

# Recommender Systems – Collaborative Filtering

Prof. Rajib L. Saha



[24x7 Customer Support](#) - [Contact Us](#) | [Home](#) | [Account](#) | [Wishlist](#) | [Login](#) | [Signup](#)

[See all Categories](#)

Search: Mobiles & Accessories


 Cart (0)

[Mobiles & Accessories](#)
[Mobile Accessories](#)
[Tablets](#)
[Featured Deals](#)
[New Releases](#)
[Bestsellers](#)
[Hottest Smart Phones](#)
[Recently Sold](#)

[Home](#) > [Mobiles](#) > [Samsung](#) > [Samsung Galaxy R I9103](#)








Rate this product ★★★★★  
[Write a review](#)

### Samsung Galaxy R I9103

★★★★★ 285 Ratings | [56 Reviews](#)

- Android v2.3 OS
- 5 MP Primary Camera
- 1.3 MP Secondary Camera
- 4.19-inch Touchscreen

with 1 year manufacturer warranty for Phone and 6 months warranty for in the box accessories

Price: **Rs. 20999**

(Prices are inclusive of all taxes)

**FREE Home Delivery**

**Buy This Now**  with an option to pay **Cash on Delivery**

**In Stock.**  
Delivered in **2-4 business days.**  
[See Details](#)

**30 DAY** **Replacement Guarantee**  
[Know More »](#)

**0% INTEREST** **EMI Options**  
ICICI, HDFC and Citibank  
[Know more»](#)

**Customers Who Looked At This Mobile Also Viewed**

- 
**Samsung Galaxy S Plus I9001 (Metallic Black)**  
Price: **Rs.22120**
- 
**LG Optimus 2X P990**  
Price: **Rs.19990**  
Cashback: **Rs. 400**
- 
**Samsung Galaxy S LCD I9003**  
Price: **Rs.18440**  
Cashback: **Rs. 1000**
- 
**Sony Ericsson Xperia neo V**  
Price: **Rs.18000**  
Cashback: **Rs. 400**

Online Recommender Systems: collaborative filtering (different from Association Rules!)

# Collaborative Filtering

- If Person A has the same opinion as Person B on an issue, A is more likely to have B's opinion on a different issue  $x$ , when compared to the opinion of a person chosen randomly

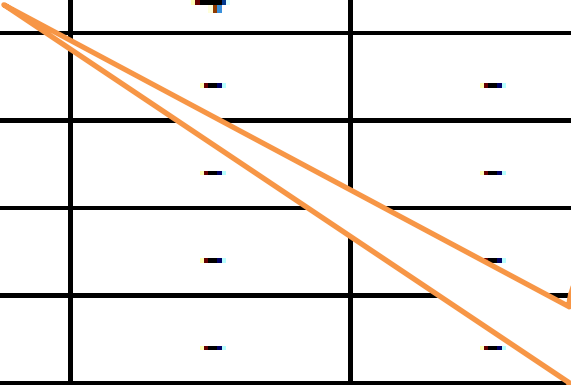
# Traditional Collaborative Filtering

- Customer as a  $p$ -dimensional vector of items
  - $p$ : the number of distinct catalog items
  - Components: bought (1)/not bought (0); ratings; rated (1)/not rated(0).
- Find **Similarity** between Customers A & B

n customers X p items

|          | Item 1 | Item 2 | ... | Item p |
|----------|--------|--------|-----|--------|
| Person 1 | -      | ✓      | -   | -      |
| Person 2 | ✓      | -      | -   | -      |
| Person 3 | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| Person n | -      | -      | -   | ✓      |

|          | Item 1 | Item 2 | ... | Item p |
|----------|--------|--------|-----|--------|
| Person 1 | 3      | 4      | -   | -      |
| Person 2 | 2      | -      | -   | -      |
| Person 3 | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| Person n | 5      | 1      | -   | -      |



These entries could be numbers (ratings) as well

Information on rating along with the information on whether items were rated or not rated can be better than either of them alone. **Example: Netflix Recommendation**

# Cosine-based similarity

$$\text{Cos}(\mathbf{A}, \mathbf{B}) = \mathbf{A} \cdot \mathbf{B} / |\mathbf{A}| * |\mathbf{B}|$$

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

- $\mathbf{A}$ :  $(a_1, a_2, \dots, a_N)$
- $\mathbf{B}$ :  $(b_1, b_2, \dots, b_N)$
- $\mathbf{A} \cdot \mathbf{B}$ :  $a_1 * b_1 + a_2 * b_2 + \dots + a_N * b_N$
- $|\mathbf{A}|$ :  $(a_1^2 + a_2^2 + \dots + a_N^2)^{1/2}$ ;  $|\mathbf{B}|$ :  $(b_1^2 + b_2^2 + \dots + b_N^2)^{1/2}$

Example:

|          | 1 | 2 | 3 | 4 |
|----------|---|---|---|---|
| <b>A</b> | 3 | 5 | 0 | 1 |
| <b>B</b> | 1 | 4 | 0 | 0 |

$\text{Cos}(\mathbf{A}, \mathbf{B}) =$

$$(3 * 1 + 5 * 4 + 0 * 0 + 1 * 0) / ((3^2 + 5^2 + 0^2 + 1^2)^{1/2} * (1^2 + 4^2 + 0^2 + 0^2)^{1/2}) = 0.94$$

# Correlation-based Similarity

$$\text{sim}(i, j) = \text{corr}_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}.$$

|          | 1 | 2 | 3 | 4 |
|----------|---|---|---|---|
| <b>A</b> | 3 | 5 | 0 | 1 |
| <b>B</b> | 1 | 4 | 0 | 0 |

$$\text{Corr}_{AB} = \frac{\text{Covariance}(A, B)}{\text{Stdev}(A) * \text{Stdev}(B)}$$



# n customers X p items

|          | Item 1 | Item 2 | ... | Item p |
|----------|--------|--------|-----|--------|
| Person 1 | 3      | 4      | -   | -      |
| Person 2 | 2      | -      | -   | -      |
| Person 3 | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| Person n | 5      | 1      | -   | -      |

While computing similarity between Persons 1 & 2, Item 2's rating cannot be included, since Person 2 hasn't bought Item 2.

For binary data (bought/didn't buy), it is NOT an issue.

# Normalizing Ratings

- Multiply the vector components by the **inverse frequency**
- **Inverse frequency:** the inverse of the number of customers who have purchased or rated the item

# Other measures ...

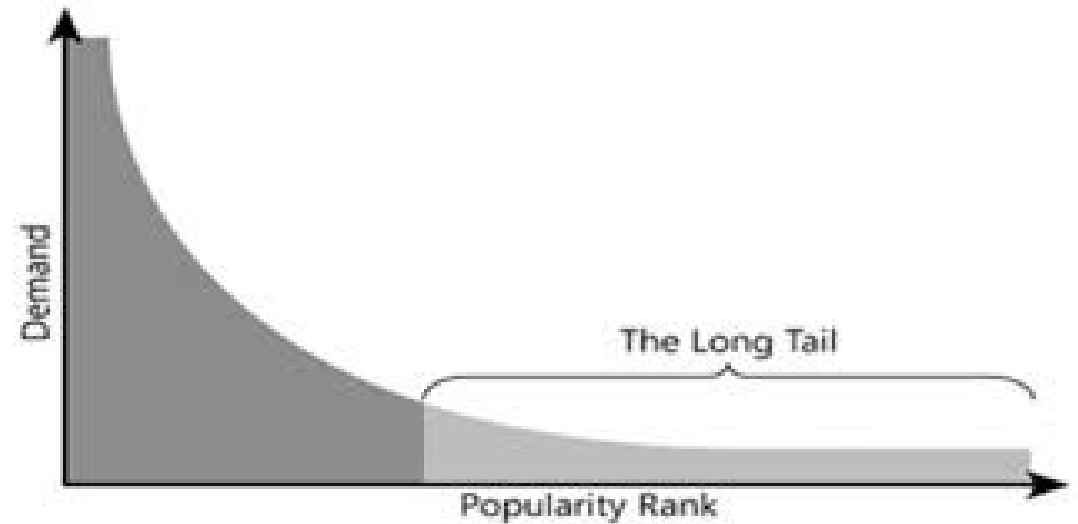
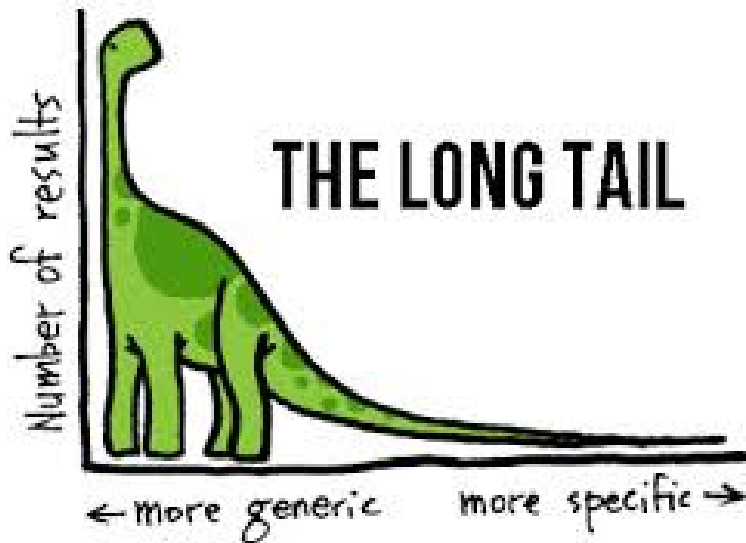
- Find **Nearest Neighbor(s)** based on **distance** (dissimilarity)
  - Can use other ***Distance*** measures to identify neighbors
    - Euclidean distance
$$= \sqrt{(3-1)^2 + (5-4)^2 + (0-0)^2 + (1-0)^2}$$
    - Manhattan distance
$$= (|3-1| + |5-4| + |0-0| + |1-0|)$$

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| A | 3 | 5 | 0 | 1 |
| B | 1 | 4 | 0 | 0 |

# Once similar, what item(s) to recommend?

- The item that hasn't been bought by the user yet
- You may create a list of multiple items to be considered for recommendation
- Finally, recommend the item he/she is **most likely** to buy
  - Rank each item according to how many similar customers purchased it
  - Or rated by most
  - Or highest rated
  - Or some other popularity criteria

# Long Tail



## Supply-side drivers:

- centralized warehousing with more offerings
- lower inventory cost of electronic products

## Demand-side drivers:

- search engines
- **recommender systems**

# Negatives

- Memory-based / Lazy-learning
  - **When** does the recommendation engine compute the “recommendation”?
- Computation-intensive
  - Recall **how** it computes “recommendation”?  $n^2$  similarities

|          | Item 1 | Item 2 | ... | Item p |
|----------|--------|--------|-----|--------|
| Person 1 | √      | √      | -   | -      |
| Person 2 | √      | -      | -   | -      |
| Person 3 | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| Person n | -      | -      | -   | √      |

# How to reduce computation?

- Randomly sample customers
- Discard infrequent buyers
- Discard items that are very popular or very unpopular
- Clustering can reduce #of rows
- PCA can reduce #of columns

|          | Item 1 | Item 2 | ... | Item p |
|----------|--------|--------|-----|--------|
| Person 1 | √      | √      | -   | -      |
| Person 2 | √      | -      | -   | -      |
| Person 3 | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| .        | -      | -      | -   | -      |
| Person n | -      | -      | -   | √      |

# Runtime vs. Quality of Recommendation

- Recommend while the customer is browsing

vis-à-vis

- Recommend **better** but **later**

HOME > RECOMMENDED PRODUCTS

## Cut Veg Salad (ready to eat)

Cleaned, washed, dried.  
With salad dressing

BUY NOW



## Playa Hotels & Resorts: All-inclusive properties



**Special offer:** Save up to 55% on your stay, plus kids stay free.

Playa Resorts are all-inclusive, all-suite properties located in Cancun and the Riviera Maya. With two adult-only properties (The Royal Cancun and The Royal Playa Carmen) and two family properties (Gran Caribe Real and Gran Porto Real), you have several vacation options to choose from. Whether you're looking for a romantic escape or a fun-filled family getaway, Playa Resorts will exceed your expectations.

GET DETAILS >



# Search-based Methods

- Based on previous purchases
  - Books of the same/similar authors
  - DVD titles of the same director
  - Products that are identified by similar keywords

A more sophisticated variation of the search-based methods:

### **Item-to-Item Collaborative Filtering**

- Cosine similarity among items
  - Item being the vector
  - Customers as components of the vector
- Correlation similarity among items
  - Correlation of ratings of Items I & J  
**where users rated both I & J**

|        | Person 1 | Person 2 | ... | Person n |
|--------|----------|----------|-----|----------|
| Item 1 | 3        | 2        | -   | 5        |
| Item 2 | 4        | -        | -   | 1        |
| Item 3 | -        | -        | -   | -        |
| .      | -        | -        |     | -        |
| .      | -        | -        |     |          |
| .      | -        | -        |     |          |
| Item p | -        | -        |     |          |

While computing similarity between Items 1 & 2, Person 2's rating cannot be included, since Person 2 hasn't bought Item 2

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

$$sim(i, j) = corr_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

# Scalability and Performance of Item-to-Item Collaborative Filtering

- Computation-expensive, **however**, similar-items table is computed offline
- Online component: lookup similar items for the user's purchases & ratings
- Dependent only on how many titles the user has purchased or rated

# Disadvantages of Item-based Collaborative Filtering

- Less diversity between items, compared to users' taste, therefore the recommendations are often obvious
- When considering what to recommend to a user, who purchased a popular item, the association rules and item-based collaborative filtering might yield the same recommendation, whereas the user-based recommendation will likely differ.

# Association Rules vs. Recommender Systems

## Market basket analysis (Association rules)

- Finds many baskets that contain the same items
- Need lots of baskets
- Used for generating *impersonal, common* strategies
- Useful for large physical stores (chains, supermarkets)



## Recommender Systems (Collaborative filtering)

- Finds items which have a large fraction of their baskets in common
- Number of baskets unimportant
- Used for **personalization**
- Useful for online recommendation (long-tail)



# A Critical Limitation of Collaborative Filtering

- Cold Start:
  - How to create recommendation for new users
  - How about new items

# How to address Cold Start?

- Approaches to address cold start with new users:
  - Popular items (get quick reaction of the users)
  - Demographically relevant items
  - Browsing history
  - Secondary source of data --- social network, subscription
  - Netflix – start with rating a few movies
- Approaches to address cold start with new items:
  - Recommend to random users/ or some selective users based on certain criteria
  - How about offering the product to influential people in the social network



# Issues with Rating Matrix-based Recommender Systems

- Person i likes Adventure Movie 1 (AM1), but has never watched or rated Adventure Movie 2 (AM2); whereas Person j has watched AM2 but not watched/rated AM1.
- What happens to the similarity between Persons i & j?

|          |   |      |     |   |
|----------|---|------|-----|---|
|          | - | AM 1 | AM2 | - |
| .        | - | -    | -   | - |
| Person i | - | 4    | -   | - |
| Person j | - | -    | 3   | - |
| .        | - | -    | -   | - |

- The ratings of AM1 and AM2 are not even included while computing similarity!
- Rating matrices are huge, and usually sparse. What is the implication on computational burden on dealing with the rating matrix?

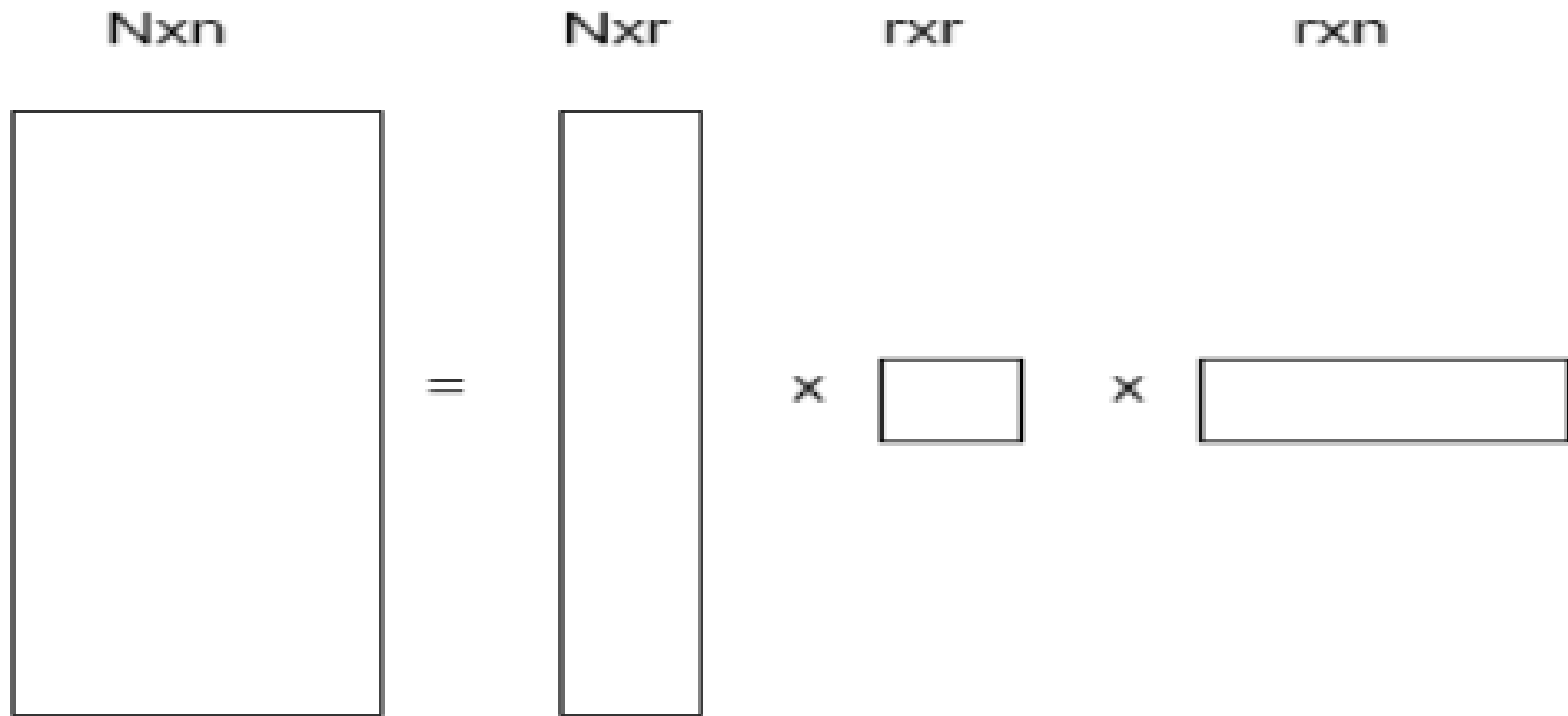
# Recommendation using SVD

$$R = U \Sigma V^T$$

$\mathbf{R}_{N \times n}$ : Rating Matrix

$\mathbf{U}_{N \times r}$ : User-feature matrix

$\mathbf{V}_{n \times r}$ : Item-feature matrix



# Dealing with New Users

$$R = U \Sigma V^T$$

- $r_i = i_{\text{th}}$  row of rating matrix = item ratings of user  $i$
- $u_i = i_{\text{th}}$  row of user-feature matrix = feature ratings of user  $i$
- $r_i = u_i \Sigma V^T$  {dimension:  $1 \times n = 1 \times r \times r \times n$ }
- $r_i V = u_i \Sigma V^T V = u_i \Sigma$
- $r_i V \Sigma^{-1} = u_i \Sigma \Sigma^{-1} = u_i$
- $u_{\text{new}} = r_{\text{new}} V \Sigma^{-1}$ 
  - Let the new users rate a few items and use those partial ratings to compute feature ratings

# Dealing with missing values before applying SVD

- Impute the missing values in the Rank matrix with user mean or item mean
- If the rank matrix is already normalized (mean-subtracted), missing values can be simply zeroes

# Vulnerability of Recommender Systems

- Recommender accuracy and neutrality may be compromised
  - malicious users may try to push or kill a product through using fake accounts
  - inconsistent ratings
  - Some mechanism to establish integrity is necessary
- Privacy of users' ratings or preferences can be compromised
  - In some systems it may be desired that users do not get to know each other's opinions
  - Use of less transparent algorithm is less vulnerable to hack
  - SVD decomposition

# Amazon Recommender System

- Non-personal, based on sales statistics
  - Best sellers, promotional, etc.
- Recommendations based on browsing history, personal
- Recommendations based on Association Rules, non-personal
- Personalized recommendation: *Sign in > Account > Your Recommendations*
- Personalized recommendation over email

# Netflix Recommender System

- How did Cinematch work?
- How did Cinematch added value to Netflix?
- How did Netflix created itself an advantage through Long Tail effect?
- Blockbuster's data disadvantage/The Competitive advantage of Netflix
- Opportunities of Netflix in the Video-On-Demand market