40.016: The Analytics Edge
Week 11 Lecture 2

RECOMMENDATION SYSTEMS (PART 2)

Term 5, 2022

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

Outline

Recommendation systems

- Personalize the user experience for online applications
- Leverage data on items and customers (e.g., likes, purchase history)
- A key challenge: they must
  - be fast
  - be accurate
  - work with large/small datasets
  - some variables may have large variance
- Common underlying analytics:
  - clustering (Monday)
  - collaborative and content filtering (Today)

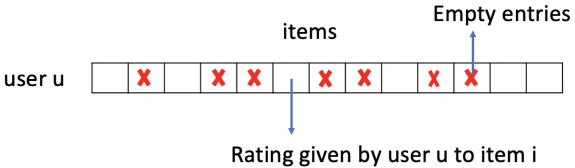
Recommendation systems (cont'd)

There are three main types of recommendation systems:

- Collaborative filtering
- Content filtering
- Hybrid recommendation systems

Collaborative filtering

- Recommendations are based on attributes of users.
- Each user is represented by a vector of items where the i-th entry gives the customer's rating of the i-th item.
- This vector will typically have many empty entries (only a small fraction of the items is ranked or purchased).



- The data of ratings of users for items are used to predict missing ratings or create top-N recommendation lists for an active user.

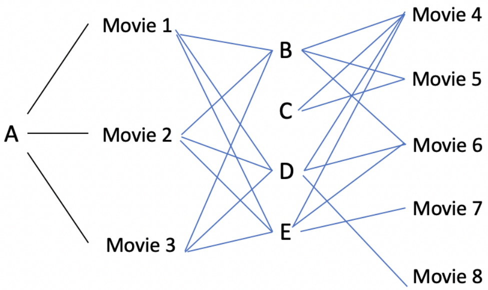
Collaborative filtering (cont'd)

Collaborative filtering systems have many forms, but many common systems can be reduced to two steps:

Step 1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).

Step 2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.

Example



- Person A likes movies 1, 2 & 3
- Suppose persons B, C, D & E also likes movies 1, 2 & 3
- All of them (B, C, D & E) also like movie 4, followed by movie 6.
- Thus we can recommend them to A in order.

Advantages and Disadvantages

Advantages:

- Domain free (we don't need domain knowledge because the embeddings are automatically learned)
- Helping users discover new interests (the system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item)

Disadvantages:

- Note that collaborative filtering will suffer from a cold start problem, since it will be unable to address new items or new users

Content filtering

- Recommendations are made based on attributes of items.
- Each item is represented by a set of attributes (e.g., genre of movie, keywords, or webpage).
- For example, Pandora uses the attributes of a song (e.g., style and artist) to seed the station with other songs with similar attributes.

Content filtering (cont'd)

Advantages and Disadvantages

Advantages:

- The model doesn't need any data about other users, since the recommendations are specific to this user.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

Disadvantages:

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge.
- The model can only make recommendations based on existing interests of the user.

Hybrid recommendation systems

- This is a combination of both collaborative and content filtering.
- Netflix, for example, makes recommendations by comparing the watching and searching habits of similar users (collaborative filtering) as well as recommending movies that share similar attributes (content filtering).

Some discussions

- Content filtering often works better than collaborative filtering if the user has not rated or purchased many items.
- However, if a user has rated many items, it is hard for content filtering to make recommendations, since there might be many items with similar recommendations.
- Traditionally, the cold start problem of collaborative filtering is tackled by resorting to an additional interview process to establish the user (item) profile before making any recommendations – content filtering

Outline

Data

Matrix containing ratings:

$$
\begin{array}{cc}
 & p \text{ items} \\
n \text{ users} & \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{np} \end{pmatrix} \\
\text{Active user} & \boxed{\; ? \qquad r_{u2} \qquad\qquad r_{up} \;}
\end{array}
$$

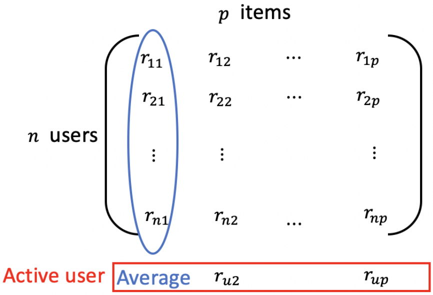
Let r_{ui} be the rating of user u for item i.

Baseline model

A simple baseline model is to predict the average rating based on the items' average popularity:

$$b_{ui} = \bar{r}_i$$

where b_{ui} is the baseline prediction for user u and item i, and $\bar{r}_i$ is the average rating for item i across all users who rated it.



User-based collaborative filtering

This model is based on

- Identifying users whose ratings are similar to those of the active user
- Using their ratings on other items to predict what the active (current) user will like

Challenge:

- How to measure similarity?
- How to predict?

Pearson correlation coefficient

To measure the similarity between users u and v, we can use the Pearson correlation coefficient:

$$S_{uv} = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$

where S_{uv} is the similarity between users u and v, I_u and I_v the items rated by users u and v, $\bar{r}_u$ and $\bar{r}_v$ the average ratings of users u and v.

- Recall the correlation of random variables x and y is

$$r_{xy} = \frac{\mathrm{Cov}(x, y)}{\sigma_x \sigma_y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \cdot \sqrt{\sum_i (y_i - \bar{y})^2}}$$

- This similarity metric was adopted by Netflix.

Cosine similarity

An alternative similarity measure is the cosine similarity:

$$S_{uv} = \frac{\sum_{i \in I_u \cap I_v} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} r_{vi}^2}}$$

- Recall given two sample feature vectors $x = (x_1, \cdots, x_n)$, $y = (y_1, \cdots, y_n)$, the cosine similarity is defined as

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2}}$$

Pearson correlation v.s. cosine similarity

- The Pearson correlation coefficient ranges from -1 to 1. 1 means the two random variables are perfectly positively correlated, -1 means perfectly negatively correlated, 0 means not correlated.
- The cosine similarity ranges from -1 to 1. 1 means the two samples are the most similar and -1 means the two samples are the least similar.
- The two quantities represent two different physical entities.
  - The cosine similarity computes the similarity between two samples
  - whereas the Pearson correlation coefficient computes the correlation between two jointly distributed random variables.

User-based collaborative filtering: Prediction

To predict the rating, the simplest method is to choose a set of neighbours of user u, denoted by N_u (say the k nearest neighbours with a certain level of similarity),

$$p_{ui} = \frac{\sum_{v \in N_u} r_{vi}}{|N_u|},$$

where p_{ui} is the predicted rating for user u and item i.

- $|N_u|$: number of neighbours
- While all users could be used in the set N_u, it helps strict it to a smaller number of neighbours (in the range of 20 to 200, typically).

Prediction alternative 1

We can also use the observation that some users are more similar to u using the similarity metric:

$$p_{ui} = \frac{\sum_{v \in N_u} S_{uv} r_{vi}}{\sum_{v \in N_u} |S_{uv}|}$$

- Weighting each rating r_{ui} by the similarity S_{ui}
- Note that not all users in N_u have the same similarities

Prediction alternative 2

- The performance of recommendation systems can be improved by normalizing the ratings so as to compensate for users differences on rating scales.
- For example, there could be bias caused by users who consistently rate higher than (or lower than) other users.

Let $\hat{r}_{ui} = r_{ui} - \bar{r}_u$: this centers the scores. After applying collaborative filtering and normalizing back to the original scale, we get:

$$p_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} S_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in N_u} |S_{uv}|}$$

- Other transformations include taking the rating variance into account.

Assessing performance

To measure the quality of predicted ratings, we can use:

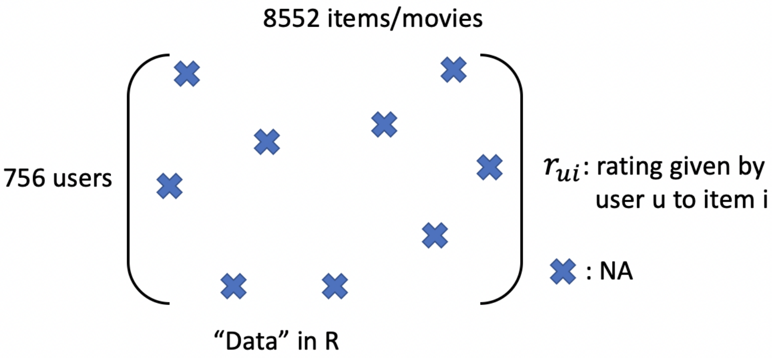
$$\mathrm{RMSE} = \sqrt{\sum_i \frac{(p_{ui} - r_{ui})^2}{n}}$$

where p_{ui} is the predicted rating for user u and item i, r_{ui} the real rating, and n the total number of ratings that are predicted.

Outline

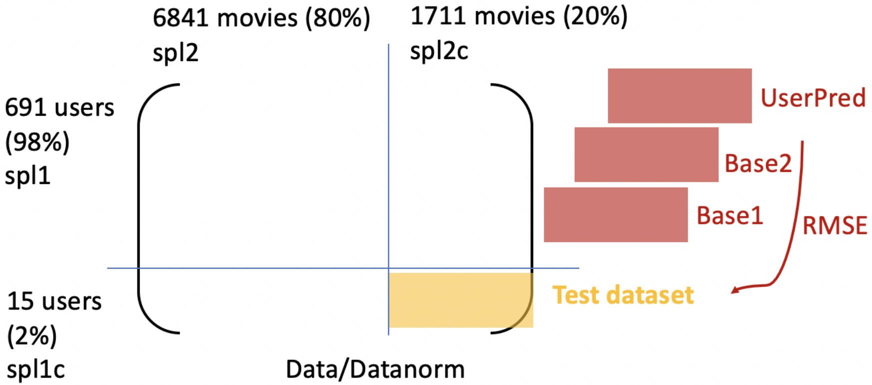
R implementation

`ratings` dataset



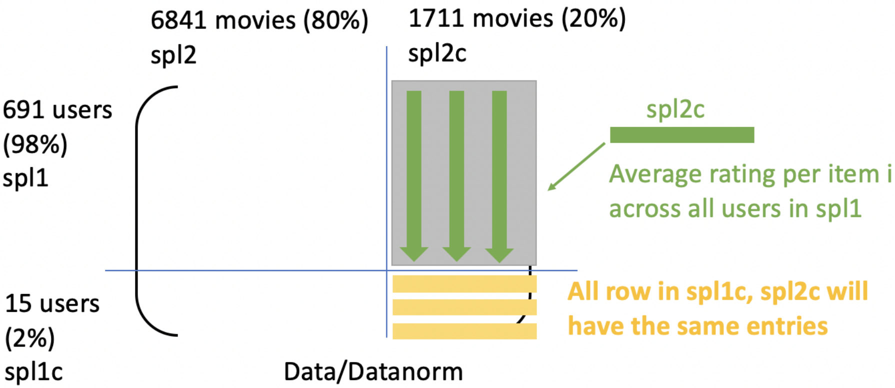
R implementation (cont'd)

Data preparation



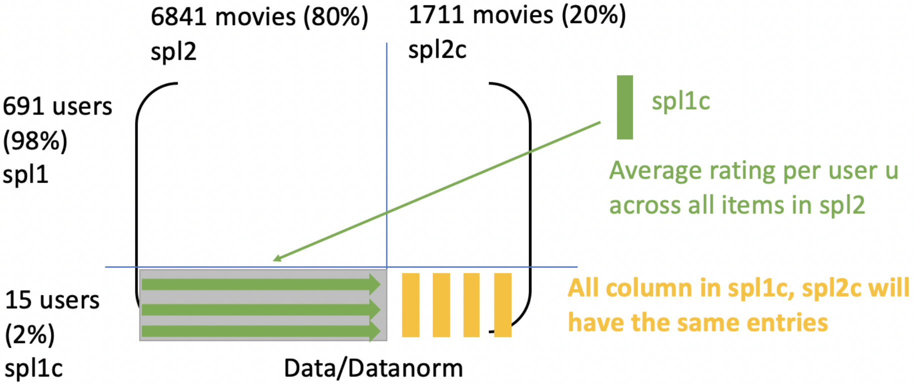
R implementation (cont'd)

Baseline model 1



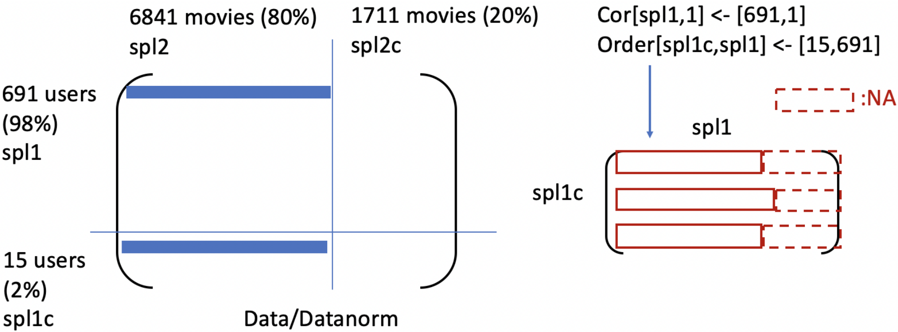
R implementation (cont'd)

Baseline model 2



R implementation (cont'd)

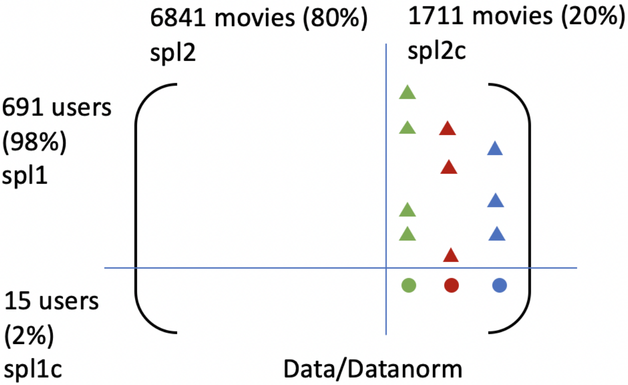
User-based model



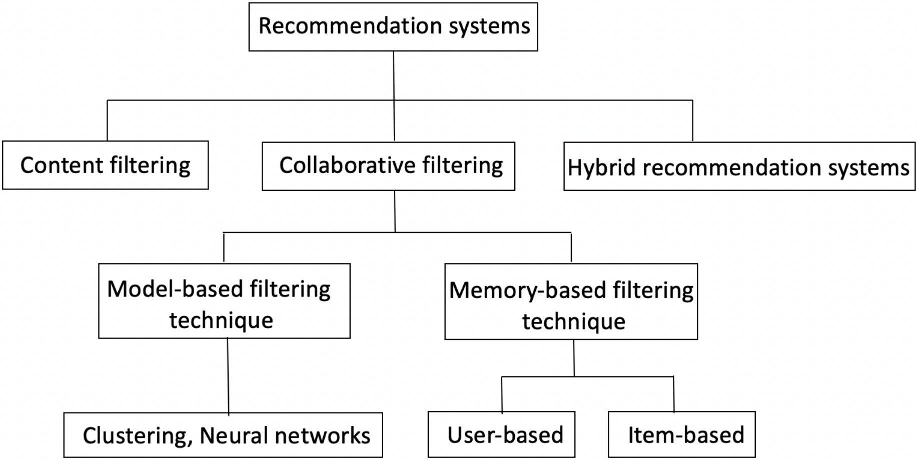
- We sort the users in spl1 by decreasing correlations
- The NA accounts for users who have no common ratings of movies with the user

R implementation (cont'd)

User-based model



Summary


Recommendation systems
Content filtering
Collaborative filtering
Hybrid recommendation systems
Model-based filtering technique
Memory-based filtering technique
Clustering, Neural networks
User-based
Item-based

References

- Teaching notes.