Collaborative Filtering and Recommender Systems

Definitions

Recommendation generation problem. Given a **set of users** and their (incomplete) **preferences** over **set of items**, find, for each user **new items for which they would have high preferences**.

Users. $C = \{c_1, \ldots, c_M\}$: a set of **users**.

Items. $S = \{s_1, \ldots, s_n\}$: a set of **items**.

Utility. $u : C \times S \longrightarrow \mathcal{R}$.

$u(c, s)$: *utility*, or *rating* or *preference* of user c for item s.

Typically, **utility function** is incomplete.

Utility function $u(c, s)$ can also be viewed as a **utility matrix** $u[.,.]$, where $u[i, j] = u(c_i, s_j)$. Matrix $u[]$ is **typically sparse**.

Problem. The main problem solved by collaborative filtering methods/recommender systems can be phrased in a number of ways:

- **User-based recommendations.** Given a user c find items $s'_1, \ldots s'_k$ (such that $u(c, s'_i)$ is undefined), for which c is predicted to have highest utility.
- **Individual ratings.** Given a user c and an item s, predict $u(c, s)$.
- **Item-based recommendations.** Given an item s, find users $c'_1, \ldots, c'_k$, (s.t., c'_i, s is undefined) for which $u(s, c'_i)$ is predicted to be the highest.

Recommender Systems

Recommender System: a system, which given C, S and a *partial* utility function u, solves one or more of the problems of recommendation generation.

Content-based recommendation systems: recommend items *similar* to the ones preferred by the user in the past.

Collaborative recommendation systems: recommend items that *other users with similar preferences* find to be of high utilitiy.

Hybrid recommendation systems: combine content-based and collaborative recommendations.

Content-based recommendation systems. Content-based recommendation systems use methodology similar to that used in Information Retrieval. These methods will be covered separately.

Collaborative Filtering in Recommender Systems

Idea. Given $c \in C$ and $s \in S$, estimate $u(c, s)$ based on the **known utilities** $u(c', s)$ for item s for users $C' = \{c'\} \subseteq C$.

Types. There are two types of collaborative filtering approaches:

1. **Memory-based methods.** These methods use different **heuristics** to construct utility predictions.
2. **Model-based methods.** These methods use the utility function u to **learn a model** of a specific type. The model is then used to generate predictions.

Memory-based Collaborative Filtering.

Aggregation of the utilities. Memory-based collaborative filtering methods aggregrate the known utilities $u(c_i, s)$ for item s to predict the utility (rating) of s for user c (user-based aggregation):

$$u(c, s) = \mathsf{aggregate}_{c_i \in C} u(c_i, s).$$

Similar aggregation exists for items:

$$u(c, s) = \mathsf{aggregate}_{s_i \in S} u(c, s_i).$$

Notation. Let $s \in S$ be some item. As C^s we denote the set

$$C^s = \{c \in C | u(s, c) \;\; \mathbf{is\ \ defined}\},$$

i.e., the set of all users which have an existing rating (utility) for item s.

Similarly, for a user $c \in C$,

$$S^c = \{s \in S | u(s,c) \text{ is defined}\}.$$

Notation. Let $c \in C$ and let $S = \{s_1, \ldots, s_n\}$. As $u[c]$ we denote the (sparse) vector:

$$u[c] = (u(c,s_1), u(c,s_2), \ldots, u(c,s_n)).$$

Note. In the computations below, whenever we see a value of $u(c,s)$ that is not defined in the dataset, we assume that its value is 0 in all computations.

Mean utility. The most simple collaborative filtering predictor is the mean utility.

$$u(c,s) = \frac{1}{|C^s|} \sum_{c_i \in C^s} u(c_i, s).$$

This is a *very simplistic* prediction, as it ignores various information about current user's preferences that is available to us.

Weighted sum. This predictor is one of the most commonly used. It assumes existence of a **similarity function** $sim(.,.)$ which reports the proximity between utility vectors for two users.

$$u(c,s) = k \cdot \sum_{c' \neq c} sim(u[c], u[c']) \cdot u(c', s),$$

where k, the *normalization factor* is typically set to

$$k = \frac{1}{\sum_{c' \neq c} |sim(u[c], u[c'])|}$$

$$u(c,s) = \frac{1}{\sum_{c' \neq c} |sim(u[c], u[c'])|} \cdot \sum_{c' \neq c} sim(u[c], u[c']) \cdot u(c', s),$$

Weighted sum predictor has one weakness:

- **insensitivity** to the fact that different users employ the rating/utility scale **differently** when reporting their preferences.

Adjusted weighted sum. In predicting the utilities for a specific user, we take into account, the user's approach to rating the items. First, we compute the user's average rating $\hat{u_c}$:

$$\bar{u_c} = \frac{1}{|S^c|} \sum_{s' \in S^c} u(c, s').$$

We then predict $u(c,s)$ for some item $s \in S$ as follows:

$$u(c,s) = \bar{u_c} + k \cdot \sum_{c' \neq c} sim(u[c], u[c']) \cdot (u(c', s) - \bar{u_{c'}}).$$

Here, k is the same normalizing factor as above.

N Nearest Neighbors predictors

All predictors discussed above can be updated **to include only N nearest neighbors of the user c in the comparison**.

Let $C'_c = \{c' \in C | rank(sim(u[c], u[c'])) \leq N\}$ be the set of N **nearest neighbors to user** c using similarity function $sim(.,.)$.

Average Nnn ranking.

$$u(s, c) = \frac{1}{N} \sum_{c' \in C'_c} u(c', s).$$

Weighted Nnn sum.

$$u(c, s) = k \cdot \sum_{c' \in C'_c} sim(u[c], u[c']) \cdot u(c', s).$$

$$k = \frac{1}{\sum_{c'C'_c} |sim(c, c')|}.$$

Adjusted weighted Nnn sum.

$$u(c, s) = \bar{u_c} + k \cdot \sum_{c' \in C} sim(u[c], u[c']) \cdot (u(c', s) - \bar{u_{c'}}).$$

Similarity Measures

Two similarity measures are typically used in collaborative filtering.

Pearson Correlation.

$$sim(u[c], u[c']) = \frac{\sum_{i=1}^{n}(u(c, s_i) - \bar{u_c}) \cdot (u(c', s_i) - \bar{u_{c'}})}{\sqrt{\sum_{i=1}^{n}(u(c, s_i) - \bar{u_c})^2 \cdot \sum_{i=1}^{n}(u(c', s_i) - \bar{u_{c'}})^2}}.$$

This measure reflects *statistical correlation* between the two (sparse) vectors of data.

Cosine similarity.

$$sim(u[c], u[c']) = cos(u[c], u[c']) = \frac{u[c] \cdot u[c']}{||u[c]|| \cdot ||u[c']||} = \frac{\sum_{i=1}^{n} u(c, s_i) \cdot u(c', s_i)}{\sqrt{\sum_{i=1}^{n} u(c, s_i)^2 \cdot \sum_{i=1}^{n} u(c', s_i)^2}}.$$

Cosine similarity measures the *colinearity* of the two vectors (it is 1 if the vectors are colinear, and 0 if they are orthogonal).

References

[1] G. Adomavicius, A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, No. 6, June 2005.