Deep Matrix Factorization Models for Recommender Systems

- Nanjing University
- Hong-Jian Xue

- IJCAI 2017

2019年6月

Content

- Motivation
- DMF Model

Motivation

- Collaborative filtering (CF)
- Matrix factorization (MF)
  - Biased matrix factorization
  - Additional extra data
- Deep learning methods
  - Explicit ratings
  - Implicit feedback

Motivation

- Collaborative filtering (CF)
- Matrix factorization (MF)
  - Biased matrix factorization
  - Additional extra data
- Deep learning methods
  - Explicit ratings
  - Implicit feedback

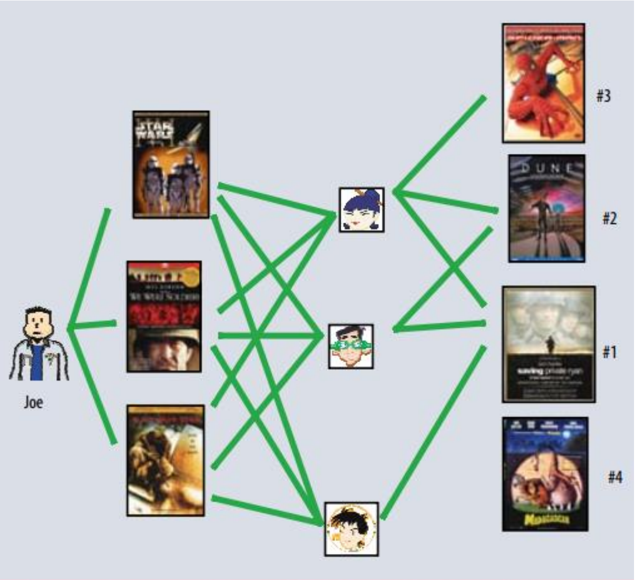
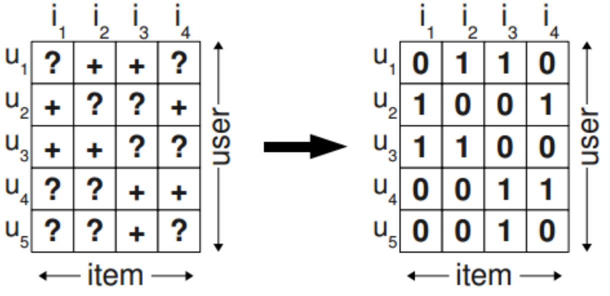
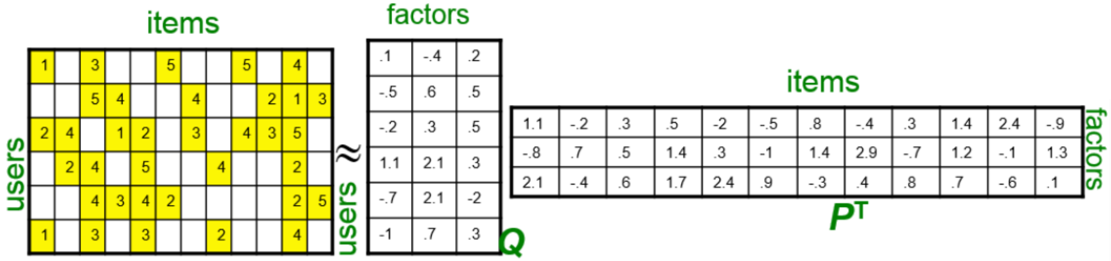


Figure 1. The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked *Saving Private Ryan*, so that is the first recommendation. Two of them liked *Dune*, so that is next, and so on.





Motivation

- Collaborative filtering (CF)
- Matrix factorization (MF)

  $\hat{r}_{ui} = q_i^T p_u.$

  - Biased matrix factorization

    $\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$

  - Additional extra data

    social relations
    contents or reviews text

- Deep learning methods
  - Explicit ratings

    We refer to explicit user feedback as *ratings*.

  - Implicit feedback

    implicit feedback, which indirectly reflects opinion by observing user behavior including purchase history, browsing history, search patterns, or even mouse movements.

DMF Model

- To make use of both explicit ratings and implicit feedback, we propose a new neural matrix factorization model for *top-N recommendation*.
- We design a new loss function based on binary cross entropy, in which we consider both explicit ratings and implicit feedback for a better optimization.

Relevance measured by cosine similarity

$$\hat{Y}_{ij} = F^{DMF}(u_i, v_j|\Theta) = cosine(p_i, q_j) = \frac{p_i^T q_j}{\|p_i\| \|q_j\|}$$

Latent representation

$$p_i = f_{\theta_N^U}(...f_{\theta_3^U}(W_{U2} f_{\theta_2^U}(Y_{i*} W_{U1}))...)$$

$$q_j = f_{\theta_N^I}(...f_{\theta_3^I}(W_{V2} f_{\theta_2^I}(Y_{*j}^T W_{V1}))...)$$

Multi-layer non-linear projection

Output layer & Hidden layer: *Relu*

$$f(x) = max(0, x)$$

$$l_1 = W_1 x$$
$$l_i = f(W_{i-1} l_{i-1} + b_i), i = 2, ..., N-1$$
$$h = f(W_N l_{N-1} + b_N)$$

Interaction Matrix Y

$$Y_{ij} = \begin{cases} 0, & if \quad R_{ij} = unk \\ R_{ij}, & otherwise \end{cases}$$

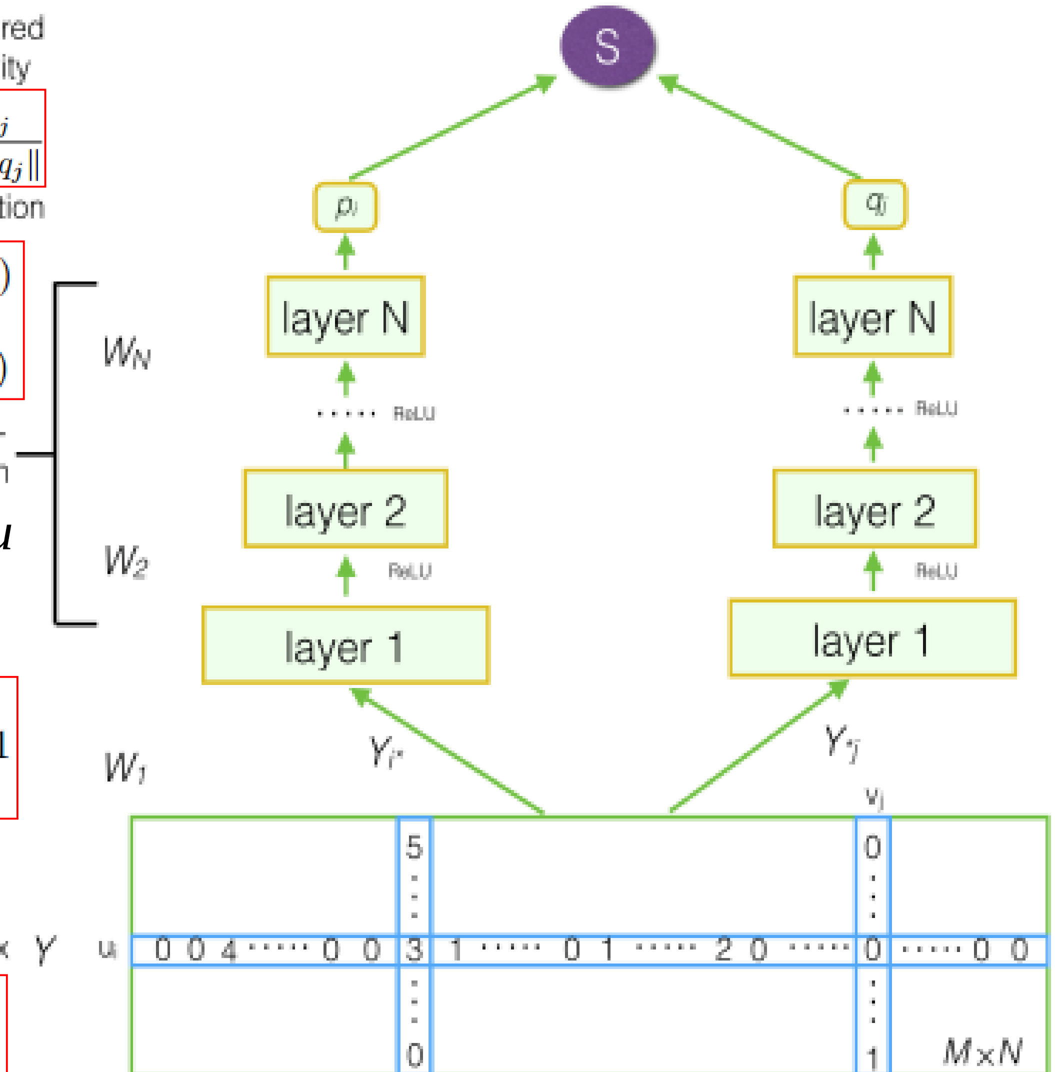


Figure 1: The architecture of Deep Matrix Factorization Models

Loss Function

- Loss function
  - Squared loss

$$\boxed{L_{sqr} = \sum_{(i,j)\in Y^{+}\cup Y^{-}} w_{ij}(Y_{ij} - \hat{Y}_{ij})^2}$$

  - The binary cross-entropy loss (ce)

$$\boxed{L = - \sum_{(i,j)\in Y^{+}\cup Y^{-}} Y_{ij} log \hat{Y}_{ij} + (1 - Y_{ij}) log(1 - \hat{Y}_{ij})}$$

Loss Function

- Normalized cross entropy loss
  - Incorporate the explicit ratings into cross entropy loss, so that explicit and implicit information can be used together for optimization.

$$L_{sqr} = \sum_{(i,j)\in Y^+ \cup Y^-} w_{ij}(Y_{ij} - \hat{Y}_{ij})^2$$

$$L = -\sum_{(i,j)\in Y^+ \cup Y^-} Y_{ij} log \hat{Y}_{ij} + (1 - Y_{ij}) log(1 - \hat{Y}_{ij})$$

$$L = -\sum_{(i,j)\in Y^+ \cup Y^-} \left(\frac{Y_{ij}}{max(R)} log \hat{Y}_{ij} + (1 - \frac{Y_{ij}}{max(R)}) log(1 - \hat{Y}_{ij})\right)$$

We use the $max(R)$ (5 in a 5-star system) for normalization which is the max score in all ratings, so that different values of Y_{ij} have different influences to the loss.

DMF Training Algorithm with Normlized Cross Entropy

Algorithm 1 DMF Training Algorithm With Normalized Cross Entropy

Input: $Iter$: # of training iterations,
neg_ratio: Negative sampling ratio,
R: original rating matrix,

Output: W_{Ui}(i=1..N-1): weight matrix for user,
W_{Vi}(i=1..N-1): weight matrix for item,

Initialisation :
randomly initialize W_U and W_V;
set $Y \leftarrow$ use Equation 2 with R;
set $Y^+ \leftarrow$ all none zero interactions in Y;
set $Y^- \leftarrow$ all zero interactions in Y;
set $Y^-_{sampled} \leftarrow$ sample $neg\_ratio * \|Y^+\|$ interactions from Y^-;
set $T \leftarrow Y^+ \cup Y^-_{sampled}$;
for it from 1 to $Iter$ **do**
for each interaction of User i and Item j in T **do**
set $p_i, q_j \leftarrow$ use Equation 7 with input of Y_{i*}, Y_{*j};
set $\hat{Y}^o_{ij} \leftarrow$ use Equation 8,13 with input of p_i, q_j;
set $L \leftarrow$ use Equation 11 with input of $\hat{Y}^o_{ij}, Y_{ij}$;
use back propagation to optimize model parameters
end for
end for

$$Y_{ij} = \begin{cases} 0, & if \quad R_{ij} = unk \\ R_{ij}, & otherwise \end{cases} \tag{2}$$

$$\begin{aligned} p_i &= f_{\theta^U_N}(\ldots f_{\theta^U_3}(W_{U2} f_{\theta^U_2}(Y_{i*}W_{U1}))\ldots) \\ q_j &= f_{\theta^I_N}(\ldots f_{\theta^I_3}(W_{V2} f_{\theta^I_2}(Y^T_{*j}W_{V1}))\ldots) \end{aligned} \tag{7}$$

$$\hat{Y}_{ij} = F^{DMF}(u_i, v_j|\Theta) = cosine(p_i, q_j) = \frac{p_i^T q_j}{\|p_i\| \, \|q_j\|} \tag{8}$$

$$\hat{Y}^o_{ij} = max(\mu, \hat{Y}_{ij}) \tag{13}$$

For cross entropy loss, because the predicted score of Y_{ij} can be negative, we need to use Equation 13 to transform the original predictions. Let μ be a very small number, and we set $1.0e^{-6}$ in our experiments.

$$L = -\sum_{(i,j)\in Y^+ \cup Y^-} \left(\frac{Y_{ij}}{max(R)} log\hat{Y}_{ij} + (1 - \frac{Y_{ij}}{max(R)}) log(1 - \hat{Y}_{ij})\right) \tag{12}$$