Homework 1 Solution

May 29, 2021

1 Problem 1

To begin with, we compute the user bias corrected rating matrix:

Item-ID ⇒	1	2	3	4	5	6
1	0	1	2	-1	-2	?
2	0	?	-1	?	1	0
3	?	0.75	1.75	-1.25	-1.25	?
4	2.2	-0.8	-1.8	1.2	?	-0.8
5	-1.6	?	0.4	-0.6	-0.6	2.4

1.1 User-based CF

To predict the value of unspecified ratings of user 2, we compute the user-user Pearson correlation according to common entries:

$$Sim(1,2) = \frac{(2\times -1) + (-2\times 1)}{\sqrt{2^2 + (-2)^2}\times\sqrt{(-1)^2+1}} = -1$$

$$Sim(2,3) = \frac{(-1\times 1.75) + (1\times -1.25)}{\sqrt{(-1)^2 + 1^2}\times\sqrt{1.75^2+(-1.25)^2}} \approx -1$$

$$Sim(2,4) = \frac{(0\times 2.2) + (-1\times -1.8) + (0\times -0.8)}{\sqrt{0^2 + (-1)^2 + 0^2}\times\sqrt{2.2^2+(-1.8)^2+(-0.8)^2}} \approx 0.6$$

$$Sim(2,5) = \frac{(0\times -1.6) + (-1\times 0.4) + (1\times -0.6) + (0\times 2.4)}{\sqrt{0^2 + (-1)^2 + 1^2 + 0^2}\times\sqrt{(-1.6)^2+0.4^2+(-0.6)^2+2.4^2}} \approx -0.2$$

Since we need to filter out negative correlation, only user 4 survives, and we directly use bias-corrected rating of user 4 plus the bias of user 2 to obtain the final prediction, which are 3.2 and 5.2 respectively.

1.2 Item-based CF

To predict the value of unspecified rating of user 2 on item 2/4, we compute the Cosine similarity between item 2/4 and other rated items (Cosine similarity of columns of the bias corrected rating matrix), which is approximately the following results

j	1	3	5	6
Cosine(2, j)	-0.6	1	-1	1
Cosine(4, j)	0.8	-1	0.9	-0.7

Filtering out negative correlation and use the weighted average of (original) ratings of other items to make prediction, we get:

$$r_{2,2} \approx \frac{1 \times 3 + 1 \times 4}{1 + 1} = 3.5$$

$$r_{2,4} \approx \frac{0.8 \times 4 + 0.9 \times 5}{0.8 + 0.9} \approx 4.5$$

2 Problem 2

The standard latent factor model objective function:

$$\frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js})^2$$

When we fix V, each u_i can be optimized separately, write down the relevant terms of u_i in matrix form:

$$L_i = \frac{1}{2} \sum_{j:(i,j) \in S} (r_{ij} - v_j^T u_i)^2$$

This is a standard least square linear regression problem, which has a closed-form solution. Write down the above objective in matrix form:

$$L_i = ||V u_i - r_i||^2$$

where $V = [v_{j_1}; v_{j_2}; \ldots; v_{j_k}]^T$ is the transpose of the matrix of relevant v vectors, then the standard solution of linear least square is:

$$u_i = (V^T V)^{-1} V^T r_i$$

Similarly, because of symmetric role of u and v, we also have

$$v_j = (U^T U)^{-1} U^T r_j$$

Remark: in practice, the linear least square problem is not solved using the above analytical form, but rather using iterative numerical methods, which is faster and more stable.

3 Problem 3

3.1 Item-based CF Regression POV

For each item, we use the ratings of the remaining $M-1$ items to make prediction, thus the total number of shared parameters is $M(M-1)$. There is no user-specific free parameter.

Alternatively, if M is large, we can use the adjusted cosine similarity to prune out irrelevant parameters, suppose for each item we only consider its k nearest neighbor and set remaining coefficients to 0. In this formulation, the total number of shared parameters is Mk.

3.2 Latent Factor Model

Each item has 100 dimensions in its latent factor vector, so the total number of shared parameters is $100M$ in the unified model. Each user vector also has 100 dimensions, so the number of free parameters for each user is 100. The total number of parameters is $100(M+N)$.

3.3 Restricted Boltzmann Machine

In RBM, all users are treated as data points for the RBM model, there is no user-specific free parameter. In RBM, the number of visible units is equal to the number of items, since we have 100 hidden units, the weight matrix is of dimension $100 \times M$. The total number of bias parameters is equal to $100+M$. Therefore, the total number of shared parameters is equal to $100M+100+M$.

4 Problem 4

In stochastic gradient descent, each time we consider exactly one data point, and in this case it is a triple (i,j,k).

$$J_{i,j,k} = \max(0, 1 - u_i^T v_j + u_i^T v_k)$$

The stochastic gradient is the gradient of $J_{i,j,k}$ with respect to the model parameters:

$$\frac{\partial J_{i,j,k}}{\partial u_i} = \begin{cases} 0 & (1 - u_i^T v_j + u_i^T v_k) < 0 \\ v_k - v_j & \textbf{Otherwise} \end{cases}$$

$$\frac{\partial J_{i,j,k}}{\partial v_j} = \begin{cases} 0 & (1 - u_i^T v_j + u_i^T v_k) < 0 \\ -u_i & \textbf{Otherwise} \end{cases}$$

$$\frac{\partial J_{i,j,k}}{\partial v_k} = \begin{cases} 0 & (1 - u_i^T v_j + u_i^T v_k) < 0 \\ u_i & \textbf{Otherwise} \end{cases}$$

5 Problem 5

There are many ways this can be done. In the following we describe a very simple approach based on cross validation:

1. Partition the entire dataset into 10 subsets of equal size
2. Assume that there are k_1, k_2, k_3 levels of granularity for each context dimension respectively. Then the total number of choices for aggregation level combination is just $k_1 \times k_2 \times k_3$.
3. For each aggregation level combination, consider ith validation dataset: use pre-filtering technique to partition the remaining 9 subsets into corresponding cells, train a CF model for each cell, and evaluate its performance on the validation dataset.
4. Compute the mean squared error of all predictions over all cells, then average over 10 different validation sets to obtain the final validation error of the aggregation level combination.
5. Choose the aggregation level combination that minimizes the mean squared error of prediction.
6. Train a final model via pre-filtering using all available data points.

6 Problem 6

The objective function is

$$J = \frac{1}{2} \sum_{(i,j,k) \in S} (r_{ijk} - u_i^T v_j - v_j^T c_k - u_i^T c_k)^2$$

Each time we fix two of U, V, C and update the remaining one. For instance, we can fix V, C and update U, and each u_i can be updated independently.

Write down the relevant terms of u_i:

$$J_i = \frac{1}{2} \sum_{(j,k):(i,j,k) \in S} (r_{ijk} - v_j^T c_k - (v_j + c_k)^T u_i)^2$$

This is again a standard linear least square problem, and we can write it down in matrix form:

$$J_i = \frac{1}{2} ||X_i u_i - Y_i||$$

where

$$X_i = [v_{j_1} + c_{k_1}; v_{j_2} + c_{k_2}; \ldots; v_{j_n} + c_{k_n}]^T$$

and

$$Y_i = [r_{ij_1k_1} - v_{j_1}^T c_{k_1}; \ldots; r_{ij_nk_n} - v_{j_n}^T c_{k_n}]^T$$

The update formula for u_i is

$$u_i = (X_i^T X_i)^{-1} X_i^T Y_i$$

The update formula for v_j and c_k are similar.