Framework for recommendation system

By:- (Group 15)
19BTRCR058-ASMITA GURUNG
19BTRCR059-BANDANA RAWAL
19BTRCR060-EMRANUL HAQUE RAQUE
19BTRCR061-NISHEN GANEGODA

Problem Definition

- In most common formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user. Formally,
  - Let C be the set of users,
  - Let S be the set of all possible items,
  - Let utility function u: $C \times S \rightarrow R, where\ R\ is\ a\ totally\ ordered\ set$

$$\forall c \in C, s'_c = argmax_{s \in S} u(c, s)$$

Fundamental Recommendation Approaches

- Collaborative (collaborative filtering) recommendations
  - Recommend items that people with similar tastes and preferences liked in the past
- Content-based recommendations
  - Recommend items similar to the ones the user preferred in the past
- Hybrid approaches
  - Combine several approaches together

Collaborative Recommendation

- Automate the process of seeking advices from our trusted people
  - Weight all users with respect to similarity with the active users
  - Select a subset of users (neighbors) to use as recommenders
  - Predict the rating of active user for specific items
  - Recommend items with maximum prediction
- Memory-based and Model-based
- Example (memory-based)

	Shrek	Snow-white	Superman
Alice	Like	Like	Dislike
Bob	?	Dislike	Like
Chris	Like	Like	Dislike
Jon	Like	Like	?

Memory-based Recommendation

- The value of the unknown rating $r_{c,s}$ for user c and item s is usually computed as an aggregate of the ratings of some other users for the same item s.
- The predictive rating:

$$r_{c,s} = \bar{r}_c + k \sum_{c' \in \hat{C}} sim(c, c') \times (r_{c',s} - \bar{r}_{c'})$$

- The mean ratings for user i:

$$\bar{r}_c = (1/|S_c|) \sum\nolimits_{s \in S_c} r_{c,s}, \text{ where } S_c = \{s \in S | r_{c,s} \neq \oslash\}$$

Memory-based Recommendation (Cont.)

- Cosine based similarity between users:

$$sim(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} = \frac{\sum\limits_{s\in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum\limits_{s\in S_{xy}} r_{x,s}^2}\sqrt{\sum\limits_{s\in S_{xy}} r_{y,s}^2}}$$

- Person based similarity between users:

$$sim(x,y) = \frac{\sum\limits_{s\in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum\limits_{s\in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum\limits_{s\in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}$$

Compute Similarities between Items

- $\overline{r_{c,s}} = \kappa \sum_{k=1}^{|S|} sim(k,s) r_k$
- Example

	Shrek	Snow-white	Superman
Alice	Like	Like	Dislike
Bob	?	Dislike	Like
Chris	Like	Like	Dislike
Jon	Like	Like	?

Model-based Recommendation

- Unknown ratings are calculated as (an example)

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)$$

- Estimate model:
  - Cluster models: like-minded users are clustered into classes. Given the user's class membership, the user ratings are assumed to be independent, i.e., the model structure is that of a naive Bayesian
  - Bayesian Networks: represents each item in the domain as a node in as Bayesian network

Model-based Recommendation (Cont.)

- Machine Learning Framework Models
  - Artificial neural networks, etc.
  - Outperforms memory-based approaches in accuracy but no theoretical evidence supporting is provided
- Statistical Models
  - K-means clustering, Gibbs sampling
- Other Models
  - Bayesian model, probabilistic relational model, linear regression, maximum entropy model, latent Dirichlet Allocation, etc.
  - Recently: view the recommendation process as a sequential decision problem and propose using Markov decision processes

Problems

- Cold Start
  - New User
  - New Item
- Sparsity
- Practical Challenges
  - Rating data is often sparse and pairs of users with few co-ratings are prone to skewed correlations
  - Fails to incorporate agreement about an item in the population as a whole
  - Calculating a user's perfect neighborhood is expensive

Content-based Recommendation

- $u(c, s)$ of item s for user c is estimated based on the utilities assigned by user c to item $s_i \in S$ that are "similar" to item s
- Let $Content(s)$ be an item profile, i.e., a set of attributes characterizing item s
- Let $ContentBasedProfile(c)$ be the profile of user c containing tastes and preferences of this user, which can be defined as $(w_{c1}, \ldots, w_{ck})$, where each w_{ci} denotes the importance of keyword (attribute) k_i to user c

$$u(c, s) = score(ContentBasedProfile(c), Content(s))$$

Content-based Recommendation (Cont.)

- The utility function is usually represented by some scoring heuristic, such as the cosine similarity measure

$$u(c,s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{||\vec{w}_c||_2 \times ||\vec{w}_s||_2}$$

$$= \frac{\sum_{i=1}^{K} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2}\sqrt{\sum_{i=1}^{K} w_{i,s}^2}},$$

- The weighting measure can be defined in several different ways. One of the best-known measure for specifying keyword weights in IR is the **term frequency/inverse document frequency (TF-IDF)**

Hybrid Approaches

- Implementing recommenders separately and combining their predictions
- Incorporating some content-based characteristics into a collaborative approach
- Incorporating some collaborative characteristics into a content-based approach
- Constructing a general unifying model that incorporates both content-based and collaborative characteristics

Combining Separate Recommenders

- Weighted: scores of several recommenders are combined
- Switching: switch between recommenders according to the current situation
- Mixed: present recommendations that are coming from different recommenders
- Cascade: one recommender refines the recommendations by another

Adding Content-based Characteristics to Collaborative Models

- Content-based profile is also used to calculate the similarity between users

- Use the variety of different filterbots—specialized content-analysis agents that act as additional participants in a collaborative filtering community.
  - The users whose ratings agree with some of the filterbots' ratings would be able to receive better recommendation

Adding Collaborative Characteristics to Content-Based Models

- Use some dimensionality reduction technique on a group of content-based profiles
- Example: uses latent semantic indexing (LSI) to create collaborative view of a collection of user profiles

Developing a Single Unifying Recommendation Model

- Content-based and collaborative characteristics in a single rule-based classifier

- Uses the profile information of users and items in a single statistical model

- Uses knowledge-based techniques to improve recommendation accuracy and to address some of the limitations

Extending Capabilities of Recommender Systems

- Comprehensive understanding of users and items
- Multidimensionality of Recommendations: add additional contextual information to the User X Item space
- Multcriteria Ratings: find Pareto optimal solutions, take a linear combination of multiple criteria, optimize the most important criterion, consecutively optimize one criterion at a time
- Nontrusiveness
- Flexibility
- Effectiveness of Recommendations

Social Recommender System

- Motivation: Social Overload
  - Information Overload
  - Interaction Overload
- Target the social media domain
- Aim at coping with the challenge of social overload
- Aim at increasing adoption and engagement
- Often apply personalization techniques
- Utilize social network and content, incorporate short-term interest and long-term interest, Accuracy vs. Serendipity tradeoff

Social Recommender System

- Tag Recommendation
- People Recommendation
- Community Recommendation
- Recommendation for groups
- Recommenders in Enterprise
- Recommenders in Activity Stream
- Problems
  - Cold start
  - Trust and distrust (Reputation), explanaition
  - Temporal Aspects in Social Recommendation

Mobile Recommender System

- Offer personalized, context-sensitive recommendations
- Models
  - Context-Dependent Recommendations
  - Distributed Models, e.g. P2P
  - Proactive Recommendations
- Difficulties:
  - Data is more complex
  - Transplantation problem—recommendations may not apply in all regions

Our doing

- Developers sometimes find restricted to use some libraries and also after a time the length of code increases it can go up to thousands of lines which becomes a limitation for recommendations system.
- So what we are going to do is we will design a framework to make this work simpler and more efficient for the developer at our level.
- In our framework what we are going to do is we will add all the libraries such as **numpy,scikit,scipy,matlplotlib** that are needed from the python and ML so basically we don't have to import the libraries one by one and make the code more lengthy.
- We will just have to use our framework which will contain all the libraries which definitely will make the work easier for the developer.

Thank you