Machine Discovery HW3 Report

0. Team Members

> Validation, Testing and Debugging
> — B03902010 耿宗揚
>
> Task2/3 Model Design and Implementation
> — B03902015 簡瑋德
>
> Task1 Model Design and Implementation
> — B03902086 李鈺昇

1. Description of Tasks

Discovering connections in/across domains to boost the quality of recommendation.

- Given
  - Sparse source/target rating matrix with many unknown entries.
- Output
  - Some predicted values in target rating matrix.

Task	Domain	User	Item	Mapping	Sets of records
1	same	same	same	unknown	disjoint
2	same	different	different	unknown	mostly disjoint
3	different	different	different	unknown	disjoint

2. Implementations

Task 1

Let the target, source matrices be R_1, R_2, respectively.

For $i \in \{1, 2\}$, we first use matrix factorization (MF) to obtain $P_i Q_i^T \approx R_i$. The first dimensions of P and Q are clearly the dimensions of R, and the second dimensions of P and Q are some chosen integer, K. For the sake of memory usage and running time, we only test $K \leq 10$. Larger value of K may help increase the rating performance, but it's a trade-off between resources and results.

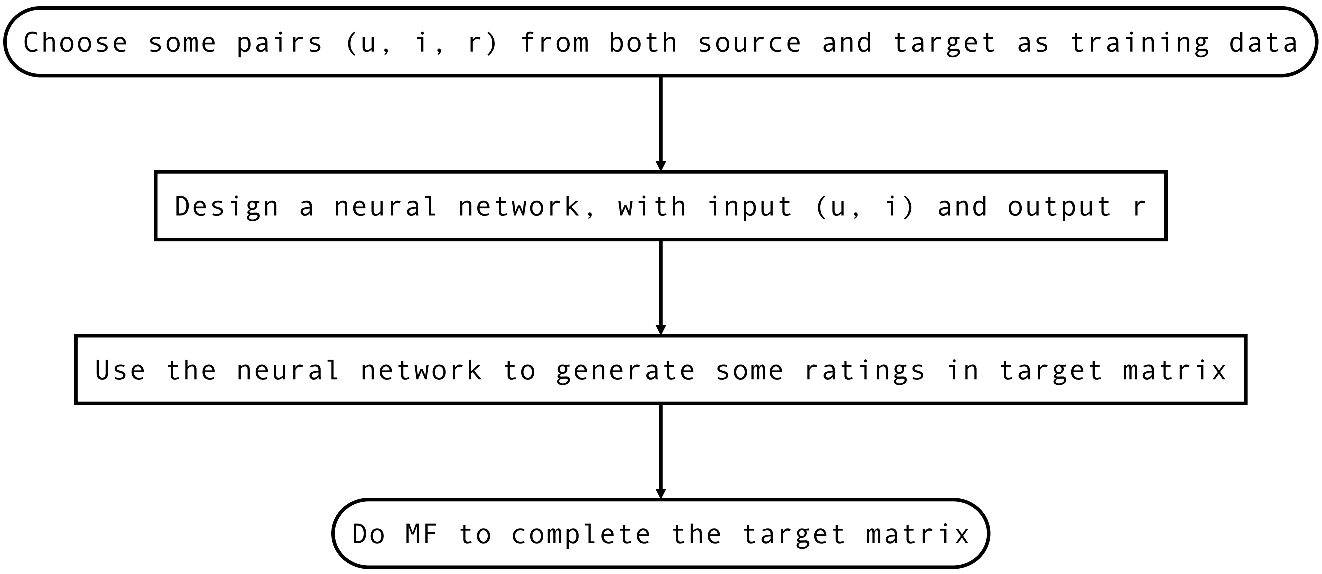
Next, by singular value decomposition (SVD), we obtain orthogonal U_i, V_i and diagonal D_i such that $\hat{R}_i \equiv U_i D_i V_i^T \approx P_i Q_i^T$.

Since the mapping is unknown, one simple way is to find out two auxiliary permutation-like matrices G_L, G_R such that $\hat{R}_1 \approx G_L \hat{R}_2 G_R$. Then we let $R \equiv \lambda \hat{R}_1 + (1-\lambda) G_L \hat{R}_2 G_R$ to be the resulting rating matrix with the help of R_2, where $\lambda \in [0, 1]$ is determined by cross validation. Finally, we choose λ to be 0.38. More details are given later.

Now we can make predictions with R. Given user-item pair (u, i), our predicted rating that user u gives item i is simply R_{ij}.

Task 2

Flow



Input of the Neural Network

$$\text{For each (user, item) pair, } D = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{|U|} \\ i_1 \\ i_2 \\ \vdots \\ i_{|I|} \end{bmatrix}, \text{ where}$$

- $|U|$ is the number of users and $|I|$ is the number of items
- $u_k = \mathbb{1}$[user is the k^{th} one]
- $i_k = \mathbb{1}$[item is the k^{th} one]

Variables of the Neural Network

$$\text{Embedding Matrix } E = \begin{bmatrix} e_{11} & \cdots & e_{1|U|} & e_{1(|U|+1)} & \cdots & e_{1(|U|+|I|)} \\ e_{21} & \cdots & e_{2|U|} & e_{2(|U|+1)} & \cdots & e_{2(|U|+|I|)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{r1} & \cdots & e_{r|U|} & e_{r(|U|+1)} & \cdots & e_{r(|U|+|I|)} \end{bmatrix}$$

$$\text{Weight Vector } W = \begin{bmatrix} w_1 & w_2 & \dots & w_r \end{bmatrix}$$

- r is the dimension of the latent vector (embedding), in this task, we set $r = 10$
- $[e_{1k}, e_{2k}, \dots, e_{rk}]$ is the latent vector of the k^{th} user
- $[e_{1(|U|+k)}, e_{2(|U|+k)}, \dots, e_{r(|U|+k)}]$ is the latent vector of the k^{th} item

Output and Training of the Neural Network

- Potential Value $P = W \cdot E \cdot D$
- Output Rating $R = \frac{1}{1+e^{-P}}$, where $0 \leq R \leq 1$
- Lost Function $= \sum_{(u,i)\in\text{Training Pairs}}(R_{u,i} - r)^2$
- Parameters are initialized using random uniform distribution and updated using gradient decent

Adding New Ratings to Target Matrix

- After training the neural network, we are able to add some new ratings
- If we have the latent vector of a user u and item i, we can calculate the rating r using the network

Performing Matrix Factorization

- When the target matrix becomes denser, perform MF to complete the matrix
- Parameters are randomly initialized

Task 3

The only differences between the models of Task 2 and Task 3 are the weight matrix and the output layer.

$$\text{Weight Matrix } W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1r} \\ \vdots & \vdots & \ddots & \vdots \\ w_{51} & w_{52} & \dots & w_{5r} \end{bmatrix}$$

$$\text{For Output Layer, Potential } P = \begin{bmatrix} p_1 \\ \vdots \\ p_5 \end{bmatrix} = W \cdot E \cdot D$$

$$\mathbb{P}(R = r) = \frac{e^{p_r}}{\sum_{i=1}^{5} e^{p_i}}, r \in \{1, 2, 3, 4, 5\}, \text{which is the softmax function.}$$

3. More Details on Validation

We only do validation for task 1, partly because we need to find out the best value of λ. We randomly take 10% of the data from train.txt for validation. The results for different values of λ follows.

λ	**RMSE**
0.0	0.19121172836885114
0.1	0.18981906234427
0.2	0.18888050921949595
0.3	0.1884028557827018
0.4	0.1883896078382336
0.5	0.18884086312584142
0.6	0.18975330774741603
0.7	0.1911203364018659
0.8	0.1929322864166143
0.9	0.19517676672141657
1.0	0.19783905672039154

And, as mentioned above, 0.38 is the value we choose.

4. References

[1] https://www.tensorflow.org/tutorials/ (https://www.tensorflow.org/tutorials/)

[2] C.-Y. Li and S.-D. Lin. Matching Users and Items Across Domains to Improve the Recommendation Quality