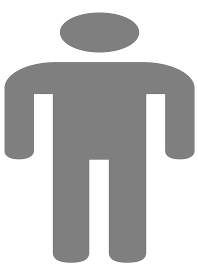
Slide-1

Recommender Systems - Collaborative Filtering

- E-commerce website advertisements showing previously viewed products from the website.
- Matrimonial portal
- Naukri job portal recommends jobs based on our skills.

Slide-2

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.

Slide-3

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)
    - Number of products purchased
- Find Similarity between Customers A & B

Slide-4

Collaborative Filtering

Items rating

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

n customer x p items

	Item 1	Item 2	....	Item p
Person 1	1	1	0	0
Person 2	1	1	0	0
Person 3	0	1	0	0
.	0	0	0	0
.	0	0	0	0
.	0	0	0	0
Person n	1	0	0	1

While computing similarity between Persons 1 & 2, item 2's rating cannot be included since Person 2 hasn't bought Item 2.	This is not an issue for binary data (bought / didn't buy)

Slide-5

Similarity Measures

Cos (A, B) = A.B / |A|*|B|

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

	1	2	3	4
A	3	5	4	1
B	1	4	4	2

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

Cos(A, B) = $(3*1 + 5*4 + 4*4 + 1*2)/((3^2+5^2+4^2+1^2)^{1/2}*(1^2+4^2+4^2+2^2)^{1/2}) = 0.94$

Slide-6

Similarity Measures

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

	1	2	3	4
A	3	5	4	1
B	1	4	4	2

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

Slide-7

Normalization & Dissimilarity measures

- Multiply the vector components by the *inverse frequency*
- ***Inverse frequency**:* The inverse of the number of customers who have purchased or rated the item
- Find Nearest Neighbor(s) based on distance
- Can use other Distance measures to identify neighbors

	1	2	3	4
A	3	5	4	1
B	1	4	4	2

✓ Euclidean distance

$= \text{sqrt}((3-1)^2 + (5-4)^2 + (0-0)^2 +(1-0)^2)$

✓ Manhattan distance

$= (|3-1| + |5-4| + |0-0| + |1-0|)$

Slide-8

What items to recommend?

The item that has’t been bought by the user yet

Create a list of multiple items to be considered for recommendation & recommend the item that the person is MOST LIKELY to buy

1) Rank each item according to how many similar customers purchased it

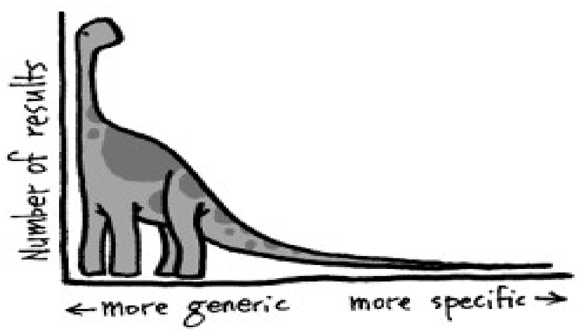
2) Or rated by most

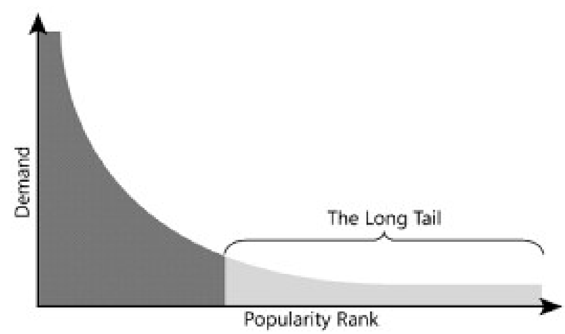
3) Or highest rated

4) Or some other popularity criteria

SLIDE-9

Long Tail





LONG TAIL

Supply-side drivers:

- Centralized warehousing with more offerings
- Lower inventory cost of electronic products

Demand-side drivers:

- Search engines
- Recommender systems

- http://www.ebay.com/sch/Helicopters/63680/bn_16581810/i.html

SLIDE-10

Disadvantages

- 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	1	1	0	0
Person 2	1	0	0	0
Person 3	0	0	0	0
.	0	0	0	0
.	0	0	0	0
.	0	0	0	0
Person n	0	0	0	1

SLIDE-11

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

SLIDE-12

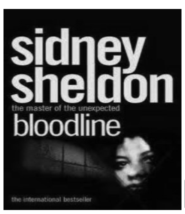
Runtime vs. Quality of recommendation

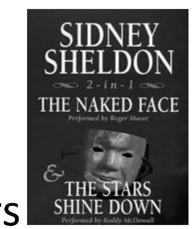
- Recommend while the customer is browsing
- Recommend better but later

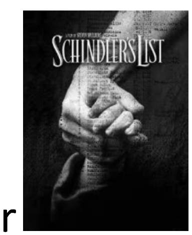
SLIDE-13

Search-based Methods

- Based on previous purchases to reduce computation



Books of the same / similar authors


DVD titles of the same director



Products identified by similar keywords


SLIDE-14

Similarity Measure

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

**Note: *While computing similarity between items 1 & 2, Person 2's rating cannot be included since Person 2 hasn't bought item 2.*

Slide-15

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

Slide-16

Item-to-Item collaborative filtering

Scalability & Performance

- 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

Slide-17

Item-based collaborative filtering

Disadvantage 1

- Less diversity between items, compared to the users' taste, therefore the recommendations are often obvious

Disadvantage 2

- When considering what to recommend to a user, who purchased a popular item, the association rules are item-based collaborative filtering might yield the same recommendation, whereas the user-based recommendation will likely differ

Slide-18

Dichotomies

Association Rules	*Recommender Systems*
• Impersonal, Common, Generic Strategy • No. of baskets is important • Useful for large physical stores	• Personalized strategy • No. of baskets is unimportant • Useful for online recommendation

Slide-19

New customer/item

- Challenges
- New customer
- New product

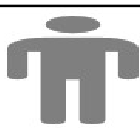
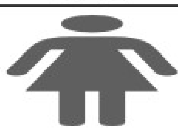
- Popular items
- Demographically relevant items
- Browsing history
- Secondary source of data – social network subscription
- Netflix – start with rating a few movies

- Recommend to random users

- Recommend selective users based on certain criteria
- How about offering the product to influential people in the social network

Slide-20

Netflix kind of Recommender System

	-			-
-	-	-	-	-
	-	4	-	-
	-	-	5	-
-	-	-	-	-

Sparsity & Computational burden

Slide-21

SVD application in Recommendation

R_{Nxn}: Rating Ma ⇦ **$R = U\Sigma V^T$** ⇨ **V_{nxr}: Item - Feature Matrix**

⇩

U_{Nxr}: User - Feature Matrix

r x n **=** **N x n** **x** **N x r** **x** **r x r**

Slide-22

SVD in Recommendation for new users

- r_i = i_{th} row of rating matrix = item ratings of user i
- u_i= i_{th} row of user-feature matrix = feature ratings of user i
- $r_i = u_i \Sigma V^T$ {dimension: 1xn = 1xr rxr rxn}
- $r_i V = u_i \Sigma V^T V = u_i \Sigma$
- $r_i V \Sigma^{-1} = u_i \Sigma \Sigma^{-1} = u_i$
- $u_{new} = r_{new} V \Sigma^{-1}$

– Let the new user rate a few items and use those partial ratings to compute feature ratings

Note:

Missing Values: Impute the missing values in the Rank matrix with 'user mean' or 'item mean'.

Slide-23

Vulnerability of Recommender Systems

1) **Fake accounts are used to push a product by high ratings or kill a product by low ratings**
2) **Accuracy of the recommendation & Neutrality is negatively impacted**
3) **Have user authenticate before rating**
4) **Mobile sms code / Sign in using LinkedIn**

Slide-24

- **Vulnerability of Recommender Systems**

- Privacy - Netflix does not show users the preferences of other users because the movie watching is very private to you & because of hacking issues

- **SVD**
  - Less transparent algorithm
  - Decomposition

Slide-25

Recommender System Cases