Recruitment Analytics: Predict Applicant Behavior from Sparsely Labeled

1 Introduction

2 Formulation

2.1 Optimization

2.1.1 Classic Matrix Completion

The latent factor model has an underlying assumption: the real exact rating matrix are low-rank matrix. By modelling target rating variable as

$$R_{ij} = U_i^T V_j \tag{1}$$

Then we are able to use least square cost to guide the optimization procedure:

$$\min_{U,V} \quad \sum_{(i,j)} (A_{ij} - U_i^T V_j)^2 + \frac{\lambda}{2}(||U||_F^2 + ||V||_F^2) \tag{2}$$

where $A_{i,j}$ is the observed ground-truth entry and Ω is the index set of entries that are observed.

2.1.2 One-Class Matrix Completion

$$\min_{U,V} \quad \sum_{(i,j)\in\Omega} (1 - U_i^T V_j)^2 + \frac{\lambda}{2}(||U||_F^2 + ||V||_F^2) \tag{3}$$

2.1.3 Inductive Matrix Completion

Formulate the problem as that of recovering a low-rank matrix W_* using observed entries $R_{ij} = x_i^T W_* y_j$ and the user/job feature vectors x_i, y_j. By factoring $W = UV^T$, we see that this scheme constitutes a bi-linear prediction $(x^T U_*)(V_* y)$ for a new user/job pair (x, y).

$$\min_{U,V} \quad \sum_{(i,j)} (A_{ij} - x_i^T U V^T y_j)^2 + \frac{\lambda}{2}(||U||_F^2 + ||V||_F^2) \tag{4}$$

According to [1], under standard set of assumptions, the alternating minimization provably converges at a linear rate to the global optimum of two low-rank estimation problems: a) RIP measurements based general low-rank matrix sensing, and b) low-rank matrix completion. A more recent paper [2] in bioinformatics demonstrated successful application of such inductive matrix completion framework on gene-disease analytics.

2.1.4 One-Class Inductive Matrix Completion

$$\min_{U,V} \quad \sum_{(i,j)\in\Omega} (1 - x_i^T U V^T y_j)^2 + \frac{\lambda}{2}(||U||_F^2 + ||V||_F^2) \tag{5}$$

2.1.5 One-Class Inductive Matrix Completion with Biases

$$\min_{U,V} \quad (1-\alpha) \sum_{A_{ij}=1} (1 - x_i^T U V^T y_j)^2 + \alpha \sum_{A_{ij}=0} (0 - x_i^T U V^T y_j)^2 + \frac{\lambda}{2}(||U||_F^2 + ||V||_F^2) \tag{6}$$

2.2 Assessments

In this research project, we employ Recall-Precision curve and MAP@N curve to asess and visually present the accurate performance of a particular model.

2.2.1 Recall-Precision Curve

In pattern recognition and information retrieval with binary classification, precision, a.k.a. positive predictive value, is the fraction of retrieved instances that are relevant, while recall, a.k.a. sensitivity, is the fraction of relevant instances that are retrieved. According to the general definition, the precision and recall in this recommendation problem can be formulated as

$$\text{precision} = \tag{7}$$

$$\text{recall} = \tag{8}$$

Nevertheless, precision and recall are single-value metrics that take no ordering information into account. Thus, the recall-precision curve cannot assess performance of a model from the "order-does-matter" perspective.

2.2.2 MAP@N Curve

For job recommendation system, it is supposed to return a ranked sequence of suitable jobs for users to apply. Therefore, it is desirable to also consider the order in which the returned jobs are presented. The mean average precision (MAP) does vary in terms of the order of returned jobs and thus cumulative MAP@N curve can serve as an effective tool to visualize the quality of queried recommendations for both ordering and accuracy.

The prerequisite of computing mean average precision is to figure out average precision (AP) for each queried user. The average precision at certain position N is defined as

$$\text{AP@N} = \tag{9}$$

Taking the mean of average precision at the corresponding position for all queried users derives the mean average precision .

$$\text{MAP@N} = \tag{10}$$

3 Implementation

3.1 Algorithm

The algorithmic description for *One-class Inductive Matrix Completion with biased weight* are shown in Algorithm 1.

3.2 Practical Issues

Bias. Bias parameter $\alpha \in (0, 1)$ is the weight to balance optimization objective between observed and missing labels. Note that we can simply set $\alpha = 1$ to solve unbiased version of one-class inductive matrix completion.

Initialization. Randomizing all components of $U_{(0)}$ and $V_{(0)}$ to uniform distribution over $(0, 1)$ works well in our experimentation.

Optimization. During each stage of alternating minimization, standarded conjugate gradient method is employed to solve least square problem.

Algorithm 1 Alternating Minimization for One-Class Inductive Matrix Completion with Biases

INPUT:
a) sparse matrices X and Y that denote features of users and jobs respectively.
b) matrix A that denotes partial observation of "apply" association with observed index set Ω

Initialize $U_{(0)}$ and V_0 by uniform randomization
Do
$V_{(k+1)}$ = argmin $(1-\alpha)\sum_{(i,j)\in\Omega}(1-\boldsymbol{x}_i^T U_{(k)} V_{(k)}^T \boldsymbol{y}_j)^2 + \alpha\sum_{(i,j)\notin\Omega}(0-\boldsymbol{x}_i^T U_{(k)} V_{(k)}^T \boldsymbol{y}_j)^2$
$U_{(k+1)}$ = argmin $(1-\alpha)\sum_{(i,j)\in\Omega}(1-\boldsymbol{x}_i^T U_{(k)} V_{(k+1)}^T \boldsymbol{y}_j)^2 + \alpha\sum_{(i,j)\notin\Omega}(0-\boldsymbol{x}_i^T U_{(k)} V_{(k+1)}^T \boldsymbol{y}_j)^2$
Until Convergence.

OUTPUT:
a) Model Parameter U_* and V_*

Algorithm 2 Newton Conjugate Descent Subroutine

INPUT:
a)
b)

Initialize $U_{(0)}$ and V_0 by uniform randomization

OUTPUT:
a) Model Parameter U_* and V_*

Algorithm 3 Prediction

INPUT:
a) sparse matrices X and Y denote features of users and jobs.
b) matrix A denotes partial observation of application association with observed index set Ω

Initialize $U_{(0)}$ and V_0 by uniform randomization

OUTPUT:
a) Model Parameter U_* and V_*

Algorithm 4 Preclustering

INPUT:
a) sparse matrices X and Y that denote features of users and jobs respectively.
b) matrix A that denotes partial observation of "apply" association with observed index set Ω

Initialize $U_{(0)}$ and V_0 by uniform randomization
Do
$V_{(k+1)}$ = argmin $(1-\alpha)\sum_{(i,j)\in\Omega}(1-\boldsymbol{x}_i^T U_{(k)} V_{(k)}^T \boldsymbol{y}_j)^2 + \alpha\sum_{(i,j)\notin\Omega}(0-\boldsymbol{x}_i^T U_{(k)} V_{(k)}^T \boldsymbol{y}_j)^2$
$U_{(k+1)}$ = argmin $(1-\alpha)\sum_{(i,j)\in\Omega}(1-\boldsymbol{x}_i^T U_{(k)} V_{(k+1)}^T \boldsymbol{y}_j)^2 + \alpha\sum_{(i,j)\notin\Omega}(0-\boldsymbol{x}_i^T U_{(k)} V_{(k+1)}^T \boldsymbol{y}_j)^2$
Until Convergence.

OUTPUT:
a) Model Parameter U_* and V_*

4 Experiment I: Feature Extraction

4.1 Dataset

4.2 Preprocessing

5 Experiment II: Basic Parameters

5.1 Effects of Rank K

We trace the main term in training stage.

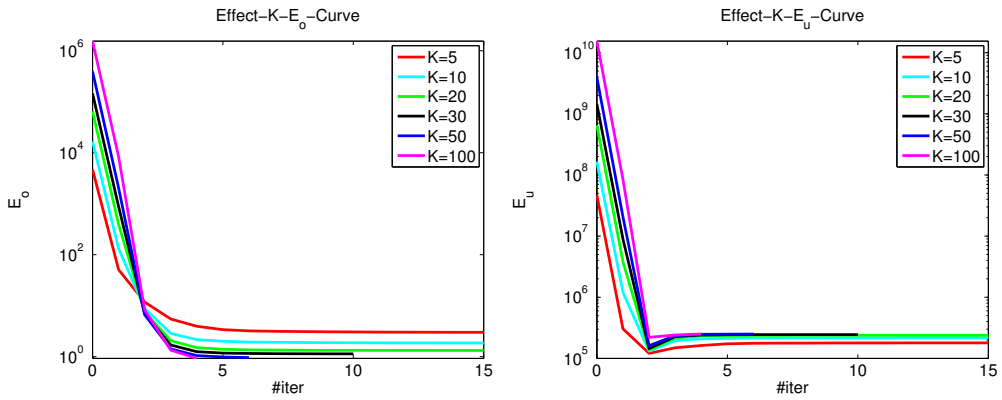

Figure 1: Trace E_o and E_u between the trained models with various $K = 5, 10, 20, 30, 50$. Left: E_o Curve. Right: E_u Curve.

The performance evaluation is as follows.

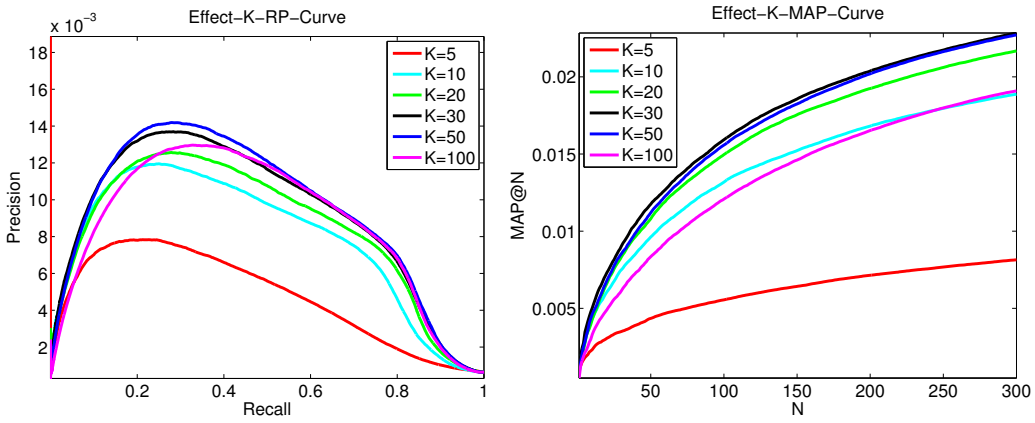

Figure 2: Performance Comparison between the trained models with various $K = 5, 10, 20, 30, 50$. Left: Precision-Recall Curve. Right: MAP@N Curve.

5.2 Effects of Bias Weight α

5.3 Effects of Regularization Parameter λ

5.4 Effects of Early Stopping

6 Experiment III: Revisits Feature Engineering

6.1 Best-Tuned Setting

6.2 Principal Component analysis

7 Experiment IV: Pre-clustered Matrix Completion

7.1 Results

Figure 3: Precision v.s. Recall Comparison between models with $\lambda = \lambda_0, \lambda_1, \lambda_2, \lambda_3$

Figure 4: Precision v.s. Recall Comparison between various models with different preprocessed data

8 Conclusions

Features do provide supervision for recognizing the pattern of new users.

Future work can be extended to modeling time-series recommendation problem by means of one-class inductive matrix completion.

References

[1] Prateek Jain and Inderjit S Dhillon. Provable inductive matrix completion. *arXiv preprint arXiv:1306.0626*, 2013. 1

[2] Nagarajan Natarajan and Inderjit S Dhillon. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12):i60–i68, 2014. 1