Acknowledgement

- Slides are based on previous classes by:
  - Hanghang Tong (ASU)
  - Bing Liu (UIC)

Road Map

- **Introduction**
- Content-based recommendation
- Collaborative filtering based recommendation
  - K-nearest neighbor
  - Association rules
  - Matrix factorization

Introduction

- Recommender systems are widely used on the Web for recommending products and services to users.
- Most e-commerce sites have such systems.
  - and many more (bio-informatics, scholarly domain)
- These systems serve two important functions.
  - They help users deal with the information overload by giving them recommendations of products, etc.
  - They help businesses make more profits, i.e., selling more products.

E.g., movie recommendation

- The most common scenario is the following:
  - A set of users has initially rated some subset of movies (e.g., on the scale of 1 to 5) that they have already seen.
  - These ratings serve as the input. The recommendation system uses these known ratings to predict the unknown ratings that each user would give to those not rated movies by him/her.
  - Recommendations of movies are then made to each user based on the predicted ratings.

Different variations

- In some applications, there is no rating information (one-class rec. sys) while in some others there are also additional attributes
  - about each user (e.g., age, gender, income, marital status, etc), and/or
  - about each movie (e.g., title, genre, director, leading actors or actresses, etc).
- When no rating information, the system will not predict ratings but predict the likelihood that a user will enjoy watching a movie.

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 ($\in U$)

As Prediction

- Rating prediction, i.e., 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*.
- Item prediction, i.e., predict a ranked list of items that a user is likely to buy or use.

Two basic approaches

- Content-based recommendations:
  - The user will be recommended items similar to the items the user preferred in the past;
- Collaborative filtering (or collaborative recommendations):
  - The user will be recommended items that people with similar tastes and preferences liked in the past.
- **Hybrids**: Combine collaborative and content-based methods.

Road Map

- Introduction
- **Content-based recommendation**
- Collaborative filtering based recommendation
  - K-nearest neighbor
  - Association rules
  - Matrix factorization

Content-Based Recommendation

- Perform item recommendations by predicting the utility of items for a particular user based on how "similar" the items are to those that he/she liked in the past. E.g.,
  - In a movie recommendation application, a movie may be represented by such features as specific actors, director, genre, subject matter, etc.
  - The user's interest or preference is also represented by the same set of features, called the **user profile**.

Content-based recommendation (contd)

- Recommendations are made by comparing the user profile with candidate items expressed in the same set of features.
- The top-*k* best matched or most similar items are recommended to the user.

- The simplest approach to content-based recommendation is to compute the similarity of the user profile with each item.

Road Map

- Introduction
- Content-based recommendation
- Collaborative filtering based recommendations
  - **K-nearest neighbor**
  - Association rules
  - Matrix factorization

Collaborative filtering

- Collaborative filtering (CF) is perhaps the most studied and also the most widely-used recommendation approach in practice.
  - *k*-nearest neighbor,
  - association rules based prediction, and
  - matrix factorization
- Key characteristic of CF: it predicts the utility of items for a user based on the items previously rated by other like-minded users.

k-nearest neighbor

- *k*NN (which is also called the *memory-based approach*) utilizes the entire user-item database to generate predictions directly, i.e., there is no model building.
- This approach includes both
  - User-based methods
  - Item-based methods

User-based *k*NN CF

- A user-based *k*NN collaborative filtering method consists of two primary phases:
  - the neighborhood formation phase and
  - the recommendation phase.
- There are many specific methods for both. Here we only introduce one for each phase.

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**

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

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

Road Map

- Introduction
- Content-based recommendation
- Collaborative filtering based recommendation
  - K-nearest neighbor
  - **Association rules**
  - Matrix factorization

Association rule-based CF

- Association rules obviously can be used for recommendation.
- Each transaction for association rule mining is the set of items bought by a particular user.
- We can find item association rules, e.g.,

  buy_X, buy_Y -> buy_Z
- Rank items based on measures such as confidence, etc.

Road Map

- Introduction
- Content-based recommendation
- Collaborative filtering based recommendation
  - K-nearest neighbor
  - Association rules
  - **Matrix factorization**

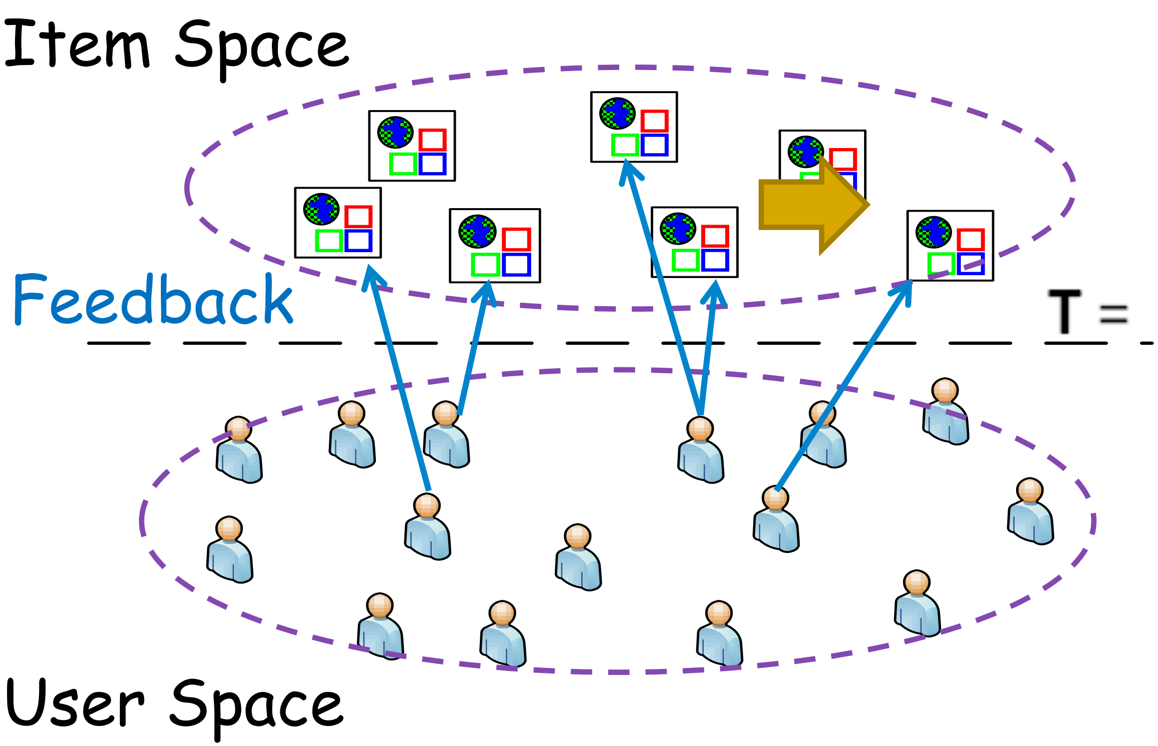
Matrix factorization

- The idea of **matrix factorization** is to decompose a matrix ***M*** into the product of several factor matrices, i.e.,

$$\boldsymbol{M} = \boldsymbol{F}_1 \boldsymbol{F}_2 \ldots \boldsymbol{F}_n$$

where *n* can be any number, but it is usually 2 or 3.

Matrix-factorization CF



items

$$\mathbf{T} = \begin{bmatrix} / & 1 & 1 & ? & 1 \\ 0.5 & / & 1 & ? & ? \\ ? & ? & / & ? & ? \\ ? & ? & ? & / & ? \\ 0.5 & ? & ? & 1 & / \end{bmatrix}$$

users

$$\min_{\mathbf{F},\mathbf{G}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i,j) - \mathbf{F}(i,:)\mathbf{G}(j,:)')^2 + \lambda\|\mathbf{F}\|_{fro}^2 + \lambda\|\mathbf{G}\|_{fro}^2$$

Matrix-factorization CF

$$\min_{\mathbf{F},\mathbf{G}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i,j) - \mathbf{F}(i,:)\mathbf{G}(j,:)')^2 + \lambda||\mathbf{F}||_{fro}^2 + \lambda||\mathbf{G}||_{fro}^2$$

items

users

$$\mathbf{T} = \begin{bmatrix} / & 1 & 1 & ? & 1 \\ 0.5 & / & 1 & ? & ? \\ ? & ? & / & ? & ? \\ ? & ? & ? & / & ? \\ 0.5 & ? & ? & 1 & / \end{bmatrix}$$

$$\mathbf{F} = \begin{array}{c} \begin{array}{cc} \text{Delivering time} & \text{Product price} \end{array} \\ \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}}_{\text{factors}} \end{array} \text{items}$$

$$\mathbf{G} = \begin{array}{c} \begin{array}{cc} \text{Delivering time} & \text{Product price} \end{array} \\ \underbrace{\begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}}_{\text{factors}} \end{array} \begin{array}{l} \text{Alice} \\ \text{Bob} \\ \text{Carol} \\ \text{David} \\ \text{Elva} \end{array}$$