Recommender Systems

CMPT 412/868
Social Computing and Participative Web (Web 2.0)

Jie Zhang
October 27, 2009

Motivations

- Too much user generated content
  - Everyone can publish a website; post articles on blogs
- Online stores with too much inventory
  - Amazon has many books
  - Which movies to watch?
  - Which music to listen to?
- Too many types of ads
  - Which ads should be presented to a particular user?

Too many choices, which ones a user should choose?

Purpose of Recommender Systems

To assist users to find what they want; help them make correct decisions according to needs, interests, and goals

- Examples:
  - Find websites or articles that users are interested in
  - Recommend books that users will likely buy
    - www.amazon.com
  - Recommend movies that users will likely watch
    - www.movielens.org
  - Recommend music that users will likely listen to
  - Present ads that will likely lead to a purchase by users
- More systems having above functionalities will be demonstrated later

Classic Algorithms for Recommendation

- Feature based (Content based)
  - Recommendation based on the commonalities among the items that a user has rated before
  - Example: recommend movies that are similar to what the user has watched before (MovieLens)
- Collaborative filtering
  - Recommendation based on ratings from other users
  - Example: recommend books that other users have bought before (Amazon)
- Hybrid:
  - Feature based + Collaborative filtering

Feature based Recommendation

- General Idea
  - Given a set of items that user has shown interests before or the preferences explicitly specified by user, find the most similar items that user might be interested in now
- Similarity of items is determined based on features
  - Example: movie's features: director, actors, etc.
- Advantages:
  - Takes into account features of items
  - Similarity of items can be processed offline in advance to save online computation time
  - Recommendation right away based on explicit specification of user's preferences (even works for newcomers)

Feature based recommendation

- Gmail Ads
  - http://mail.google.com/mail/help/about_privacy.html
  - "We have ads, but only the good kind"
    - Users will see text ads and links to related pages that are relevant to the content of their messages
- Booklamp
  - http://beta.booklamp.org/reader
  - Pacing, density, dialog, action, description
- Beatunes
  - http://www.beatunes.com
  - Based on features of songs

Feature based recommendation

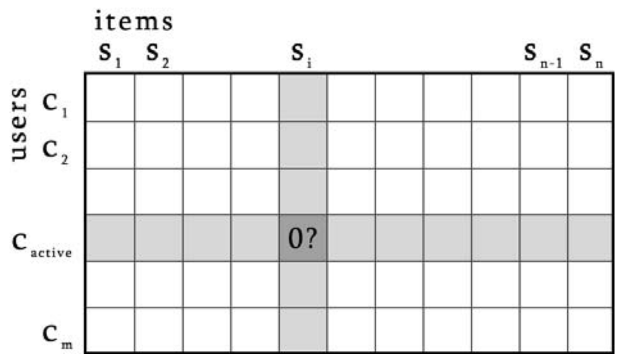
- Limitations:
  - Overspecialization
    - Recommend only items that are similar to those already rated
    - Example: a person with no experience with Greek food would never receive a recommendation for even the greatest Greek restaurant in town
  - Limited content analysis
    - Limited by features that are explicitly associated with objects that these systems recommend, normally manually encoded
    - Features depend on context, type of content
    - Example: features of video content cannot be processed automatically
  - No wisdom of the group

Collaborative Filtering

- General idea
  - Recommendations from nearest neighbors – those who share the most similar interests
  - E.g. ask friends who have the similar tastes of food about whether should go to a restaurant
- Collaborative filtering algorithms
  - *Memory-based*: uses statistical methods to derive predictions from like-minded users
  - *Model-based*: builds models of user ratings through *machine learning* algorithms (e.g. Bayesian networks)
    - Will only see an example later

Memory based CF Algorithm

- *User-Item* matrix: *R*, where entry $r_{c,s}$ indicates the rating score user *c* has given item *s*



Memory based CF Algorithm

- The goal is to predict the score of each empty cell (corresponding to an item) for the *active user*
  - The active user currently requests recommendations
  - The active user may have rated <u>some other items</u> before
  - Recommendations generated based on the scores of the item given by other users who have given similar ratings to <u>the other items</u>
- Once scores for all empty cells have been predicted, a straightforward approach is to recommend the *Top-N* items that have the highest scores.

Memory based CF Algorithm

- If the active user c has not rated item s, but the recommender system can find similar or correlated users c' (i.e. a set of nearest *neighbors* $\hat{C}$) who have, then a rating score can be predicted using:

$$r_{c,s} = k \sum_{c' \in \hat{C}} sim(c, c') \times r_{c',s}$$

  - where $k = 1 / \sum_{c' \in \hat{C}} sim(c, c')$ is a normalizing coefficient
- Users tend to use ratings scales differently
  - For example, on 5-scale rating, the active user may seldom rate 1 or 5 while a neighbour only rates 1 and 5

Memory based CF Algorithm

- More advanced calculation: based on how diverse from the average rating

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

  - where $\bar{r}_c$ is the average rating of the active user
  - $\bar{r}_{c'}$ is the average rating of neighbors
  - *sim (c, c')* is the Pearson correlation coefficient

$$sim\ (c, c') = \frac{\sum_{s \in S_{cc'}} (r_{c,s} - \bar{r}_c)(r_{c',s} - \bar{r}_{c'})}{\sqrt{\sum_{s \in S_{cc'}} (r_{c,s} - \bar{r}_c)^2 \sum_{s \in S_{cc'}} (r_{c',s} - \bar{r}_{c'})^2}}$$

  - $S_{cc'}$ is the set of items rated by both users

Collaborative Filtering Advantages

- Content irrelevant (based on ratings)
  - Domain independent: work in any domain
- No tagging or specified user interests
  - Only keeps track of ratings
  - Requires less effort from users
- Built in community
  - Identify like-minded users (nearest neighbors)
- The most commonly adopted technique in commercial recommender systems
- The most studied technique in academic community

Collaborative Filtering Examples

- Amazon
  - www.amazon.ca
  - Rate books on a five-star scale
  - Shopping for yourself or your niece
- StumbleUpon
  - www.stumbleupon.com
  - Thumb up (like) or thumb down (dislike) for a website
- Last.FM
  - www.last.fm

Limitations

- The lack of ratings presents well-known challenges: if rating information does not exist, then predictions will be inaccurate or undefined (Cold Start Problem)
  - New users must rate some number of items before recommendations can be made,
  - New items must receive ratings before being recommended to users
- The computational complexity of memory-based algorithms requires offline calculations (e.g. at night), and they do not scale well under millions of items and users

Hybrid Recommender Systems

- Collaborative filtering (CF) + Feature based: (FB)
  - Have benefits of both and minimize their problems
- Examples:
  - Collaborative filtering has the cold start problem
    - For example, a new user has not rated many items
    - Feature based method can recommend items based on user specified interests and so on
  - Feature based method doesn't use wisdom of groups
    - Collaborative filtering method recommends items based on ratings of other users
  - Collaborative filtering method requires a lot of computation
    - Feature based method can be processed offline in advance

Hybrid Recommender Systems

- IMDB (The Internet Movie Database)
  - www.imdb.com/chart/top
  - CF based on user ratings
  - FB based on genres, directors, actors, tags etc.
  - FB to narrow search, CF ratings to sort search results
- MovieLens
  - Explicit ratings upon joining for cold start
  - Features of movies, such as tags, for FB
  - 5 star ratings for CF

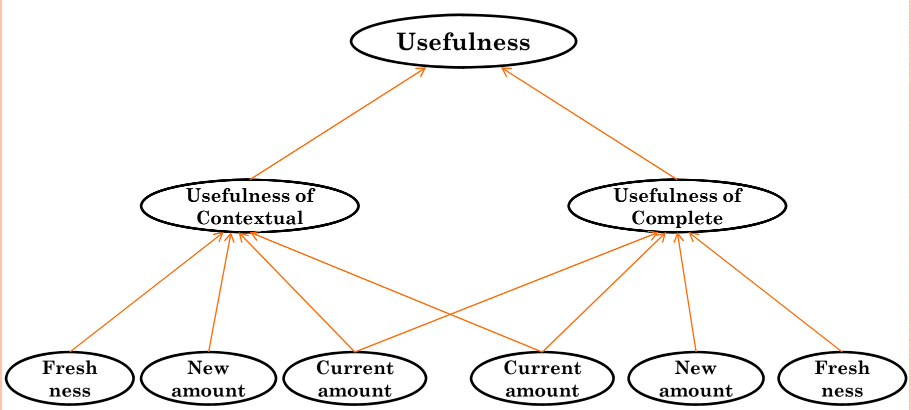
Recommender on Social Web

- Perkpipe
  - A universal recommender on today's social web
  - www.perkpipe.com
  - Provide recommendation based on
    - Social contacts
    - Web histories
    - Their relationships
  - Two types of recommendations
    - By examining users' web histories and social connections, tell what you're really into
    - Tell how you are similar with many others

Recommender on Social Web

- A social network based approach to personalized recommendation of participatory media content (Seth and Zhang, ICWSM 2008)
  - Recommend messages to a user based on the user's knowledge status
    - Why types of information the user wants
  - Context versus completeness
    - Friends in a same cluster provide contextual information
      - Strong ties: deeper understanding of a topic
    - Friends in a different cluster provide complete information
      - Weak ties: diverse opinions about a topic
  - Bayesian network based user model
- http://videolectures.net/icwsm08_seth_snba

Recommender on Social Web



Concluding Remarks

- Purpose of recommender systems
  - Assist users find what they want; help make correct decisions according to their needs, interests, goals
- Classic algorithms for recommendation
  - Feature based (Content based)
  - Collaborative filtering
  - Hybrid recommender (FB + CF)
- Some typical examples
  - Amazon, MovieLens, Last.fm, …
- Recommender on social web
  - Perkpipe
  - A social network based recommender

Open Discussions

- Facebook provides recommendations of friends, what kind of algorithms they might make use of?

Open Discussions

- If Facebook recommends people sharing similar interests with you, would you add them as friends? Why?

Open Discussions

- What other types of recommendations can be provided on Facebook and Twitter?

References

- Adomavicius, G. & Tuzhilin, A. 2005, "Toward the Next Generation of Recommender Systems: a Survey of the State-of-the-art and Possible Extensions", IEEE Trans. on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749.
- Tang, Ya (2008), "The Design and Study of Pedagogical Paper Recommendation". Published PHD Thesis. University of Saskatchewan, Saskatoon, SK.
- Seth, A. & Zhang, J. 2008, "A Social Network Based Approach to Personalized Recommendation of Participatory Media Content", In Proceedings of the International Conference on Weblogs and Social Media (ICWSM), Seattle, USA.