Complex Networks and Statistical Learning
Homework 1

Chenghua Liu
liuch18@mails.tsinghua.edu.cn
Department of Computer Science
Tsinghua University

1 Neighborhood-based CF

"Recommender Systems: The Textbook", Exercise 2.10.2

Consider the following ratings table between five users and six items:

Item-Id ⇒	1	2	3	4	5	6
1	5	6	7	4	3	?
2	4	?	3	?	5	4
3	?	3	4	1	1	?
4	7	4	3	6	?	4
5	1	?	3	2	2	5

(a) Predict the values of unspecified ratings of user 2 using user-based collaborative filtering algorithms. Use the Pearson correlation with mean-centering.

Item-Id ⇒	1	2	3	4	5	6	$Mean\ Rateing$	$Pearson(i, 2)$
1	5	6	7	4	3	?	5	-1.0000
2	4	?	3	?	5	4	4	1.0000
3	?	3	4	1	1	?	2.25	-0.9864
4	7	4	3	6	?	4	4.8	0.4310
5	1	?	3	2	2	5	2.6	-0.2378

$\hat{r}_{2v} = \mu_2 + \frac{\sum_{v \in P_2(j)} \mathrm{Sim}(2,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_2(j)} |\mathrm{Sim}(2,v)|}$

We set k= 4 in k-closest users, which means that we consider all other users. Then,

$\hat{r}_{22} = 3.14 \quad , \hat{r}_{24} = 5.09$

(b) Predict the values of unspecified ratings of user 2 using item-based collaborative filtering algorithms. Use the adjusted cosine similarity.

Item-Id $\Rightarrow$	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
$Cosine(2, j)$	-0.6247	1	0.9977	-0.9759	-0.9964	1
$Cosine(4, j)$	0.7909	-0.9759	-0.9751	1	0.9428	-1

$\hat{r}_{2t} = \frac{\sum_{j \in Q_{t(2)}} \text{AdjustedCosine } (j,t) \cdot r_{2j}}{\sum_{j \in Q_{t(2)}} | \text{ AdjustedCosine } (j,t)|}$

We set k= 2 in k-closest items. Then,

$\hat{r}_{22} = 3.50 \quad , \hat{r}_{24} = 4.54$

2 Latent Factor Model

Derive the update formula for using alternating least square to optimize the following standard latent factor model:

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

compute the partial derivative of J:

$$\frac{\partial J}{\partial u_{iq}} = \sum_{j:(i,j) \in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right) (-v_{jq}) = 0 \quad \forall i \in \{1 \ldots m\}, q \in \{1 \ldots k\}$$

$$\Rightarrow u_{iq} = \frac{\sum_{j:(i,j) \in S} (e_{ij} + u_{iq} v_{jq}) v_{jq}}{\sum_{j:(i,j) \in S} v_{jq}^2}$$

$$\frac{\partial J}{\partial v_{jq}} = \sum_{i:(i,j) \in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right) (-u_{iq}) = 0 \quad \forall j \in \{1 \ldots n\}, q \in \{1 \ldots k\}$$

$$\Rightarrow v_{jq} = \frac{\sum_{i:(i,j) \in S} (e_{ij} + u_{iq} v_{jq}) u_{iq}}{+\sum_{i:(i,j) \in S} u_{iq}^2}$$

3 Unified Framework

What is the total number of free/shared parameters for the following methods (assuming N users and M items):

- Regression version of item-based CF
- Latent Factor Model with 100 dimensions
- Standard Restricted Boltzmann Machine with 100 hidden units

Explain your answer

(a) we now that the regression version of item-based CF has the fomula:

$$\hat{r}_{ut} = \sum_{j \in Q_t(u)} w_{jt}^{item} \cdot r_{uj}$$

Where the w_{jt}^{item} is unknown parameter. So the total number of free parameters is $m \times m$.

(b) the total number of free parameters is $k \times (m+n) = 100(m+n)$.

(c) the total number of free parameters is $100 \times m + m + 100$.

4 Learning to Rank

Derive the stochastic gradient of parameters in latent factor model when using pairwise rank learning objective (hinge loss)

$$\text{Minimize } J = \sum_{(i,j,k)} \max\left(0, 1 - u_i^T v_j + u_i^T v_k\right)$$

By equivalent change, we get

$$\text{Minimize } J = \sum_{(i,j,k)} 1 - u_i^T v_j + u_i^T v_k$$

$$\text{suject to} \quad u_i^T(v_j - v_k) - 1 \leq 0$$

This is a constrained optimization problem, we add the logarithmic barrier function to approximate the unconstrained problem:

$$\text{Minimize } J = \sum_{(i,j,k)} 1 - u_i^T v_j + u_i^T v_k - \frac{1}{t}\log(1 - u_i^T v_j + u_i^T v_k)$$

compute the partial derivative of J:

$$\frac{\partial J}{\partial u_i} = \sum_{(i,j,k)} (v_k - v_j)[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

$$u_i \Leftarrow u_i - \alpha \times \sum_{(i,j,k)} (v_k - v_j)[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

$$\frac{\partial J}{\partial v_j} = \sum_{(i,j,k)} -u_i[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

$$v_j \Leftarrow v_j + \alpha \times \sum_{(i,j,k)} u_i[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

$$\frac{\partial J}{\partial v_k} = \sum_{(i,j,k)} u_i[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

$$v_k \Leftarrow v_k - \alpha \times \sum_{(i,j,k)} u_i[1 - \frac{\lambda}{1 - u_i^T(v_j - v_k)}]$$

Where $\lambda = \frac{1}{t}$.

5 Context-sensitive CF

"Recommender Systems: The Textbook", Exercise 8.8.4

Suppose that you have three contextual attributes (say, location, time, and companion), each of which has its own taxonomy. Your system is designed to recommend items for a given user in the context of location, time, and companion. For a given context at the lowest level of the hierarchy, you might have the sparsity problem because only a modest number (say, 500) of the observed ratings might be available in which the three contexts are fixed to the queried values. This can cause overfitting in a pre-filtering method if only 500 ratings are used for the training process. You decide that you will use a more general level of the taxonomy from each of the three contexts, in order to extract the relevant segments and increase the amount of training data. Describe how to determine the specific level of the taxonomy to use for each contextual attribute. Once you have extracted the taxonomy level for each context, describe the collaborative filtering algorithm.

We choose to use dimensionality reduction methods such as singular value decomposition, principal component analysis. It compress user-item matrix into a low-dimensional representation in terms of latent factors. One advantage of using this approach is that instead of having a high dimensional matrix containing abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space.

Then we choose neighborhood-based method to collaborative filter. Calculate the similarity of rating matrices corresponding to each user by using Pearson coefficients. After the most like-minded users are found, their corresponding ratings are aggregated to identify the set of items to be recommended to the target user.

6 Context-sensitive Latent Factor Model

Write down the full objective function of pairwise interaction tensor factorization, assuming a single context dimension. Derive the update formula for using alternating least square to optimize it.

In general, we consider the case where the context dimension is d.

Let $R = [r_{ijc}]$be a 3-dimensional ratings cube of size m×n×d with m users, n items, and d different values of the contextual dimension.Let $U = [u_{is}]$, $V = [v_{js}]$, and $W = [w_{cs}]$ be $m \times k, n \times k$, and $d \times k$, matrices. Here, U denotes the userfactor matrix, V denotes the item-factor matrix, and W denotes the context-factor matrix. The notation k denotes the rank of the latent factor model. Then, the prediction function:

$$\begin{aligned}\hat{r}_{ijc} &= \left(UV^T\right)_{ij} + \left(VW^T\right)_{jc} + \left(UW^T\right)_{ic} \\ &= \sum_{s=1}^{k} \left(u_{is}v_{js} + v_{js}w_{cs} + u_{is}w_{cs}\right)\end{aligned}$$

It is easy to see that this prediction function is a straightforward generalization of latent factor models. As in all latent factor models, let S be the set of all observed entries in R.

$$S = \{(i, j, c) : r_{ijc} \text{ is observed } \}$$

Then, the errors over all the observed entries needs to be minimized by using alternating least square:

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j,c)\in S} \left(r_{ijc} - \hat{r}_{ijc}\right)^2 = \frac{1}{2} \sum_{(i,j,c)\in S} \left(r_{ijc} - \sum_{s=1}^{k} [u_{is}v_{js} + v_{js}w_{cs} + u_{is}w_{cs}]\right)^2$$

compute the partial derivative of J:

$$\frac{\partial J}{\partial u_{iq}} = \sum_{j,c:(i,j,c)\in S} e_{ijc} \cdot (v_{jq} + w_{cq}) = 0 \quad \forall i \quad \forall q \in \{1 \dots k\}$$
$$\Rightarrow u_{iq} = \frac{\sum_{j,c:(i,j,c)\in S} [e_{ijc} + u_{iq}(v_{jq} + w_{cq})](v_{jq} + w_{cq})}{\sum_{j,c:(i,j,c)\in S} (v_{jq} + w_{cq})^2}$$
$$\frac{\partial J}{\partial v_{jq}} = \sum_{i,c:(i,j,c)\in S} e_{ijc} \cdot (u_{iq} + w_{cq}) = 0 \quad \forall j \quad \forall q \in \{1 \dots k\}$$
$$\Rightarrow v_{jq} = \frac{\sum_{i,c:(i,j,c)\in S} [e_{ijc} + v_{jq}(u_{iq} + w_{cq})](u_{iq} + w_{cq})}{\sum_{i,c:(i,j,c)\in S} (u_{iq} + w_{cq})^2}$$
$$\frac{\partial J}{\partial w_{cq}} = \sum_{i,j:(i,j,c)\in S} e_{ijc} \cdot (u_{iq} + v_{jq}) = 0 \quad \forall c \quad \forall q \in \{1 \dots k\}$$
$$\Rightarrow w_{cq} = \frac{\sum_{i,j:(i,j,c)\in S} [e_{ijc} + w_{cq}(v_{jq} + u_{iq})](v_{jq} + u_{iq})}{\sum_{i,j:(i,j,c)\in S} (v_{jq} + u_{iq})^2}$$