Neighbourhood-based recommender system example from the lecture (with user-based similarity)

October 24, 2023

1 Prediction task

Table 1 shows ratings of six movies (m_i) by four users (u_i). The movies are m_1=Gladiator, m_2=Godfather, m_3=Ben-Hur, m_4=Goodfellas, m_5=Scarface, m_6=Spartacus. The ratings scale is from 1 (didn't like at all) to 5 (loved it). Missing values (–) mean that the user hasn't rated (or watched) the movie. The task is to predict the missing ratings and decide whether to recommend a certain movie to a certain user (if the user would like it more than average, i.e., the rating would be more than the user's mean rating).

2 User-based approach

The idea is to use similar users' ratings to predict the missing rating. One commonly used measure of similarity is a modified version of Pearson correlation coefficient. The difference to normal Pearson correlation coefficient

Table 1: Ratings of 6 movies by 4 users.

	m_1	m_2	m_3	m_4	m_5	m_6
u_1	–	1	2	2	3	–
u_2	3	1	1	2	4	3
u_3	4	2	3	3	–	5
u_4	2	5	4	–	1	2

is that only co-rated items are included. The required mean values can be calculated over all rated items by the user or only over co-rated items.
Pearson correlation coefficient r for similarity between two users' rating vectors $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j \in J}(x_j - \mu_x)(y_j - \mu_y)}{\sqrt{\sum_{j \in J}(x_j - \mu_x)^2 \sum_{j \in J}(y_j - \mu_y)^2}},$$

where $J = \{j \mid x_j \neq na, y_j \neq na\}$ (items rated by both) and μ_x and μ_y are average ratings. Two alternatives for μ_x:

i) $\mu_x = \frac{1}{|J|} \sum_{j \in J} x_j$ (only common items) or

ii) $\mu_x = \frac{1}{|J_x|} \sum_{j \in J_x} x_j$, where $J_x = \{j \mid x_j \neq na\}$ (all rated items; more common approach)

Note that the choice of μ_x affects the results. Beware special cases, where the denominator becomes zero! (E.g., a user has given only constant ratings.)

Basic approach to predict missing ratings in rating vector $\mathbf{x}$:

1. search K nearest neighbours of $\mathbf{x}$, notated $NN_{\mathbf{x}}$, using similarity r
2. remove neighbours from $NN_{\mathbf{x}}$ if $r \leq \theta$ (negative or weak correlations)
3. normalize ratings: $y'_j = y_j - \mu_{\mathbf{y}}$ (since in different scales)
4. calculate predicted rating for all items j with missing entries in $\mathbf{x}$:

$$\tilde{x_j} = \frac{\sum_{\mathbf{y} \in NN_{\mathbf{x}}} w_{\mathbf{y}} \cdot y'_j}{\sum_{\mathbf{y} \in NN_{\mathbf{x}}} w_{\mathbf{y}}} + \mu_x.$$

   Here weight $w_{\mathbf{y}}$ can be either 1 or the similarity $w_{\mathbf{y}} = r(\mathbf{x}, \mathbf{y})$ (more robust approach). I.e., the prediction is the weighted average rating by similar users + μ_x to return back to $\mathbf{x}$'s original scale.

Note: if the jth rating is missing also from $\mathbf{y}$, choose another neighbour (goal: use always K neighbours) $\Rightarrow$ different neighbors may be used to predict ratings of different items.

3 Example

Assume that $K = 2$ and we require that similarity is at least $r \geq 0.5$.
The mean ratings per user are
$\mu_1 = 2.000$
$\mu_2 = 2.333$
$\mu_3 = 3.400$
$\mu_4 = 2.800$

User-user similarities are:

	u_1	u_2	u_3	u_4
u_1	1.000 (4)	0.836 (4)	0.927 (3)	-0.917 (3)
u_2	0.836 (4)	1.000 (6)	0.822 (5)	-0.974 (5)
u_3	0.927 (3)	0.822 (5)	1.000 (5)	-0.862 (4)
u_4	-0.917 (3)	-0.974 (5)	-0.862 (4)	1.000 (5)

The number of common ratings is given in parentheses. If the r calculation is based on too few ratings, it is unreliable, but in toy examples, we can accept similarity based on even as few as 2–3 ratings. Just remember this with real systems! Rather less neighbours with reliable similarity values than many with unreliable similarities.

Let us predict the rating of user u_1 for movie m_1. The similarities between u_1 and other users are:
$r(u_1, u_2) = 0.836$
$r(u_1, u_3) = 0.927$
$r(u_1, u_4) = -0.917$

So, the two nearest neigbours are u_2 and u_3 and both similarities are sufficiently strong. Both u_2 and u_3 have also rated movie m_1 and the prediction can be calculated.
The predicted rating for u_1, m_1 is $\frac{0.836 \cdot (3-2.333) + 0.927 \cdot (4-3.400)}{0.836+0.927} + 2.000 = 2.632$.
This is more than u_1's mean rating, and thus we can assume u_1 will like the movie more than average and recommend it.
The predicted rating for u_1, m_6 is 3.158 (recommended).
For u_3, the nearest neighbours are u_1 and u_3. Similarities are strong and both have rated m_5. The predicted rating for m_5 is 4.713 (recommended).
For u_4, there are not enough sufficiently similar neighbours (all correlations negative) and predictions cannot be made (with this basic scheme).