A Privacy-Preserving QoS Prediction Framework for Web Service Recommendation

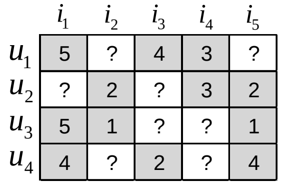
[Supplementary Report]

Jieming Zhu, Pinjia He, Zibin Zheng, Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
{jmzhu, pjhe, zbzheng, lyu}@cse.cuhk.edu.hk

This report is provided as supplemental material for our paper, which comprises two appendixes as follows: collaborative filtering, and mathematical basics.

Appendix A
Collaborative Filtering

Collaborative filtering (CF) techniques are commonly used in commercial recommender systems, such as movie recommendation in Netflix[1] and item recommendation in Amazon[2]. The CF model has been widely studied in recent years. In recommender systems, CF works for the rating prediction problem. Specifically, users likely rate the items that they know about, such as $1 \sim 5$ stars for the moives they have watched or books they have read. As illustrated in Fig. 1, the values in grey entries are observed rating data, and the blank entries are unknown values. For example, the rating value between user u_1 and iterm i_1 is 5, while the rating value between user u_1 and iterm i_5 is missing, because u_1 has not rated i_5. In practice, each user usually rate only a small set out of all of the items, due to the large number of items. As a result, the user-item rating matrix is very sparse.



	i_1	i_2	i_3	i_4	i_5
u_1	5	?	4	3	?
u_2	?	2	?	3	2
u_3	5	1	?	?	1
u_4	4	?	2	?	4

Fig. 1. An Example of Rating Prediction

The basic idea of CF is to exploit and model the observed data to predict the unknown values, based on the insight that similar users may have similar preferences on the same item, and thus have similar ratings. To achieve this goal, two types of CF techniques have been studied in recent literature: neighbourhood-based approaches and model-based approaches [1].

Neighbourhood-based CF approaches: Neighbourhood-based approaches include user-based approaches (*e.g.*, UPCC) that leverage the similarity between users, item-based approaches (*e.g.*, IPCC) that employ the similarity between items, and their fusions (*e.g.*, UIPCC [2]). However, neighbourhood-based approaches are incapable of handling the data sparsity problem and have high time complexity.

Model-based CF approaches: Model-based approaches provide a predefined compact model to fit the training data, which can be further used to predict the unknown values. Matrix factorization [3] is one of the most popular model-based approaches used for collaborative filtering. In addition, matrix factorization model can usually achieve better performance than neighbourhood-based approaches.

Appendix B
Mathematical Basics

This section provides some mathematical background for our paper.

A. *Euclidean Norm*

Euclidean norm $\|\cdot\|_2$ is a vector norm. Given a vector $V \in \mathbb{R}^n$, its Euclidean norm is defined as follows:

$$\|V\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2} \tag{1}$$

where v_i is the element of V.

B. *Gradient Descent*

Gradient descent is a widely used method to find a local minimum of an object function in an iterative way. Note that in our experiments, the approach PMF is implemented by using gradient descent algorithm, as described in the following.

As for matrix factorization model, the object function is given as follows:

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_S}{2}\|S\|_F^2, \tag{2}$$

where the definition of each symbol has been described in our paper. Then gradient descent works by updating U_i and S_j simultaneously from random initialization using the following updating rules:

$$U_i \leftarrow U_i - \eta\frac{\partial \mathcal{L}}{\partial U_i}, \quad S_j \leftarrow S_j - \eta\frac{\partial \mathcal{L}}{\partial S_j}, \tag{3}$$

[1] http://www.netflix.com
[2] http://www.amazon.com

Algorithm 1: Gradient Descent for PMF

```
Input: The collected QoS matrix $R$, the indication matrix $I$,
       and the model parameters: $\eta$, $\lambda_U$ and $\lambda_S$. /* $I_{ij} = 1$
       if $R_{ij}$ is known; otherwise, $I_{ij} = 0$ */
Output: The QoS prediction results: $\hat{R}_{ij}$, where $I_{ij} = 0$.
Initialize $U \in \mathbb{R}^{d \times n}$ and $S \in \mathbb{R}^{d \times m}$ randomly;
repeat                                   /* Batch-mode updating */
    foreach (i, j) do      /* Compute $\frac{\partial \mathcal{L}}{\partial U_i}$ and $\frac{\partial \mathcal{L}}{\partial S_j}$ */
        $\frac{\partial \mathcal{L}}{\partial U_i} \leftarrow \sum_{j=1}^{m} I_{ij}(U_i^T S_j - R_{ij})S_j + \lambda_U U_i$;
        $\frac{\partial \mathcal{L}}{\partial S_j} \leftarrow \sum_{i=1}^{n} I_{ij}(U_i^T S_j - R_{ij})U_i + \lambda_S S_j$;
    foreach (i, j) do    /* Update each $U_i$ and $S_j$ */
        $U_i \leftarrow U_i - \eta \frac{\partial \mathcal{L}}{\partial U_i}$;
        $S_j \leftarrow S_j - \eta \frac{\partial \mathcal{L}}{\partial S_j}$;
until converge;
foreach (i, j) $\in \{I_{ij} = 0\}$ do   /* Make prediction */
    $\hat{R}_{ij} = U_i^T S_j$;
```

In particular, the derivatives of U_i and S_j can be derived from Equation 2 as follows:

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{m} I_{ij}(U_i^T S_j - R_{ij})S_j + \lambda_U U_i, \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial S_j} = \sum_{i=1}^{n} I_{ij}(U_i^T S_j - R_{ij})U_i + \lambda_S S_j. \tag{5}$$

Hence, the updating rules in Equation 3 can be rewritten as follows:

$$U_i \leftarrow U_i - \eta\big(\sum_{j=1}^{m} I_{ij}(U_i^T S_j - R_{ij})S_j + \lambda_U U_i\big), \tag{6}$$

$$S_j \leftarrow S_j - \eta\big(\sum_{i=1}^{n} I_{ij}(U_i^T S_j - R_{ij})U_i + \lambda_S S_j\big). \tag{7}$$

Gradient descent works on batch-mode, which needs all the data to be available. The latent factors U_i and S_j move iteratively by a small step of the average gradient, *i.e.*, $\frac{\partial \mathcal{L}}{\partial U_i}$ and $\frac{\partial \mathcal{L}}{\partial S_j}$, where the step size is controlled by η.

The detailed algorithm of gradient descent for PMF is presented in Algorithm 1.

C. Stochastic Gradient Descent

The scheme of stochastic gradient descent (SGD) is to update the stochastically using the sequentially coming data. At each step, the model can be adjusted by only considering the current data sample. Thus, SGD naturally provides an online algorithm, where we can adjust the model using each data sample from the data stream in an online fashion.

Formally, The loss function $\mathcal{L}$ in Euqation 2 can be seen as the sum of pairwise loss functions:

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{m} I_{ij} \ell(U_i, S_j), \tag{8}$$

and the pairwise loss function $\ell(U_i, S_j)$ with respect to $(U_i, S_j, R_i j)$ is defined as

$$\ell(U_i, S_j) = \frac{1}{2}(R_{ij} - U_i^T S_j)^2 + \frac{\lambda_u}{2}\|U_i\|_2^2 + \frac{\lambda_s}{2}\|S_j\|_2^2, \tag{9}$$

Algorithm 2: Stochastic Gradient Descent for PMF

```
Input: Sequentially observed QoS data samples: $(u_i, s_j, R_{ij})$,
       and the model parameters: $\eta$, $\lambda_u$ and $\lambda_s$.
Output: The QoS prediction results: $\hat{R}_{ij}$, where $I_{ij} = 0$.
Initialize $U \in \mathbb{R}^{d \times n}$ and $S \in \mathbb{R}^{d \times m}$ randomly;
repeat                          /* Online-mode updating */
    foreach $(u_i, s_j, R_{ij})$ do
        $\frac{\partial \ell}{\partial U_i} \leftarrow (U_i^T S_j - R_{ij})S_j + \lambda_u U_i$;
        $\frac{\partial \ell}{\partial S_j} \leftarrow (U_i^T S_j - R_{ij})U_i + \lambda_s S_j$;
        $U_i \leftarrow U_i - \eta \frac{\partial \ell}{\partial U_i}$;
        $S_j \leftarrow S_j - \eta \frac{\partial \ell}{\partial S_j}$;
until converge;
foreach (i, j) $\in \{I_{ij} = 0\}$ do   /* Make prediction */
    $\hat{R}_{ij} = U_i^T S_j$;
```

Note that the regularization parameters λ_u and λ_s are on different scale from those in Equation 2. Similarly, we can derive the following updating equations for each iteration:

$$U_i \leftarrow U_i - \eta\big((U_i^T S_j - R_{ij})S_j + \lambda_u U_i\big), \tag{10}$$

$$S_j \leftarrow S_j - \eta\big((U_i^T S_j - R_{ij})U_i + \lambda_s S_j\big). \tag{11}$$

The detailed algorithm of stochastic gradient descent for PMF is presented in Algorithm 2.

References

[1] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artificial Intellegence*, 2009.

[2] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," *IEEE T. Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[3] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. of of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.