broken down into combination of one or more features, represented by the item feature decomposition matrix
- Each person has certain affinity towards certain features, represented by the user feature affinity matrix

Latent Factor Models | Theory

- We wish to find matrices P and Q such that

$$\arg \min_{P, Q} \left\| U - PQ^T \right\|_2^2$$

- This can be solved by minimizing the reconstruction loss $\left\| U - PQ^T \right\|_2^2$ using gradient descent
- Since we have to find optimum value of two matrices, we need to alternate between the two matrices to achieve the minima
- The matrix estimation can be made more robust by introducing regularisation in the loss function

Latent Factor Models | Theory

Pseudo Code to minimise reconstruction error and find the optimal P, Q matrices

- Choose appropriate hyperparameters, step size and regularization constant
- For each element in the utility matrix
 - Compute prediction error
 - Update p_u , the user vector
 - Use the updated p_u vector to update q_i , the item vector
 - Stop if convergence criteria satisfied

Latent Factor Models | Caveats

- Estimating P, Q matrices to minimise reconstruction loss can be done by performing Singular Value Decomposition
- Theoretically, SVD ensures decomposition with minimum loss
- This is advantageous as it overcomes the drawbacks of the gradient descent approach
- These models also require orders of magnitude less resources to store and save the data
- They eliminate the cold start problem and popularity bias problem encountered in collaborative filtering

Content Based Filtering

Content Based Filtering | Theory

- We wish to recommend a customer only those items which are similar to the items that the customer has highly rated
- This involves extracting features from the items, using various techniques
 - TF-IDF
 - Word embeddings
- The item features and user likes are used to build a user profile
- The user profile is then matched with each item in the catalogue, and the highest scored items are served as recommendations

Content Based Filtering | Caveats

- No need for data on other users
- Captures unique preferences of user groups
- Able to recommend new and unpopular items
- Explanations for recommended items available, since features are available for items
- Finding appropriate features is hard
- Rabbit holed recommendations, no surprise in recommendations
- Susceptible to cold start problem

Q & A