Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Optimal Discrete Matrix Completion

Zhouyuan Huo[1], **Ji Liu**[2], **Heng Huang**[1*]
[1]Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, 76019, USA
[2]Department of Computer Science, University of Rochester, Rochester, NY, 14627, USA
huozhouyuan@gmail.com, jliu@cs.rochester.edu, heng@uta.edu

Abstract

In recent years, matrix completion methods have been successfully applied to solve recommender system applications. Most of them focus on the matrix completion problem in real number domain, and produce continuous prediction values. However, these methods are not appropriate in some occasions where the entries of matrix are discrete values, such as movie ratings prediction, social network relation and interaction prediction, because their continuous outputs are not probabilities and uninterpretable. In this case, an additional step to process the continuous results with either heuristic threshold parameters or complicated mapping is necessary, while it is inefficient and may diverge from the optimal solution. There are a few matrix completion methods working on discrete number domain, however, they are not applicable to sparse and large-scale data set. In this paper, we propose a novel optimal discrete matrix completion model, which is able to learn optimal thresholds automatically and also guarantees an exact low-rank structure of the target matrix. We use stochastic gradient descent algorithm with momentum method to optimize the new objective function and speed up optimization. In the experiments, it is proved that our method can predict discrete values with high accuracy, very close to or even better than these values obtained by carefully tuned thresholds on Movielens and YouTube data sets. Meanwhile, our model is able to handle online data and easy to parallelize.

Introduction

In this era of Internet, people spend much more time and energy on the internet. We watch movies and series online through YouTube or Netflix, make friends and maintain relationships online through Facebook or Twitter, place orders and purchase products online through Yelp or Amazon, and even meet and work online through Skype or Github. Our traces and preferences performed on the internet are precious information and resource for these websites to improve user experience and offer customized service. However, these data are always extremely sparse and new algorithms are needed to process them effectively. For example, in Netflix, only a small part of users tend to rate on the movies or series they have watched, and each user just watches and rates a few movies or series compared to the number of items in the whole video database.

As with the example above, our task is predicting missing elements based on a little information already known, and it can be thought as a matrix completion problem. In matrix completion problem, we need to infill a sparse matrix, when only a few entries are observed. There are many methods proposed to solve this problem, including SVD (Billsus and Pazzani 1998), SVT (Cai, Candès, and Shen 2010), Rank-k Matrix Recovery (Huang et al. 2013) and so on. All of these methods hold the same assumption that the approximation matrix has a low-rank structure.

These low-rank matrix approximation methods can be used to solve matrix completion problems like Netflix movie ratings prediction, and their outputs are in real number domain. However, continuous outputs are hard to interpret sometimes, when the inputs of this problem are discrete values. For example, we want to know whether two users are connected or not, 1 denotes connected and 0 means not connected. A decimal number between 0 and 1, like 0.6 (which is not a real probability), can make us confused and is hard to interpret. The most intuitive way is to find a threshold and project these continuous values to discrete ones. However, this method is time consuming and may destroy the low-rank structure of output matrix. There are methods, *e.g.* Robust Discrete Matrix Completion (Huang, Nie, and Huang 2013) solving matrix completion problem in discrete number domain. This method has been proved to be more effective than those general matrix completion methods in the case where the entries of the matrix are discrete values. However, there are still two problems. Firstly, this method needs to go through all the entries and discrete number domain in each iteration, which makes it hard to process big data. Secondly, solution via trace norm minimization may not approximate the rank minimization well.

In this paper, we propose a new optimal discrete matrix completion algorithm to solve matrix completion problem in discrete number domain. We introduce new threshold variables such that we can integrate continuous matrix completion and threshold learning in the same loss function. We provide a novel error estimation loss for discrete matrix completion to learn optimal thresholds. Instead of min-

*To whom all correspondence should be addressed. This work was partially supported by NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NIH R01 AG049371 at UTA and NSF-CNS 1548078 at UR.

imizing trace norm, we recover the matrix with exact rank-k, and limits the parameter tuning within a set of integers instead of infinite possible values. We also use stochastic gradient descent optimization algorithm to solve our proposed new objective function. Our new algorithm can be used in online applications, and only two feature vectors are to be updated when a new entry comes. Meanwhile, there are many parallel stochastic gradient descent algorithms, so our model is easy to be parallelized. We conduct experiments on real Movielens and YouTube data sets. The empirical results show that our method outperforms seven other compared methods in most cases with tuning procedure.

Related Work

In this paper, we use $M \in \mathbb{R}^{n\times m}$ to represent a sparse matrix which contains missing values, and Ω to represent the positions of entries which are already known. Our task is to predict unknown entries by making use of its intrinsic structure and information. Low-rank structure has been widely used to solve matrix completion problem, and the standard formulation can be represented as,

$$\min_X rank(X) \;\; s.t. \; X_{ij} = M_{ij}, (i,j) \in \Omega\,. \tag{1}$$

Even though it is easy to form and understand this formulation, it's very hard to optimize. This is an NP-hard problem and any algorithm to compute an exact solution needs exponential time complexity (Woeginger 2003). To solve this problem, researchers (Cai, Candès, and Shen 2010; Candès and Recht 2009) proposed to use trace norm as a convex approximation of the low-rank structure of a matrix. It is also proved that under some specific conditions we can perfectly recover most low-rank matrices from what appears to be an incomplete set of entries. Thus, the problem (1) is often alternatively formulated as:

$$\min_X ||X_\Omega - M_\Omega||_F^2 + \gamma ||X||_*\,, \tag{2}$$

where $||X||_*$ denotes the trace norm (nuclear norm) of matrix X. $||X||_* = \sum_i \sigma_i(X)$ and $\sigma_i(X)$ is a singular value of X. γ denotes the regularization parameter, and it is used to balance the bias of predicted matrix and its low-rank structure. An optimal continuous solution can be obtained by optimizing problem (2).

In recent years, many algorithms were proposed to solve the trace norm based matrix completion problems (Srebro, Rennie, and Jaakkola 2004; Wright et al. 2009; Koltchinskii et al. 2011; Ji et al. 2010; Keshavan, Montanari, and Oh 2009; Nie, Huang, and Ding 2012; Nie et al. 2012; 2014). In (Huang et al. 2013), they proposed to constrain the rank of the matrix explicitly and to seek a matrix with an exact rank k. In this way, it is guaranteed that the rank of matrix is in a specific range. These algorithms can solve continuous matrix completion problem appropriately. However, in many applications, we need discrete results. An additional step is needed to project continuous values to discrete number domain. Tuning threshold intuitively (especially matrix completion is an unsupervised learning task) and projecting continuous values to discrete values may spoil the low-rank structure and lead to suboptimal results.

To tackle discrete matrix completion problem, a few methods were introduced recently. In (Huang, Nie, and Huang 2013), authors proposed to use the ℓ_1 norm as loss function and explicitly imposes the discrete constraints on prediction values in the process of matrix completion. Their objective function is as follows:

$$\min_X ||X_\Omega - M_\Omega||_1 + \gamma ||X||_* \;\; s.t. \; X_{ij} \in \mathcal{D}\,, \tag{3}$$

where $\mathcal{D} = \{c_1, ..., c_k\}$. However, it has to choose the best discrete value for each entry one by one during the optimization, which makes the time complexity of this method too large to work on big data. It's also known that predicted matrix via trace norm is not guaranteed to be good approximation of rank minimization. To overcome these challenging problems, we propose a novel optimal discrete matrix completion model to automatically and explicitly learn thresholds.

Optimal Discrete Matrix Completion

It is difficult to combine continuous matrix completion and threshold learning in the same loss function. To address this challenging problem, we introduce new threshold variables. Without loss of generality, we assume that all entries of the discrete matrix M come from set $\{1, 2, 3, \cdots, s\}$, where s is the largest discrete number in a specific application. Because the thresholds between different two continuous discrete values may not be the same, we assume the threshold variables as $\mathbf{d} = \{d_{0,1}, d_{1,2}, \cdots, d_{s-1,s}, d_{s,s+1}\}$, where $d_{t,t+1}$ denotes the threshold variable between t and $t+1$. For $x \in (t, t+1)$, if $x - t \le d_{t,t+1}$, then we round x to t, otherwise, $t+1$. We are going to design a new matrix completion model to find the prediction matrix X and optimal thresholds $\mathbf{d}$.

Firstly, we define a new penalty term of estimation error between element M_{ij} and predicted value X_{ij} as:

$$\begin{aligned} f(X_{ij}) = \;& \max(X_{ij} - M_{ij} - d_{M_{ij},M_{ij}+1}, 0)^2 \\ & + \max(-X_{ij} + M_{ij} - (1 - d_{M_{ij}-1,M_{ij}}), 0)^2\,. \end{aligned} \tag{4}$$

If $X_{ij} \in \left[M_{ij} - 1 + d_{M_{ij}-1,M_{ij}}, M_{ij} + d_{M_{ij},M_{ij}+1}\right]$, both terms are 0 so that final penalty $f(X_{ij}) = 0$. If $X_{ij} \in (-\infty, M_{ij} - 1 + d_{M_{ij}-1,M_{ij}})$, the first term is 0 and $f(X_{ij}) = (-X_{ij} + M_{ij} - 1 + d_{M_{ij}-1,M_{ij}})^2$; if $X_{ij} \in (M_{ij} + d_{M_{ij},M_{ij}+1}, +\infty)$, the second term is 0 and $f(X_{ij}) = (X_{ij} - M_{ij} - d_{M_{ij},M_{ij}+1})^2$. Thus, our new penalty term can calculate the estimation error between observed value and predicted value.

Based on our new penalty loss, we propose the following objective function for discrete matrix completion:

$$\begin{aligned} \min_{U,V,\mathbf{d}} \;\; & \tfrac{1}{2} \sum_{(i,j)\in\Omega} \max(U_i^T V_j - M_{ij} - d_{M_{ij},M_{ij}+1}, 0)^2 \\ & + \max(-U_i^T V_j + M_{ij} - 1 + d_{M_{ij}-1,M_{ij}}, 0)^2 \\ & + \tfrac{1}{2}\gamma(||U||_F^2 + ||V||_F^2) + \tfrac{1}{2}\eta \sum_{t=0}^{s} |d_{t,t+1} - \tfrac{1}{2}|^2 \\ s.t. \;\; & U \in \mathbb{R}^{r\times n}, V \in \mathbb{R}^{r\times m}, r < \min(m,n), \\ & 0 \le d_{0,1}, d_{1,2}, ..., d_{s-1,s}, d_{s,s+1} \le 1\,. \end{aligned} \tag{5}$$

In our new objective function , $U \in \mathbb{R}^{r\times n}$ and $V \in \mathbb{R}^{r\times m}$ make final predicted matrix $X = UV$ to be a low-rank matrix with rank r, and U_i, V_j are column vectors of U and V respectively. Thus, our new objective function utilizes rank-k minimization to approximate rank minimization problem. Compared to trace norm, the rank-k minimization explicitly imposes low-rank structure can approximate rank minimization better. The regularization term $\frac{1}{2}\gamma(||U||_F^2 + ||V||_F^2)$ is used to avoid overfitting by penalizing the magnitudes of the parameters. The regularization term $\frac{1}{2}\eta \sum_{t=0}^{s} |d_{t,t+1} - \frac{1}{2}|^2$ is used to constrain $d_{t,t+1}$ around $\frac{1}{2}$, which is considered as the prior of variables $d_{t,t+1}$. Therefore, our new discrete matrix completion model can predict missing values and learn optimal thresholds simultaneously.

Optimization Algorithm

In order to solve the large-scale discrete matrix completion problem, we use stochastic gradient descent algorithm to optimize our new objective function in problem (5). It receives one entry every step, and updates corresponding vector in U and V, so that our algorithm is applicable to handle big data, such as Netflix data or Yahoo Music data (Dror et al. 2012). The number of variables is just $(m + n)r + s + 1$ much smaller than $mn + s + 1$, thus this formulation can handle large-scale data easily. Meanwhile, stochastic gradient descent algorithm is also easy to parallelize (Recht et al. 2011; Zhuang et al. 2013). For every entry M_{ij}, problem (5) becomes:

$$\begin{array}{ll}\min\limits_{U,V,d} & \frac{1}{2}\max(U_i^T V_j - M_{ij} - d_{M_{ij},M_{ij}+1}, 0)^2 \\ & +\frac{1}{2}\max\left(-U_i^T V_j + M_{ij} - 1 + d_{M_{ij-1},M_{ij}}, 0\right)^2 \\ & +\frac{1}{2}\gamma||U_i||_F^2 + \frac{1}{2}\gamma||V_j||_F^2 \\ & +\frac{1}{2}\eta\left(d_{M_{ij},M_{ij}+1} - \frac{1}{2}\right)^2 + \frac{1}{2}\eta\left(d_{M_{ij}-1,M_{ij}} - \frac{1}{2}\right)^2 \\ s.t. & 0 \le d_{M_{ij},M_{ij}+1}, d_{M_{ij}-1,M_{ij}} \le 1\end{array} \tag{6}$$

According to the stochastic gradient descent strategy, for every entry M_{ij}, we update corresponding U_i, V_j, $d_{t,t+1}$ respectively.

$$U_i = U_i - \mu\frac{\partial l(U_i)}{\partial U_i},\ V_j = V_j - \mu\frac{\partial l(V_j)}{\partial V_j} \tag{7}$$

$$d_{M_{ij},M_{ij}+1} = d_{M_{ij},M_{ij}+1} - \mu\frac{\partial l(d_{M_{ij},M_{ij}+1})}{\partial d_{M_{ij},M_{ij}+1}} \tag{8}$$

$$d_{M_{ij}-1,M_{ij}} = d_{M_{ij}-1,M_{ij}} - \mu\frac{\partial l(d_{M_{ij}-1,M_{ij}})}{\partial d_{M_{ij}-1,M_{ij}}} \tag{9}$$

where μ is learning rate. The first order derivatives over each term are as follows:

$$\begin{aligned}\frac{\partial l(U_i)}{\partial U_i} = \ & \max(U_i^T V_j - M_{ij} - d_{M_{ij},M_{ij+1}}, 0)V_j - \\ & \max(-U_i^T V_j + M_{ij} - 1 + d_{M_{ij-1},M_{ij}}, 0)V_j + \gamma U_i\end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\partial l(V_j)}{\partial V_j} = \ & \max(U_i^T V_j - M_{ij} - d_{M_{ij},M_{ij+1}}, 0)U_i - \\ & \max(-U_i^T V_j + M_{ij} - 1 + d_{M_{ij-1},M_{ij}}, 0)U_i + \gamma V_j\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{\partial l(d_{M_{ij},M_{ij}+1})}{\partial d_{M_{ij},M_{ij}+1}} &= -\max(U_i^T V_j - M_{ij} - d_{M_{ij},M_{ij}+1}, 0) \\ & \quad +\eta(d_{M_{ij},M_{ij}+1} - \frac{1}{2})\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{\partial l(d_{M_{ij}-1,M_{ij}})}{\partial d_{M_{ij}-1,M_{ij}}} &= \max(U_i^T V_j - M_{ij} - 1 + d_{M_{ij}-1,M_{ij}}, 0) \\ & \quad +\eta(d_{M_{ij}-1,M_{ij}} - \frac{1}{2})\,.\end{aligned} \tag{13}$$

Because $d_{t,t+1} \in [0, 1]$, $d_{t,t+1}$ is projected as:

$$d_{t,t+1} = \begin{cases}1 & \text{if} \quad d_{t,t+1} > 1 \\ d_{t,t+1} & \text{if} \quad 0 \le d_{t,t+1} \le 1 \\ 0 & \text{if} \quad d_{t,t+1} < 0\end{cases} \tag{14}$$

In the optimization procedure of stochastic gradient descent, there exists a trade-off between quick convergence and descent step size, and it is determined by learning rate μ. If learning rate μ is very small, the convergence of objective function value is guaranteed, while it is going to take a long time to converge. On the other hand, if μ is large, the objective function value is very likely to diverge. In this experiment, we use $\mu = \frac{\mu_0}{k^\alpha}$, where μ_0 is learned through small data set (Bottou 2010), and k means the number of iterations, $\alpha = 0.1$ in the experiment. Besides, the stochastic gradient descent algorithm is easy to converge to a local optimum. In our experiments, we use momentum method to avoid the local optimum, which is a commonly used implementation.

To sum up, the whole procedure to solve problem (5) is described in Algorithm (1). It is easy to observe that for each step, time complexity of our algorithm is just $O(r)$. Considering the iteration numbers k, the total time complexity is $O(kr)$. As we know, the time complexity of SVD is $O(nm^2)$. When matrix is large-scale, our algorithm is much faster than SVD. Moreover, because we only need one entry for every iteration, our algorithm can be naturally used in online occasion. There are also many methods to parallelize stochastic gradient descent algorithms, *e.g.* HOGWILD! (Recht et al. 2011) and parallelized stochastic gradient descent (Zinkevich et al. 2010).

Algorithm 1 Optimal Discrete Matrix Completion

Input: $M, \Omega \in \mathbb{R}^{n\times m}$
Output: $U \in \mathbb{R}^{r\times n}, V \in \mathbb{R}^{r\times m}, d_{t,t+1}$
Set: Regularization parameters: γ, η, Learning rate: μ
for $(i, j) \in \Omega$ **do**
  Update U_i via Eqs. (7) and (10).
  Update V_j via Eqs. (7) and (11).
  Update $d_{t,t+1}$ via Eqs. (9), (12), (13) and (14).
end for

Experimental Results

In this section, we apply our optimal discrete matrix completion method (ODMC) to two real world data sets: MovieLens and YouTube data sets. Both of these two data sets are

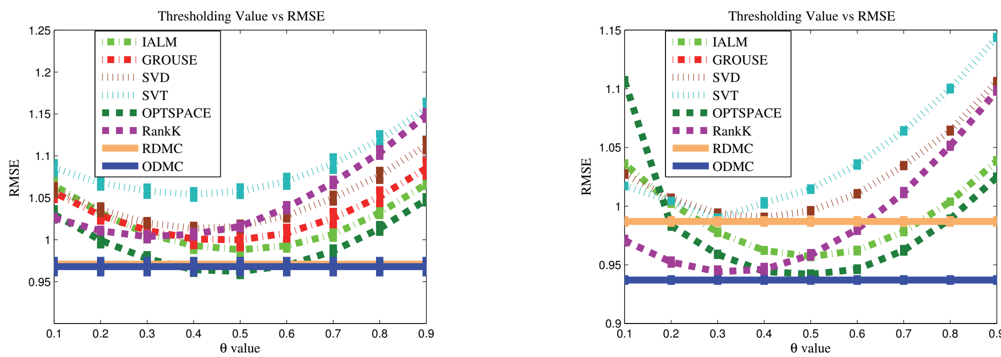
MovieLens 100k

Methods	RMSE	MAE
SVD	1.0134 ± 0.0039	0.7122 ± 0.0045
SVT	1.0540 ± 0.0047	0.7411 ± 0.0039
IALM	0.9881 ± 0.0051	0.7042 ± 0.0043
GROUSE	0.9998 ± 0.0096	0.7055 ± 0.0084
RankK	1.0037 ± 0.0039	0.7055 ± 0.0043
OPTSPACE	$\mathbf{0.9622 \pm 0.0052}$	$\mathbf{0.6794 \pm 0.0045}$
RDMC	0.9709 ± 0.0075	0.7119 ± 0.0082
ODMC	$\mathbf{0.9679 \pm 0.0076}$	$\mathbf{0.7033 \pm 0.0051}$

MovieLens 1M

Methods	RMSE	MAE
SVD	0.9905 ± 0.0011	0.6869 ± 0.0011
SVT	0.9997 ± 0.0013	0.6866 ± 0.0013
IALM	0.9571 ± 0.0011	0.6799 ± 0.0007
GROUSE	-	-
RankK	0.9442 ± 0.0024	$\mathbf{0.6559 \pm 0.0019}$
OPTSPACE	$\mathbf{0.9402 \pm 0.0012}$	0.6645 ± 0.0009
RDMC	0.9870 ± 0.0016	0.6702 ± 0.0016
ODMC	$\mathbf{0.9371 \pm 0.0012}$	$\mathbf{0.6583 \pm 0.0014}$

Table 1: MovieLens Data Set



(a) Movielens 100k

(b) Movielens 1M

Figure 1: Movielens Data Rating Prediction.

in discrete domain. In the experiment, there are seven other compared methods in total, including SVD, Singular Value Thresholding (SVT) (Cai, Candès, and Shen 2010), Inexact Augmented Lagrange Multiplier method (IALM) (Lin, Chen, and Ma 2010), Grassmannian Rank-One Update Subspace Estimation (GROUSE) (Balzano, Nowak, and Recht 2010), OPTSPACE (Keshavan and Oh 2009), Rank-k Matrix Recovery (RankK) (Huang et al. 2013) and Robust Discrete Matrix Completion (RDMC) (Huang, Nie, and Huang 2013).

Experiment Setup

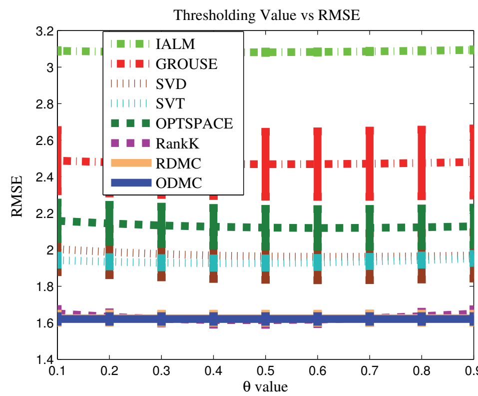
For SVT, RDMC, RankK, RDMC methods, we use a list of $\{0.01, 0.1, 1, 10, 100\}$ to tune the best parameters. For SVD, GROUSE, OPTSPACE and ODMC, an exact low-rank value should be set, and we use $\{5, 10, 15, 20, 25\}$ to tune the best rank approximation value for different matrix in the experiments. At first, we randomly hide most of ground truth data in the experiments, so that no more than 10% data are known. In the experiments, all the entries of experiment data sets are discrete values, so after fitting process, for methods SVD, SVT, RankK, IALM, GROUSE, OPTSPACE, an additional threshold tuning process is needed. We tune thresholds θ from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and select one with best performance. For each method, we run the same process 5 times and take the average as final results. In the experiment, two widely used metrics are used to evaluate these methods, namely Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Rating Prediction on MovieLens Data Sets

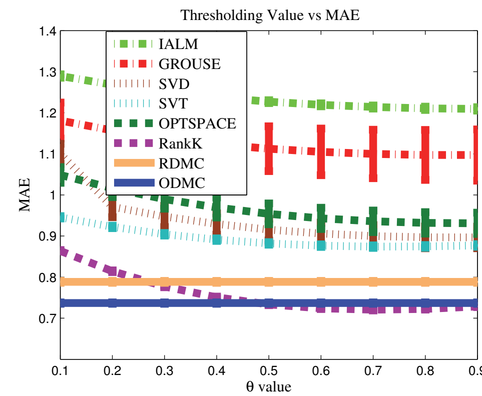
MovieLens rating data are collected from MovieLens website: https://movielens.org/. This data set are collected over various periods of time, depending on the size of the set. In the experiments, we use two data sets, MovieLens 100k and MovieLens 1M, respectively. For MovieLens 100k data set, it consists 100,000 ratings from 943 users and 1,682 movies and each user has rated at least 20 movies. In this data set, every entry is from discrete number domain $\{1, 2, 3, 4, 5\}$. This data set was collected through MovieLens website during the seven-month period from September 19th, 1997 through April 22nd, 1998. MovieLens 1M data set contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000. Each entry is in discrete number domain $\{1, 2, 3, 4, 5\}$. Please check more details at http://grouplens.org/.

In MovieLens 100k data set, about 6% entries of the matrix are rated, and 4% rated entries in Movielens 1M. We run the same procedure 5 times and take the average as final performance. Every time, we hold 75% of rated entries as observed training data, and the other 25% data as testing data.

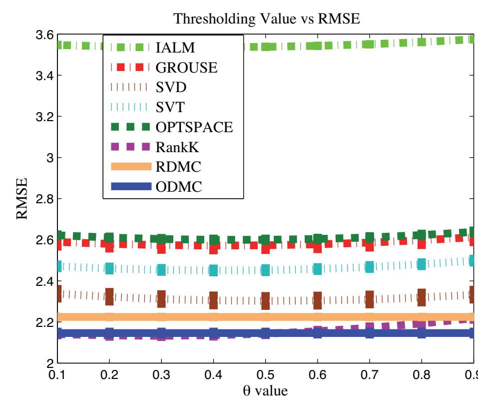
From Table 1, we can observe that our ODMC method works best on RMSE metric in MovieLens 1M, and is very close to the best method on MAE metric. The accuracies of



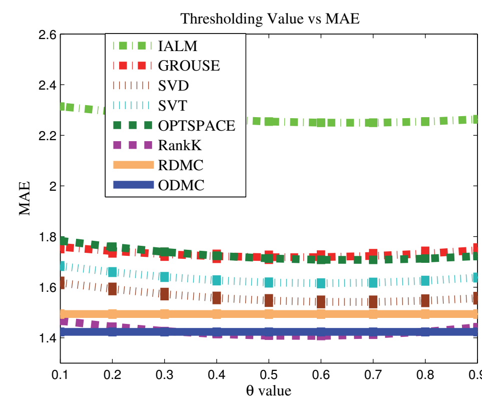
(a) Shared Subscribers Network



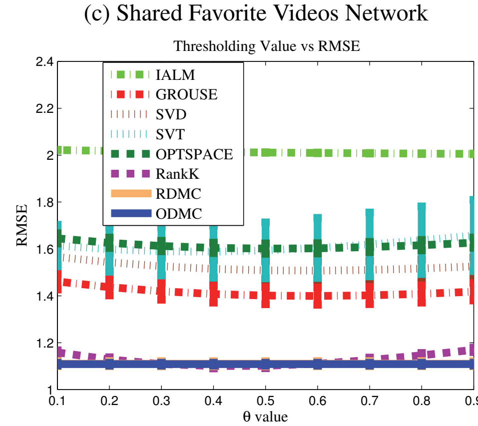
(b) Shared Subscribers Network



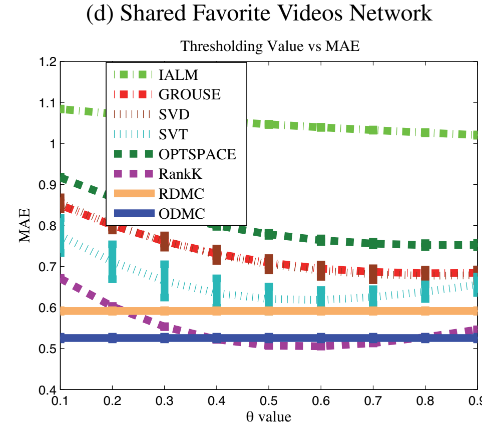
(c) Shared Favorite Videos Network



(d) Shared Favorite Videos Network



(e) Shared Subscriptions Network



(f) Shared Subscriptions Network

Figure 2: YouTube Data Relation Prediction

compared methods often degenerate after we tune thresholds and project these outputs to discrete number domain. However, our ODMC combines the threshold tuning process and low-rank matrix approximation process, and guarantees that our objective function value converges to a local optimum. In this table, it is also obvious that our model works better than other continuous matrix completion method except OPTSPACE or RankK in some cases.

Figure 1 presents the RMSE metric performance of all these methods on two MovieLens data sets. It is easy to observe that our ODMC method is reliable in the MovieLens rating prediction problem in Figure 1. Different from the performance of continuous matrix completion methods that fluctuate greatly when we use different thresholds, our method's result is consistent all the time (our method needn't tune the thresholds) and outperforms RDMC, the other discrete matrix completion method. There are two straight lines in this figure, which represent the outputs of RDMC and ODMC methods. Both of them are discrete matrix completion methods, so that their outputs are consistent under different thresholds. From Figure 1a, we can see that the performance of these two methods are nearly the same, while in Figure 1b, our ODMC method has significant superiority over RDMC method.

Shared Subscribers Network

Methods	RMSE	MAE
SVD	1.9612 ± 0.1283	0.8966 ± 0.0262
SVT	1.9266 ± 0.0254	0.8742 ± 0.0032
IALM	3.0813 ± 0.0039	1.2094 ± 0.0042
GROUSE	2.4682 ± 0.1777	1.0972 ± 0.0624
RankK	**1.6137 ± 0.0250**	**0.7206 ± 0.0017**
OPTSPACE	2.1190 ± 0.1035	0.9309 ± 0.0529
RDMC	1.6248 ± 0.0248	0.7884 ± 0.0023
ODMC	**1.6206 ± 0.0155**	**0.7368 ± 0.0018**

Shared Favorite Videos Network

Methods	RMSE	MAE
SVD	2.3024 ± 0.0245	1.5417 ± 0.0100
SVT	2.4505 ± 0.0047	1.7411 ± 0.0039
IALM	3.5360 ± 0.0016	2.2498 ± 0.0020
GROUSE	2.2716 ± 0.0235	1.7183 ± 0.0113
RankK	**2.1328 ± 0.0066**	**1.4071 ± 0.0013**
OPTSPACE	2.5997 ± 0.0061	1.7079 ± 0.0016
RDMC	2.2244 ± 0.0019	1.4939 ± 0.0037
ODMC	**2.1463 ± 0.0073**	**1.4233 ± 0.0022**

Shared Subscriptions Network

Methods	RMSE	MAE
SVD	1.5074 ± 0.0391	0.6777 ± 0.0143
SVT	1.5884 ± 0.1113	0.6186 ± 0.0239
IALM	2.0053 ± 0.0015	1.0204 ± 0.0014
GROUSE	1.3994 ± 0.0375	0.6835 ± 0.0059
RankK	**1.1006 ± 0.0101**	**0.5055 ± 0.0006**
OPTSPACE	1.6014 ± 0.0222	0.7522 ± 0.0046
RDMC	1.1106 ± 0.0111	0.5916 ± 0.0017
ODMC	**1.1085 ± 0.0061**	**0.5257 ± 0.0032**

Table 2: YouTube Data Set

Relation Prediction on YouTube Data Sets

YouTube is a video sharing site where various interactions occur among different users. This YouTube data set (Zafarani and Liu 2009), is crawled from YouTube website on 2008, and there are 15,088 user profiles in total and 5 different interactions between these users, including contact network, number of shared friends between two users, number of shared subscriptions between two users, number of shared subscribers between two users and the number of shared favorite videos. In the experiment, we use shared subscibers network, shared favorite videos network, and shared subscriptions network. For each network data, we select 2,000 most active users, so the size of data is $2,000 \times 2,000$. In shared subscribers network data, entry value is in discrete domain ranges from 1 to 326, in shared favorite videos network data, the entry value ranges from 1 to 116, and in shared subscriptions network data, the entry value ranges from 1 to 174. In the experiments, for each data set, we assume 10% of links are known as training data and the others are treated as testing data.

In Table 2, it is easy to see that the outputs of our ODMC method are nearly the same as the best outputs of RankK. Our ODMC method performs better than other continuous matrix completion methods. However, for RankK method, a tedious procedure to tune the threshold is needed. Meanwhile, our model outperforms the other discrete matrix completion model RDMC on both RMSE and MAE metrics.

In Figure 2, the performance of all methods under different thresholds are shown clearly. For SVD, SVT, IALM, GROUSE, RankK and OPTSPACE methods, an additive threshold tuning procedure is required to output discrete values. Different thresholds have significant influence on the final performance of these methods. Obviously, the prediction results of our ODMC method are similar or even better than the best prediction results of these methods which need tedious tuning. In real-world applications, we usually have no enough data to tune these methods to achieve the best results. Thus, our new ODMC method is more suitable for real discrete matrix completion problems.

Two straight lines in Figure 2 represent the outputs of RDMC and ODMC methods, both of them are discrete matrix completion methods. It is clear that our ODMC method outperforms RDMC method consistently.

Conclusion

In this paper, we propose a novel optimal discrete matrix completion method. In this method, we explicitly introduce threshold variables in objective function, so that we can learn optimal threshold variable between any two discrete values automatically. In the optimization, we use stochastic gradient descent algorithm, and for each entry, computation complexity is only $O(r)$. Thus, our method is able to handle online data and large-scale data. Moreover, stochastic gradient descent algorithm is easy to be parallelized. We perform experiments on Movielens data sets and YouTube data sets. Empirical results show that our method outperforms seven other compared methods with threshold tuning procedure in most cases.

References

Balzano, L.; Nowak, R.; and Recht, B. 2010. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 704–711. IEEE.

Billsus, D., and Pazzani, M. J. 1998. Learning collaborative information filters. In *ICML*, volume 98, 46–54.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer. 177–186.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9(6):717–772.

Dror, G.; Koenigstein, N.; Koren, Y.; and Weimer, M. 2012. The yahoo! music dataset and kdd-cup'11. In *KDD Cup*, 8–18.

Huang, J.; Nie, F.; Huang, H.; Lei, Y.; and Ding, C. 2013. Social trust prediction using rank-k matrix recovery. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2647–2653. AAAI Press.

Huang, J.; Nie, F.; and Huang, H. 2013. Robust discrete matrix completion. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Ji, H.; Liu, C.; Shen, Z.; and Xu, Y. 2010. Robust video denoising using low rank matrix completion. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1791–1798. IEEE.

Keshavan, R. H., and Oh, S. 2009. A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*.

Keshavan, R.; Montanari, A.; and Oh, S. 2009. Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems*, 952–960.

Koltchinskii, V.; Lounici, K.; Tsybakov, A. B.; et al. 2011. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics* 39(5):2302–2329.

Lin, Z.; Chen, M.; and Ma, Y. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.

Nie, F.; Wang, H.; Huang, H.; and Ding, C. 2012. Robust matrix completion via joint schatten p-norm and lp-norm minimization. *ICDM* 566–574.

Nie, F.; Wang, H.; Huang, H.; and Ding, C. 2014. Joint schatten p-norm and lp-norm robust matrix completion for missing value recovery. *Knowledge and Information Systems (KAIS)*.

Nie, F.; Huang, H.; and Ding, C. 2012. Schatten-p Norm Minimization for Low-Rank Matrix Recovery. *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2012)* 157.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 693–701.

Srebro, N.; Rennie, J.; and Jaakkola, T. S. 2004. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, 1329–1336.

Woeginger, G. J. 2003. Exact algorithms for np-hard problems: A survey. In *Combinatorial OptimizationEureka, You Shrink!* Springer. 185–207.

Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; and Ma, Y. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, 2080–2088.

Zafarani, R., and Liu, H. 2009. Social computing data repository at ASU.

Zhuang, Y.; Chin, W.-S.; Juan, Y.-C.; and Lin, C.-J. 2013. A fast parallel sgd for matrix factorization in shared memory systems. In *Proceedings of the 7th ACM conference on Recommender systems*, 249–256. ACM.

Zinkevich, M.; Weimer, M.; Li, L.; and Smola, A. J. 2010. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, 2595–2603.