19CSE337 Social Networking and Security

Lecture 21

Topics to Discuss

- Recommendation Systems
- Content based recommenders

Recommender System

- Social media users will make a variety of decisions on a daily basis.
- These decisions can be about buying a product, purchasing a service, adding a friend, renting a movie etc.
- There are diverse options, but the limited knowledge that individual has create a desire for external help.
- We have search engines to suggest these services but the results are rarely tailored to our particular tastes.
- So we need specific algorithms to recommend services tailored to our tastes.
- These algorithms are called recommendation algorithms or systems.

Recommender System

- Recommender systems are commonly used for product recommendation.
- Formally, recommendation systems takes a set of users U and set of items I and learns a function f:UxI->R, where R is a real value indicates the interest of user U in item I.
- The recommendation algorithms can be generalized to recommend movies, ads, content etc.

Challenges

- **Cold-start problem:** many recommendation algorithms use users' historical information (purchase history, ratings, browsing history, friends, user profile etc) to recommend products. But when a user opened a new account, there is no history to make a recommendation. This problem is referred to as cold-start problem.
- To solve cold start problem, some sites will ask users to complete their profile information, give ratings to some products, ask to fill survey forms etc.

Challenges

- **Data Sparsity:** similar to cold-start problem. Data sparsity occurs when not enough historical or prior information is there.
- Data sparsity occurs at system level not at individual level.
- It happens when few users rate items whereas some not.
- The problem is most common in newly launched websites or ones that are not popular.

Challenges

- **Attacks:** the recommender systems may be attacked to recommend not recommended items.
  - **Push attacks:** pushing rating up by fake users. (eg: item A recommended along with item B as both of them are competent to each other and having same rating, but after a push attack, item C gets more rating and is recommended along with item B instead of item A. But, in reality, item C is not competent with item B).
  - **Nuke attacks:** DDoS attacks, stops the entire recommendation system.

Challenges

- **Privacy:** the more information a recommender system has about the users, the better the recommendations it provides to the users.
- However, users often avoid revealing information about themselves due to privacy concerns.

Challenges

- **Explanation:** recommendation system provide recommendation without any explanation of why they did so.
- Several items are bought together by many customers without any reason.
- But based on the combination, we can’t recommend items.
- Algorithms should give appropriate explanations.

Classical Recommendation Algorithms

- Two main categories.
  - **Content based systems:** examine properties of items.
  - **Collaborative filtering systems:** examine similarity between users and or items.

Content Based Systems

- As the name says content based recommendation works based on the content of user or item.
- The term content is referred to as the attributes or features.
- The goal behind content based filtering is to classify products with specific keywords, learn what the customer likes, look up those terms in the database, and then recommend similar things.

Content Based Systems

- The base idea is like, user A likes item 1 and item 2.
- Item 3 is similar to item 1 and item 4 is similar to item 2.
- According to content based algorithms, user A will receive recommendation for item 3 and item 4.

Content Based Systems

- Content based systems are based on the fact that a user’s interest should match the description of the items that are recommended by the system.
- In other words, the more similar the item’s description to the user’s interest, the higher the likelihood that the user is going to find the item’s recommendation interesting.

Content Based Systems

- Content based recommender systems implement this idea by measuring the similarity between an item’s description and the user’s profile information.
- The higher this similarity, the higher the chance that the item is recommended.

Content Based Systems

- **Item profile**: features of item to be recommended. For eg: in movie recommendation system, item profile of movie can contain name, actors, director, genre, release date etc.
- **User profile**: user's features like browsing history, preferences and tastes, age, profession etc.

Content Based Systems

- To formalize content based method, vectorize user profiles and item descriptions using a set of k key words.
- After vectorization, item j can be represented as $I_j=\{i_{j,1}, i_{j,2}, ... i_{j,k}\}$ and user i as $U_i=\{u_{i,1}, u_{i,2}, ... u_{i,k}\}$.
- Use cosine similarity to compute similarity between user i and item j.

Content Based Systems

$$sim(U_i, I_j) = cos(U_i, I_j) = \frac{\sum_{l=1}^{k} u_{i,l} i_{j,l}}{\sqrt{\sum_{l=1}^{k} {u_{i,l}}^2}\sqrt{\sum_{l=1}^{k} {i_{j,l}}^2}}$$

- In content based recommendation, compute the top most similar items to a user and then recommend these items in the order of similarity.

Content Based Systems

Algorithm 9.1 Content-based recommendation

Require: User i's Profile Information, Item descriptions for items $j \in \{1, 2, \ldots, n\}$, k keywords, r number of recommendations.

1: **return** r recommended items.
2: $U_i = (u_1, u_2, \ldots, u_k)$ = user i's profile vector;
3: $\{I_j\}_{j=1}^n = \{(i_{j,1}, i_{j,2}, \ldots, i_{j,k})$ = item j's description vector$\}_{j=1}^n$;
4: $s_{i,j} = sim(U_i, I_j), 1 \leq j \leq n$;
5: Return top r items with maximum similarity $s_{i,j}$.

Problem

- Consider the given movie profile and movie ratings given by user A. Generate the user profile and recommend appropriate movies for the user.

ID	Movie Name	Movie Genre
1	Batsman vs Superman	Adventure, Super hero
2	Guardians of the Galaxy	Comedy, Adventure, Super hero, Scifi
3	Captain America Civil War	Comedy, Super hero
4	Hitchhiker's guide to the galaxy	Comedy, Super hero, Scifi
5	Batsman Begin	Super hero
6	Spider Man	Comedy, Super hero

Movie ID	Rating
1	2
2	10
3	8

Solution

- Step-1 Generate movie matrix for the movies user already watched.

$$\begin{array}{cc} \text{ID} & \begin{matrix} \text{Comedy} & \text{Adventure} & \text{Super Hero} & \text{Scifi} \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- Let the matrix be B.

Solution

- Step-2 Generate weighted genre matrix for the movies user already watched.
- Multiply each row of matrix B with its corresponding user rating. Note: not like normal matrix multiplication. (Scalar multiplication with vector).

$$\begin{bmatrix}2\\10\\8\end{bmatrix} \; \mathbf{x} \; \begin{array}{c} \text{ID} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc}\text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi}\end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \text{ID} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc}\text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi}\end{array} \\ \begin{pmatrix} 0 & 2 & 2 & 0 \\ 10 & 10 & 10 & 10 \\ 8 & 0 & 8 & 0 \end{pmatrix} \end{array}$$

Solution

- Step-3 Generate weighted user profile of active user.
- To find, add each column of weighted genre matrix.

ID	Com.	Adv.	Sup.	Scifi
1	0	2	2	0
2	10	10	10	10
3	8	0	8	0
	18	12	20	10
Normalise the score (0-1 range)	0.3	0.2	0.33	0.16

This user scored high for super hero genre. So it is good to recommend movies in superhero genre.

Solution

- Step-4 To recommend available movies (ID: 4,5,6), generate candidate movie matrix.
- Let the name be C.

$$
\begin{array}{c|cccc}
\text{ID} & \text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi} \\
\hline
4 & 1 & 0 & 1 & 1 \\
5 & 0 & 0 & 1 & 0 \\
6 & 1 & 0 & 1 & 0
\end{array}
$$

Solution

- Step-5 Generate weighted candidate movie matrix.
- Multiply each column of matrix C with its corresponding normalised user profile. Note: not like normal matrix multiplication. (Scalar multiplication with vector).

$$
\begin{pmatrix} 0.3 \\ 0.2 \\ 0.33 \\ 0.16 \end{pmatrix}
\times
\begin{array}{c} \text{ID} \\ 4 \\ 5 \\ 6 \end{array}
\begin{pmatrix} \text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi} \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}
=
\begin{array}{c} \text{ID} \\ 4 \\ 5 \\ 6 \end{array}
\begin{pmatrix} \text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi} \\ 0.3 & 0 & 0.33 & 0.16 \\ 0 & 0 & 0.33 & 0 \\ 0.3 & 0 & 0.33 & 0 \end{pmatrix}
$$

Solution

- Step-6 Calculate aggregate scores of each genre. (row wise addition of weighted candidate matrix).
- Known as recommendation matrix.

$$\begin{array}{c|c} \text{ID} & \begin{array}{cccc} \text{Com.} & \text{Adv.} & \text{Sup.} & \text{Scifi} \end{array} \\ \begin{array}{c} 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} 0.3 & 0 & 0.33 & 0.16 \\ 0 & 0 & 0.33 & 0 \\ 0.3 & 0 & 0.33 & 0 \end{pmatrix} \end{array} = \begin{bmatrix} 0.79 \\ 0.33 \\ 0.63 \end{bmatrix}$$

Here, the highest score is for movie ID 4. Recommend movie 4 for this user.

The missing rates for the candidate movies are 7.9, 3.3, 6.3

Content Based Systems

- **Advantages**
  - No need for data of other users.
  - Able to make recommendation for users with unique interests.
  - Able to recommend new and unpopular items.
  - No first rater problem.
  - Able to provide explanations (because of features).

Content Based Systems

- **Disadvantages**
  - Feature selection is hard (have to use some other algorithms eg;TF-IDF).
  - Recommendation for new users is difficult (how to create user profile?).
  - Over specialization (will not recommend products outside user’s profile).

Thanks…………