Recommendation Systems

Dept. of Comp. Engg.

What is a Recommmendation System?

Recommendation system is an information filtering technique, which provides users with information, which he/she may be interested in.

Examples:

NETFLIX

Video-on-demand provider in North America and UK

- Matches 23 million customers with a huge inventory of movies according to their tastes
- 60 -70% of views result from the recommendations[9]

amazon.com

Gold standard of e-commerce. Pioneer in using recommendations

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected purchases

LinkedIn / facebook

Social and professional networking sites

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected purchases

PANDORA

Music station. Offers music suggestions based on ratings

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected subscriptions[3]

Areas of Use


In E-store
Product Dicovery
Search
Recomm. Engines
Loyalty Programs
Bills and mailers
Email Campaigns
Recommendations for new products, up-sell, cross-sell suggestions and product discovery
Matches varying individual customers taste with a huge inventory
Order of search results on the store can be varied for each customer to aid product discovery
Recommendations pertaining to specific program, suggestions for redemption and program upgrades
Effectively targeting abandoned shopping carts with personalized recommendation
Whitespace can be used for cross-sell and up-sell. Usage can be analyzed for recommending plans.

Why there is a need?

“Getting Information off the internet is like taking a drink from a fire hydrant” - Mitchell Kapor

- Information Overload
- User Experience
- Revenues

Recommender systems help in addressing the information overload problem by retrieving the information desired by the user based on his or similar users' preferences and interests.

Types of Recommendation System

In General, two types of recommender system.

1. Personalized

Registered customers

John visits an online store to buy an accessory for his Blackberry and conducts an online product search. Since John is a registered customer, the recommendations engine draws up information of his previous purchases – a Blackberry phone bought a few months ago – as soon as he logs in and queries for accessories. The engine also draws up in-store trends and recognizes that a majority of customers that buy the same/ similar Blackberry also purchase a particular kind of accessory. The engine collates information about John and the collective and recommends the same accessory to John. An online store with advanced social commerce features can also display comments based on the buying behavior of John's social circle such as 'Your friend George bought this' or 'Your friend Jane reviewed this product' to influence his decision. If John adds the accessory to his online shopping cart, the engine will continue to offer real-time recommendations of products that complement his Blackberry and/or the new accessory. Thus the engine is constantly aware of John's digital actions and refines its recommendations to suit him.

Types... Cont'd

2. Non-Personalized

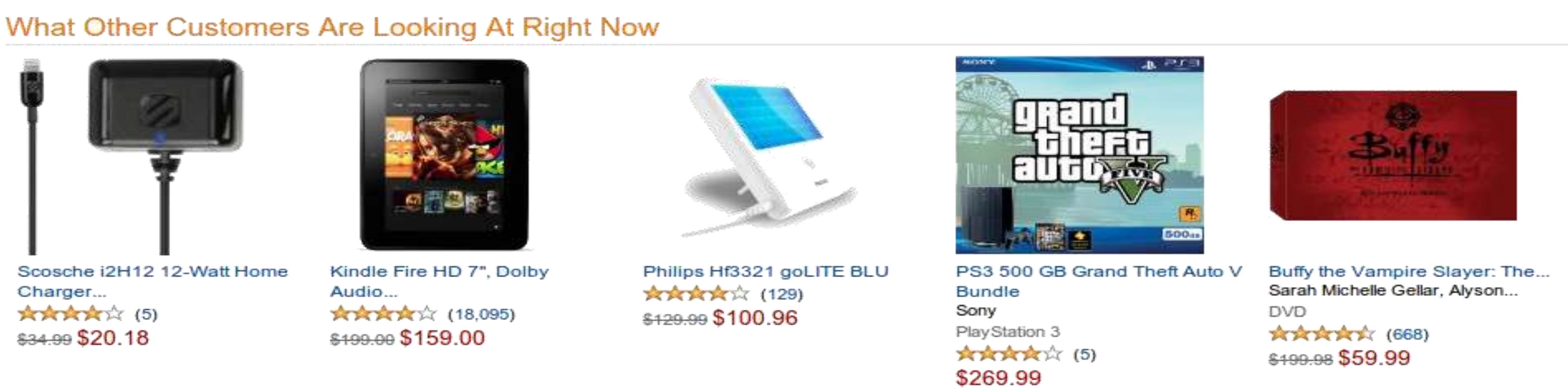
New customers

Taking the context of the above example let us say John is a new visitor to the online store, seeking to make the same type of purchase. Despite having no information about John the engine can offer recommendations about collective preferences in the form of 'Best Sellers'. As John begins to browse a few pages, the engine determines John's preferences and leverages this information to offer recommendations that may interest him.

Types... cont'd

- Personalized

- Non-Personalized



Techniques : Data Acquisition

1. Explicit Data
   - Customer Ratings
   - Feedback
   - Demographics
   - Physiographics
   - Ephemeral Needs

2. Implicit Data
   - Purchase History
   - Click or Browse History

3. Product Information
   - Product Taxonomy
   - Product Attributes
   - Product Descriptions

Techniques : Recommendation Generation

1. **Collaborative Filtering** method finds a subset of users who have similar tastes and preferences to the target user and use this subset for offering recommendations.

Basic Assumptions :

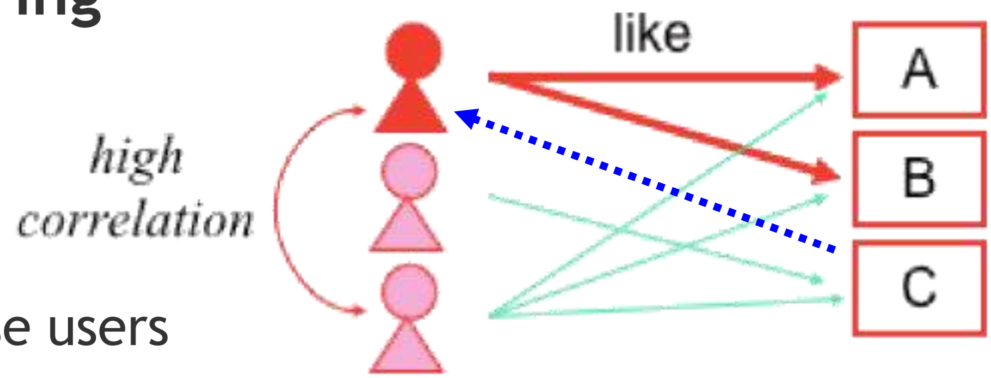
- Users with similar interests have common preferences.
- Sufficiently large number of user preferences are available.

Main Approaches :

- User Based
- Item Based

User-Based Collaborative Filtering

- Use user-item rating matrix
- Make user-to-user correlations
- Find highly correlated users
- Recommend items preferred by those users



Pearson Correlation :

$$userSim(u,n) = \frac{\sum_{i \subset \mathrm{CR_{u,n}}} (r_{ui} - \bar{r}_u)(r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \subset \mathrm{CR_{u,n}}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \subset \mathrm{CR_{u,n}}} (r_{ni} - \bar{r}_n)^2}}$$

Prediction Function :

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)}$$

Techniques : Recommendation Generation

User Based Collaborative Filtering

User ↓ / Item →	I1	I2	I3	I4	I5
U1	5	8		7	8
U2	10		1		
U3	2	2	10	9	9
U4		2	9	9	10
U5	1	5			1
User a	2		9	10	

Recommend items preferred by highly correlated user U3 → I5

Techniques : Recommendation Generation

User Based Collaborative Filtering

- **Advantage :**

  - No knowledge about item features needed

- **Problems :**

  - New user cold start problem

  - New item cold start problem: items with few rating cannot easily be recommended

  - Sparsity problem: If there are many items to be recommended, user/rating matrix is sparse and it hard to find the users who have rated the same item.

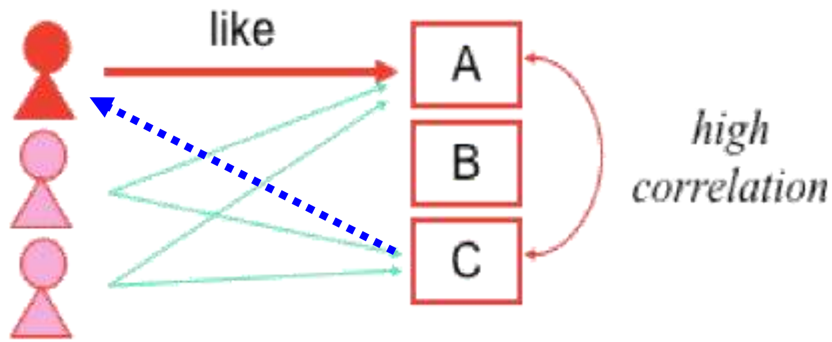
  - Popularity Bias: Tend to recommend only popular items.

  e.g. RINGO, GroupLens

Techniques : Recommendation Generation

Item Based Collaborative Filtering

- Use user-item ratings matrix
- Make item-to-item correlations
- Find items that are highly correlated
- Recommend items with highest correlation



Similarity Metric :

$$itemSim(i, j) = \frac{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \subset RB_{i,j}} (r_{uj} - \bar{r}_u)^2}}$$

Prediction Function :

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} itemSim(i, j) \cdot r_{ui}}{\sum_{j \in ratedItems(u)} itemSim(i, j)}$$

Techniques : Recommendation Generation

Item Based Collaborative Filtering

Item → / User ↓	I1	I2	I3	Item I	I5
U1	5	8		7	8
U2	10		1		
U3	2		10	9	9
U4		2	9	9	10
U5	1	5			1
User a	2		9	10	

Recommend items highly correlated to preferred items → I5

Techniques : Recommendation Generation

Item Based Collaborative Filtering

- **Advantages :**

  - No knowledge about item features needed

  - Better scalability, because correlations between limited number of items instead of very large number of users

  - Reduced sparsity problem

- **Problems :**

  - New user cold start problem

  - New item cold start problem

  e.g. Amazon, eBay

Techniques : Recommendation Generation

2. **Content Based Systems** recommend items similar to those a user has liked (browsed/purchased) in the past.

OR

Recommendations are based on the content of items rather on other user's opinion.

User Profiles: Create user profiles to describe the types of items that user prefers.

e.g. User1 likes sci-fi, action and comedy.

Recommendation on the basis of keywords are also classified under content based. e.g. Letizia

e.g. IMDB, Last.fm(scrobbler)

Techniques : Recommendation Generation

Content Based Systems Cont'd...

Advantages:

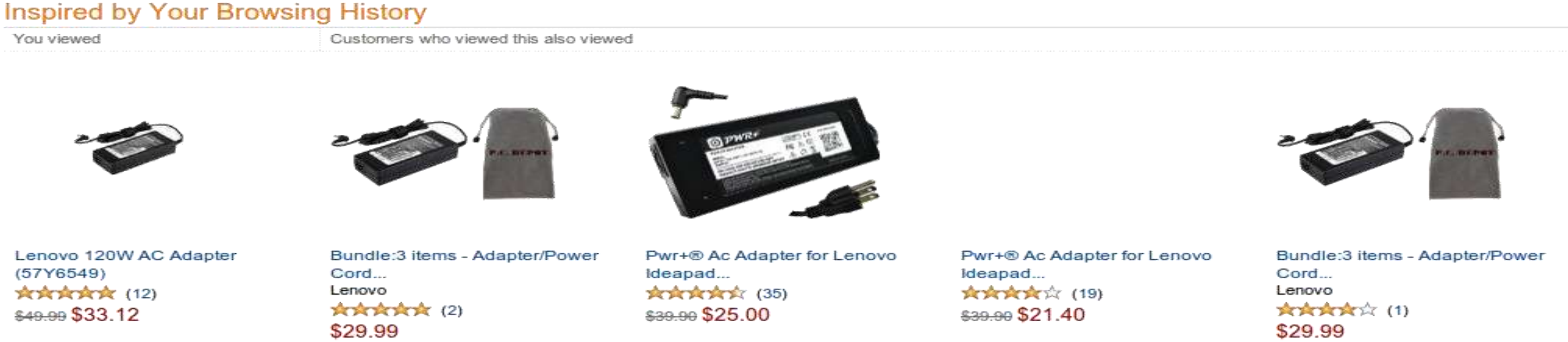
- No need for data on other users. No cold start and sparsity.
- Able to recommend users with unique taste.
- Able to recommend new and unpopular items.
- Can provide explanation for recommendation.

Limitations:

- Data should be in structured format.
- Unable to use quality judgements from other users.

Case Study

1. Amazon



2. YouTube

Amazon's Demand and Solution

Demand :

- Amazon had more than 29 million customers and several million catalog items.
- Amazon use recommendation algorithms to personalize the online store for each customer in real time.

Solution :

- Existing algorithm were evaluated over small data sets.
  - Reduce M by randomly sampling the customers or discarding customers with few purchases. (M : the number of customers)
  - Reduce N by discarding very popular or unpopular items. (N : the number of items)
  - Dimensionality reduction techniques such as clustering.

The Amazon Item-to-Item Collaborative Filtering Algorithm

Algorithm :

For each item in product catalog, I_1
For each customer C who purchased I_1
For each item I_2 purchased by customer C
Record that a customer purchased I_1 and I_2
For each item I_2
Compute the similarity between I_1 and I_2

Algorithm Complexity :

- Worst Case : $O(N^2M)$
- In practice : $O(NM)$, 'cause customers have fewer purchases.

YouTube Recommendation System

Related Videos

1. A given time period, we count for each pair of videos v_i, v_j how often they were co-watched.
2. We denote this co-visitation count by $c_i j$.
3. We define **r**elated score of video v_j to v_i as :
   $r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$
4. $f(v_i, v_j)$ is a normalization function that denote the global popularity
5. Pick the set of related videos R_i for a given seed video v_i as the top N candidate videos ranked by their scores $r(v_i, v_j)$.

YouTube Recommendation System

Generating Recommendation Candidates

1. A given seed set S (e.g. the videos user watched)
2. Each video v_i in the seed set consider its related videos R_i
3. Denote the union of these related video sets as C_1:

$$C_1(S) = \bigcup_{v_i \in C_S} R_i$$

4. Distance of n from any video in the seed set:

$$C_n(S) = \bigcup_{v_i \in C_{n-1}} R_i$$

YouTube Recommendation System

Generating Recommendation Candidates

- The final candidate set C_{final} :

$$C_{final} = (\bigcup_{i=0}^{N} C_i) - S$$

- Due to the high branching factor of the related videos graph we found expanding over a small distance yielded broad and diverse recommendations even for users with a small seed set.

Ranking

Using a linear combination of three kinds of signals(a.video quality b.user specifility c.diversification),we generate a ranked list of the candidate videos.

Q & A

no question off limits