Recommender Systems

Recommender Systems

- Systems for recommending items (e.g. books, movies, CD's, web pages, newsgroup messages) to users based on examples of their preferences.
- Many on-line stores provide recommendations (e.g. Amazon, CDNow).
- Recommenders have been shown to substantially increase sales at on-line stores.
- There are two basic approaches to recommending:
  - Collaborative Filtering (a.k.a. social filtering)
  - Content-based

Collaborative Filtering

- Maintain a database of many users' ratings of a variety of items.
- For a given user, find other similar users whose ratings strongly correlate with the current user.
- Recommend items rated highly by these similar users, but not rated by the current user.
- Almost all existing commercial recommenders use this approach (e.g. Amazon).

User-based Collaborative Filtering

- Weight all users with respect to similarity with the active user.
- Select a subset of the users (*neighbors*) to use as predictors.
- Normalize ratings and compute a prediction from a weighted combination of the selected neighbors' ratings.
- Present items with highest predicted ratings as recommendations.

Similarity Weighting

- Typically use Pearson correlation coefficient between ratings for active user, *a*, and another user, *u*.

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a} \sigma_{r_u}}$$

r_a and r_u are the ratings vectors for the m items rated by **both** a and u

$r_{i,j}$ is user i 's rating for item j

Covariance and Standard Deviation

- Covariance:

$$\mathrm{covar}(r_a, r_u) = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

- Standard Deviation:

$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^{m} (r_{x,i} - \bar{r}_x)^2}{m}}$$

Neighbor Selection

- For a given active user, *a*, select correlated users to serve as source of predictions.
- Standard approach is to use the most similar *n* users, *u*, based on similarity weights, $w_{a,u}$
- Alternate approach is to include all users whose similarity weight is above a given threshold.

Rating Prediction

- Predict a rating, $p_{a,i}$, for each item i, for active user, a, by using the n selected neighbor users, $u \in \{1,2,\ldots n\}$.
- To account for users different ratings levels, base predictions on *differences* from a user's *average* rating.
- Weight users' ratings contribution by their similarity to the active user.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} c_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} c_{a,u}}$$

Problems with Collaborative Filtering

- **Cold Start**: There needs to be enough other users already in the system to find a match.
- **Sparsity**: If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.
- **First Rater**: Cannot recommend an item that has not been previously rated.
  - New items
  - Esoteric items
- **Popularity Bias**: Cannot recommend items to someone with unique tastes.
  - Tends to recommend popular items.

Content-Based Recommending

- Recommendations are based on information on the content of items rather than on other users' opinions.
- Uses a machine learning algorithm to induce a profile of the users preferences from examples based on a featural description of content.
- Some previous applications:
  - Newsweeder (Lang, 1995)
  - Syskill and Webert (Pazzani et al., 1996)

Advantages of Content-Based Approach

- No need for data on other users.
  - No cold-start or sparsity problems.
- Able to recommend to users with unique tastes.
- Able to recommend new and unpopular items
  - No first-rater problem.
- Can provide explanations of recommended items by listing content-features that caused an item to be recommended.

Disadvantages of Content-Based Method

- Requires content that can be encoded as meaningful features.
- Users' tastes must be represented as a learnable function of these content features.
- Unable to exploit quality judgments of other users.
  - Unless these are somehow included in the content features.