# Final Project: Matrix Factorization for Recommender Systems

Junhao Hua, 11331007 *

May 28, 2015

## 1 Introduction

This is an implementation document for the final project[1] of *Introduction to Data Mining* instructed by professor Zhongfei Zhang[2]. Broadly speaking, recommender systems are based on one of two strategies: the *content filtering* approach which creates a profile for each user or product to characterize its nature, and the *collaborative filtering* which only relies on past user behavior. The later is quite appealing due to its benefit of domain free. The two primary areas of collaborative filtering are the *neighborhood methods* and *latent factor models*. In this paper, we use the latent factor models for recommender systems and realize them by *matrix factorization*. The algorithm is implemented using MATLAB.

## 2 Algorithm

Suppose there are$M$ items and $N$ users. let $Y \in \mathbb{R}^{M \times N}$ represent the rating matrix: $Y_{ij} \neq 0$ means that item $i$ is rated by user $j$ with the rating $Y_{ij}$; $Y_{ij} = 0$ means that item $i$ has not been rated by user $j$. Let $R \in \{0, 1\}^{M \times N}$ be the corresponding indicator matrix. Matrix factorization models [1] map both users and items to a joint latent factor space of dimensionality $p$, such that user-item interactions are modeled as inner products in that space. Let $X \in \mathbb{R}^{M \times p}$ and $\Theta \in \mathbb{R}^{N \times p}$ be the latent feature matrices of items and users, respectively. Each row vector $X_i \in \mathbb{R}^{1 \times p}$ and $\Theta_j \in \mathbb{R}^{1 \times p}$ represents the $i$th item-specific and $j$th user-specific latent feature vector, respectively. The resulting dot product $X_i \Theta_j^T$ captures the interaction between item $i$ and user $j$ - the user's overall interest in the item's characteristics.

Recent work [2] suggested modelling directly the observed rating only, while avoiding overfitting through a regularized model. The recommender system minimizes the regularized squared error on the set of known ratings:

$$\min_{X,\Theta} \frac{1}{2} ||R \cdot (Y - X\Theta^T)||_F^2 + \lambda(||X||_F^2 + ||\Theta||_F^2) \tag{1}$$

*Junhao Hua, PhD candidate, Collage of ISEE, Zhejiang University, huajh7@gmail.com, http://huajh7.com/.

[1] http://10.13.71.118/index.php/projects/.

[2] http://www.isee.zju.edu.cn/dsec/zhongfei_ch.html.

where $\|\cdot\|_F$ is the Frobenius norm, the operator $\cdot$ means the dot product, and $\lambda$ is the regularization parameter. Using a (stochastic) gradient descent, We can solve this optimization problem with respect to $X$ and $\Theta$ simultaneously. Take derivative of cost function (1) w.r.t. $X$ and $\Theta$, we get the gradients

$$\Delta X = R \cdot (X\Theta^T - Y)\Theta + \lambda X, \tag{2a}$$

$$\Delta\Theta = (R \cdot (X\Theta^T - Y))^T X + \lambda\Theta. \tag{2b}$$

The gradient descent update equations for $X$ and $\Theta$ are

$$X^{t+1} = X^t - \mu\Delta X^t, \tag{3a}$$

$$\Theta^{t+1} = \Theta^t - \mu\Delta\Theta^t, \tag{3b}$$

where $\mu$ is the learning rate, which can be simply set as a constant small number (such as 0.002). After learning the map of the joint latent factor space $\{X, \Theta\}$, the missing ratings in matrix $Y$ can be approximately filled by the $P = X\Theta^T$.

The gradient descent algorithm for matrix factorization is summarized in the following.

**Algorithm 1** A Gradient Descent algorithm for Matrix Factorization

**Input:** Given the rating matrices $Y$ and $R$. The latent factor matrices $X^0, \Theta^0$ are randomly initialized, and the learning rate $\mu = 0.002$. $MaxIters = 100$.

1: Set regularization parameter $\lambda$ appropriately.
2: **for** $t \leftarrow 0, 1, 2, \ldots, MaxIters$ **do** ▷ $t$: time step
3: $\quad X^{t+1} = X^t - \mu\big(R \cdot (X^t\Theta^{t,T} - Y)\Theta^t + \lambda X^t\big)$.
4: $\quad \Theta^{t+1} = \Theta^t - \mu\big((R \cdot (X^t\Theta^{t,T} - Y))^T X^t + \lambda\Theta^t\big)$.
5: $\quad$ **if** $\frac{\|X^{t+1}-X^t\|_F^2 + \|\Theta^{t+1}-\Theta^t\|_F^2}{\|X^t\|_F^2 + \|\Theta^t\|_F^2} < \epsilon$ **then** ▷ Stop Criterion
6: $\quad\quad$ **break**.
7: $\quad$ **end if**
8: **end for**
9: **return** $P = X^{t+1}\Theta^{t+1,T}$

# 3 Experimental Results

## 3.1 Data Preparation

There are 80k ratings from 943 users ($N = 943$) on 1682 items in the given dataset. The rating value is in the range of $[1, 5]$ in integer. This dataset is partial data from MovieLens 100k dataset[3], which has 100k ratings totally. We use the given 80k ratings data for training and the left 20k ratings for test. The feature number $p$ is set as 10.

## 3.2 The Selection of regularization parameter $\lambda$

We evaluate the algorithm performance with a list value of regularization parameter $\lambda$, $[0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 2.56, 5.12, 10.24, 20.48, 40.96, 81.92]$, and choice the one with the low-

[3] http://grouplens.org/datasets/movielens/

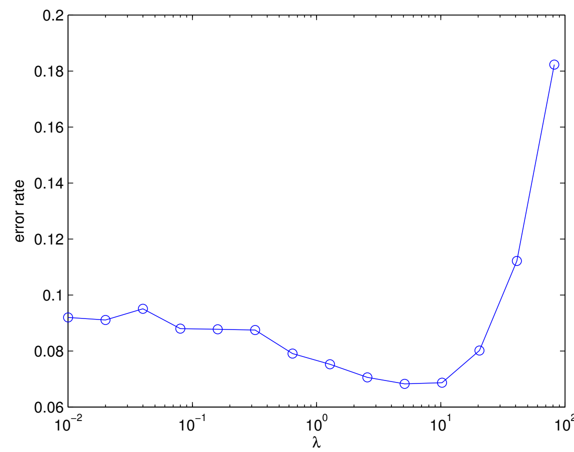


Figure 1: The error rate versus regularization parameter.

est error rate. The error rate is defined as

$$error\_rate = \frac{\|R_{test} \cdot (P - Y_{test})\|_F^2}{\|Y_{test}\|_F^2}, \tag{4}$$

where $R_{test}$ and $Y_{test}$ are the responding test rating matrices. As shown in Figure 1, we select $\lambda = 10$ in the following simulation.

### 3.3 Prediction of ratings

We set the number of feature as 10, and $\lambda = 10$. The program is started from the script file "main.m", and the main function "mf_resys_func" (perform the gradient descent for matrix factorization) is called in this script. The final result is written in the "pred_ratings.txt" in the required format.

## References

[1] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[2] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.