Foundations of Machine Learning

Module 3: Instance based Learning and Feature Reduction

Part D: Collaborative Filtering

Sudeshna Sarkar
IIT Kharagpur

Recommender Systems

- Item Prediction: predict a ranked list of items that a user is likely to buy or use. predict the rating score that a user is likely to give to an item that s/he has not seen or used before. E.g.,
  - rating on an unseen movie. In this case, the utility of item *s* to user *u* is the rating given to *s* by *u*.

- Rating Prediction: Predict whether someone will like a movie, book, webpage, etc.

The Recommendation Problem

- We have a set of users U and a set of items S to be recommended to the users.
- Let p be an utility function that measures the usefulness of item s ($\in S$) to user u ($\in U$), i.e.,
  - $p: U \times S \rightarrow R$, where R is a totally ordered set (e.g., non-negative integers or real numbers in a range)
- Objective
  - Learn p based on the past data
  - Use p to predict the utility value of each item s ($\in S$) to each user u ($\in U$)

Recommender Systems

- Content based :
  - recommend items similar to the ones the user preferred in the past
- Collaborative filtering:
  - Look at what similar users liked
  - Similar users = Similar likes and dislikes

Collaborative Filtering

- Present each user with a vector of ratings
- Two types:
  - Yes / No
  - Explicit Ratings
- Predict Rating by User-based Nearest Neighbour

Collaborative Filtering for Rating Prediction

- User-based Nearest Neighbour
  - Neighbour = similar users
  - Generate a prediction for an item i by analyzing ratings for i from users in u's neighbourhood

Neighborhood formation phase

- Let the record (or profile) of the target user be **u** (represented as a vector), and the record of another user be **v** (**v** ∉ *T*).
- The similarity between the target user, **u**, and a neighbor, **v**, can be calculated using the **Pearson's correlation coefficient**:

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i\in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i\in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i\in C}(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

Recommendation Phase

- Use the following formula to compute the rating prediction of item *i* for target user **u**
-

$$p(\mathbf{u}, i) = \bar{r}_{\mathbf{u}} + \frac{\sum_{\mathbf{v} \in V} sim(\mathbf{u}, \mathbf{v}) \times (r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v} \in V} \left| sim(\mathbf{u}, \mathbf{v}) \right|}$$

where V is the set of k similar users, $r_{\mathbf{v},i}$ is the rating of user **v** given to item *i*,

Issue with the user-based *k*NN CF

- The problem with the user-based formulation of collaborative filtering is the lack of scalability:
  - it requires the real-time comparison of the target user to all user records in order to generate predictions.
- A variation of this approach that remedies this problem is called **item-based CF**.

Item-based CF

- The item-based approach works by comparing items based on their pattern of ratings across users. The similarity of items *i* and *j* is computed as follows:

$$sim(i,j) = \frac{\sum_{\mathbf{u} \in U} (r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})}{\sqrt{\sum_{\mathbf{u} \in U} (r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2} \sqrt{\sum_{\mathbf{u} \in U} (r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})^2}}$$

Recommendation phase

- After computing the similarity between items we select a set of *k* most similar items to the target item and generate a predicted value of user **u**'s rating

$$p(\mathbf{u}, i) = \frac{\sum_{j \in J} r_{\mathbf{u},j} \times sim(i, j)}{\sum_{j \in J} sim(i, j)}$$

where *J* is the set of *k* similar items

Thank You