IT492: Recommendation Systems

Lecture - 03, 04

Collaborative Filtering

Arpit Rana

24th Jan 2022

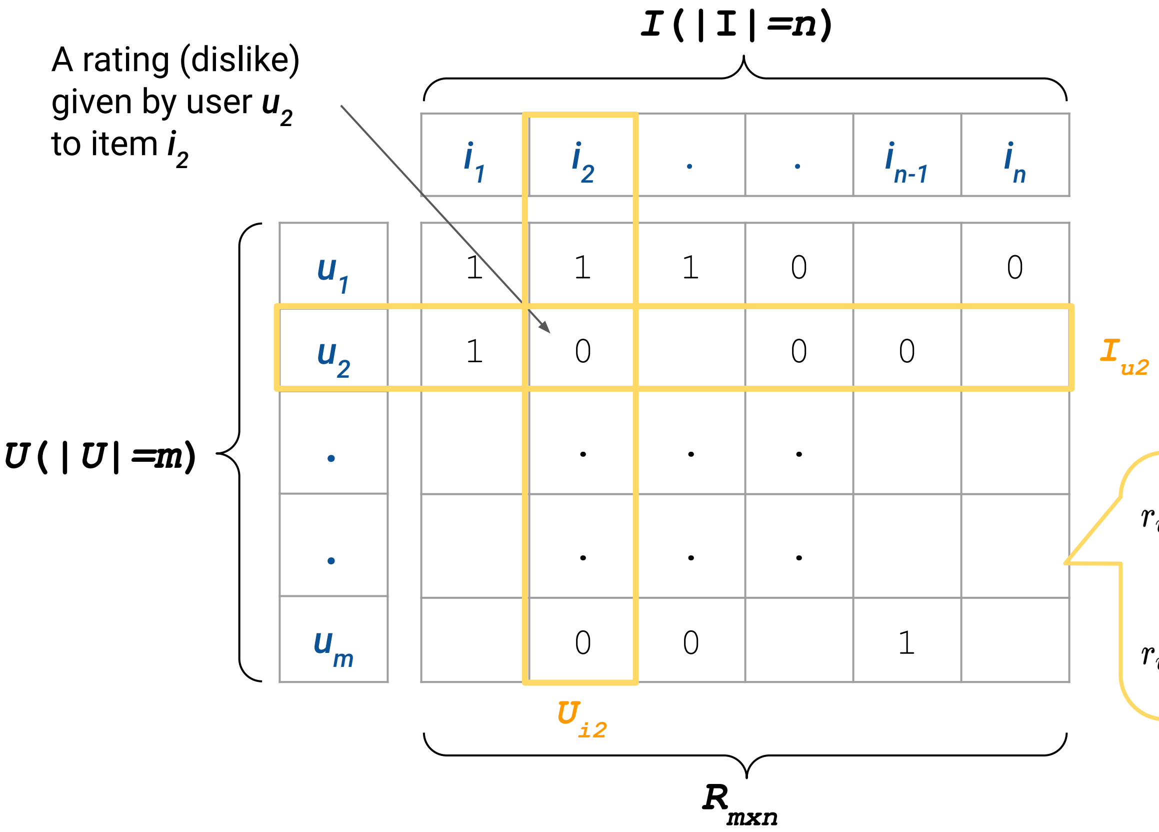
Collaborative Filtering

Collaborative Filtering Methods recommend items that -

- either users with *similar* tastes liked in the past,
- OR, according to the other users, are *similar* to items that are liked by the ***active/target user***.

- ***Similarity*** between users or items relies on users' feedback to the items that they have consumed, *e.g. ratings*

A (Binary) Rating Matrix



$$r_{ui} = \begin{cases} Observed & r_{ui} = 1/0 \\ Unobserved & r_{ui} = \phi \end{cases}$$

$$r_{ui} = \begin{cases} Like & r_{ui} = 1 \\ Dislike & r_{ui} = 0 \end{cases}$$

Recommendation as Rating Prediction

***Predict** unobserved **(missing i.e.** $r_{ui} = \phi$**) from observed ratings**.*

Two types of methods commonly used:

- ***Memory-based (Neighborhood-based) methods:*** are usually *heuristic approaches* that leverage either *inter-user* or *inter-item* *correlations*.

- ***Model-based methods:*** include machine learning and data mining methods for rating prediction. *e.g. latent factor models*

heuristic : enabling someone to discover or learn something for themselves.
"a ‘hands-on’ or interactive heuristic approach to learning"

User-user Collaborative Filtering

It predicts the missing ratings based on the ratings given by ***peers*** to that item.

	i_1	i_2	.	.	i_{n-1}	i_n
u_1	1	1	1	0		0
u_2	1	0	$\boxed{?}$	0	0	
.		.	.	.		
.		.	.	.		
u_m		0	0		1	

0.71

Cosine Similarity

0.50

User-user Collaborative Filtering

- The rating r_{ui} can be estimated as the average rating given to i by neighbours of user u (i.e. N_{ui})

$$\hat{r}_{ui} = \frac{1}{|N_{ui}|} \sum_{v \in N_{ui}} r_{vi}$$

- Not suitable, why?

User-user Collaborative Filtering

- An average of peers' ratings does not take into account the level of similarity with neighbours.
- One solution is to weigh the contribution of each neighbor by its similarity to $\boldsymbol{u}$.

$$\hat{\boldsymbol{r}}_{ui} = \frac{\sum_{v \in N_{ui}} \boldsymbol{w}_{uv} \boldsymbol{.}\, \boldsymbol{r}_{vi}}{\sum_{v \in N_{ui}} |\boldsymbol{w}_{uv}|}$$

- Why normalizing the weighted sum? Why taking | . | ?
- Still not perfect, why?

User-user Collaborative Filtering

- The weighted average does not consider that users may use different rating values to quantify the same level of appreciation for an item.

- Convert neighbours' ratings r_{vi} to normalized ones $h(r_{vi})$.

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{v \in N_{ui}} w_{uv} . h(r_{vi})}{\sum_{v \in N_{ui}} |w_{uv}|}\right)$$

Normalization techniques are covered later.

User-user Collaborative Filtering

Rating Prediction as a User-Based Classification Problem

	i_1	i_2	.	.	i_{n-1}	i_n
u_1	1	1	1	0		0
u_2	1	0	?	0	0	
.		.	.	.		
.		.	.	.		
u_m		0	0		1	

0.71

Cosine Similarity

0.50

User-user Collaborative Filtering

Rating Prediction as a User-Based Classification Problem

- The rating r □ S (rating scale, e.g. 1 -- 5) for which sum of the similarity weights of neighbors that have given rating r to the item i is maximum.

$$\hat{r}_{ui} = \arg\max_{r \in S} \sum_{v \in N_{ui}} \delta(r_{vi} = r).\, w_{uv}$$

Here δ returns 1 if $r_{vi} = r$, 0 otherwise.

- The normalized version, here S' is the normalized rating scale -

$$\hat{r}_{ui} = h^{-1}\left(\arg\max_{r \in S'} \sum_{v \in N_{ui}} \delta(h(r_{vi}) = r).\, w_{uv}\right)$$

User-user Collaborative Filtering: Regression vs. Classification

	Rating Prediction as a User-Based Regression	**Rating Prediction as a User-Based Classification**
Rating Scale is Continues	✔	
Rating Scale Size is Large	✔	
Ratings are not ordinal		✔
Neighbour size is large*	✔	

* Why?

Item-item Collaborative Filtering

It predicts the missing ratings based on the ratings of items that have *similar rating patterns* to that item.

0.35

	i_1	i_2	.	.	i_{n-1}	i_n
u_1	1	1	1	0		0
u_2	1	0	**?**	0	0	
.		.	.	.		
.		.	.	.		
u_m		0	0		1	

0.78

Rating Prediction Formulae - ???

Cosine Similarity

User-user vs. Item-item Collaborative Filtering

There are five criteria that need to be considered while choosing one of the two types of techniques:

- **Accuracy:** a small number of high-confidence neighbors is by far preferable to a large number of loosely coupled neighbors.
    - Number of users >> number of items ⇒ item-item collaborative filtering
        - Amazon recommendation
    - Number of users << number of items ⇒ user-user collaborative filtering
        - Research article recommendation

User-user vs. Item-item Collaborative Filtering

There are five criteria that need to be considered while choosing one of the two types of techniques:

- **Efficiency:** depends upon the ratio of number users and the number of items..
    - Number of users >> number of items ⇒ item-item collaborative filtering (less computation and memory will be required)
    - Number of users << number of items ⇒ user-user collaborative filtering

User-user vs. Item-item Collaborative Filtering

There are five criteria that need to be considered while choosing one of the two types of techniques:

- **Stability:** depends on the frequency and amount of change in the users and items of the system.
    - If the list of items constantly changing, user based approach is preferable, e.g. online article recommendation
    - If the list of users are updating and the list of items is fairly static, item based approach is preferable, e.g. online shopping applications

User-user vs. Item-item Collaborative Filtering

There are five criteria that need to be considered while choosing one of the two types of techniques:

- **Justifiability:** depends on the explainability of the recommendation approach.
    - The list of neighboring items is more justifiable than the list of users (most users are unknown to the active user)

User-user vs. Item-item Collaborative Filtering

There are five criteria that need to be considered while choosing one of the two types of techniques:

- **Serendipity:** depends on the ability of the recommendation approach to offer surprising recommendations.
  - Item based approach relies on items similar to active users items; thus, less prone to provide more surprising recommendations.
  - User based approach relies on peers' opinio with similar tastes. In case, a user likes one item different from her usual taste, may be recommended to her peers.

Components of Neighborhood-based Methods

While implementing neighborhood-based models, the following aspects need to be taken into account -

- **the *normalization* of ratings,**
- the computation of the *similarity* weights, and
- the *selection* of neighbors

Rating Normalization

Users may use different rating values to quantify the same level of appreciation for an item despite giving the explicit definition (e.g., 1=“strongly disagree”, 2=“disagree”, 3=“neutral”, etc.).

There are two popular schemes for rating normalization -

- **Mean-centering**
- **Z-score**

Mean-centering Normalization

Mean-centering

- It determines whether a user rating is +ve or --ve with respect to her mean rating

$$h(r_{ui}) = r_{ui} - \bar{r}_u$$

- Using this approach the user-based prediction of a rating r_{vi} is obtained as

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \left(r_{vi} - \bar{r}_v \right)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}.$$

Z-score Normalization

Z-score

- It considers the spread in an individual's rating scale.

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u}$$

- Using this approach the user-based prediction of a rating r_{vi} is obtained as

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \, (r_{vi} - \bar{r}_v)/\sigma_v}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}.$$

Mean-centering vs. Z-score

The following cases need to be considered before choosing a normalization scheme.

- Sparse ratings
- Cold-start user/ item
- User rates with only the highest value

if the rating scale has a wide range of discrete values or if the scale is continuous, Z-score is useful.

Z-score is more sensitive scheme than mean-centering.

Components of Neighborhood-based Methods

While implementing neighborhood-based models, the following aspects need to be taken into account -

- the *normalization* of ratings,
- **the computation of the *similarity* weights**, and
- the *selection* of neighbors

Computing Similarity Weights

The computation of the similarity weights is one of the most critical aspects of building a neighborhood-based recommender system -

- They allow to **select trusted neighbors** whose ratings are used in the prediction, and
- They provide the **means to give more or less importance** to these neighbors in the prediction.

Cosine Vector Similarity

- In general, cosine vector similarity between two vectors: $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$

$$\cos(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^\top \mathbf{x}_b}{||\mathbf{x}_a||||\mathbf{x}_b||}.$$

- In item recommendation, cosine similarity between two users can be defined as

$$CV(u, v) = \cos(\mathbf{x}_u, \mathbf{x}_v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} r_{ui}\, r_{vi}}{\sqrt{\sum\limits_{i \in \mathcal{I}_u} r_{ui}^2 \sum\limits_{j \in \mathcal{I}_v} r_{vj}^2}},$$

Pearson Correlation

Cosine similarity does not consider the differences in the mean and variance of the ratings made by users u and v.

- Pearson correlation (PC) takes mean and variance into account

$$\mathrm{PC}(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}}.$$

- Is it different from computing the cosine similarity on the Z-score normalized ratings? yes

Adjusted Cosine

The differences in the rating scales of individual users are often more pronounced than the differences in ratings given to individual items.

- Pearson correlation between items can be adjusted (defined as Adjusted Cosine) to consider mean of user ratings -

$$AC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}}.$$

- It has been found that Adjusted Cosine outperforms Pearson Correlation in item-based settings.

Considering the Significance of Weights

- The rating data is frequently sparse in comparison to the number of users and items of a system,
- In general, similarity weights are computed using only a few ratings given to common items or made by the same users.
- If these few ratings are equal, then the users will be considered as “fully similar” and likely to affect each other’s recommendations.

Considering the Significance of Weights

- *Significance weighting* reduces the magnitude of a similarity weight when this weight is computed using only a few ratings.
- One way to penalize such weights is -

$$w'_{uv} = \frac{\min\{|\mathcal{I}_{uv}|, \gamma\}}{\gamma} \times w_{uv}.$$

Here, γ (> 0) is a parameter that can be tuned through cross-validation. Typical value of γ is 50.

Considering the Significance of Weights

- *Significance weighting* reduces the magnitude of a similarity weight when this weight is computed using only a few ratings.
- Another way to penalize similarity weights is -

$$w'_{uv} = \frac{|\mathcal{I}_{uv}|}{|\mathcal{I}_{uv}| + \beta} \times w_{uv},$$

Here, $\beta (> 0)$ is a parameter that can be tuned through cross-validation. Typical value of β is 100.

Components of Neighborhood-based Methods

While implementing neighborhood-based models, the following aspects need to be taken into account -

- the *normalization* of ratings,
- the computation of the *similarity* weights, and
- **the *selection* of neighbors**

Selecting Neighbours

The number of nearest-neighbors to select and the criteria used for this selection affects the quality of the recommender system.

The selection of the neighbors is normally done in two steps:

- **Pre-filtering**, only the most likely candidates are kept, and
- **Per-prediction filtering**, chooses the best candidates for a prediction.

Selecting Neighbours

Pre-filtering of Neighbours

- Storing similarities between each pair of users or items increases memory requirements and the computation time.
- There are several ways of limiting the number of candidate neighbors to consider in the predictions.
    - **Top-*N* filtering**, a high value of ***N*** will require excessive memory while a lower ***N*** may reduce the coverage of the recommender.
    - **Threshold filtering**, setting a threshold on the similarity is more flexible though difficult to determine.
    - **Negative filtering**, negative correlations can be filtered out.

Selecting Neighbours

Per-prediction filtering of Neighbours

- Once a list of candidate neighbors has been computed for each user or item, the prediction of new ratings is normally made with the ***k***-nearest-neighbors.
- ***k*** neighbors whose similarity weight has the greatest magnitude.
- The choice of ***k*** can also have a significant impact on the accuracy and performance of the system.

Exercise

Apply what you have learned so far on the following rating matrix and discuss your observations on the Classroom Stream -

	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
John	5	1		2	2
Lucy	1	5	2	5	5
Eric	2	?	3	5	4
Diane	4	3	5	3	

Next lecture -

Collaborative Methods (Model-based)