Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Social Trust Prediction Using Rank-k Matrix Recovery

Jin Huang, Feiping Nie, Heng Huang*, Yu Lei, Chris Ding
University of Texas at Arlington
Arlington, TX, 76019
huangjinsuzhou@gmail.com, feipingnie@gmail.com, heng@uta.edu, ylei@cse.uta.edu, chqding@uta.edu

Abstract

Trust prediction, which explores the unobserved relationships between online community users, is an emerging and important research topic in social network analysis and many web applications. Similar to other social-based recommender systems, trust relationships between users can be also modeled in the form of matrices. Recent study shows users generally establish friendship due to a few latent factors, it is therefore reasonable to assume the trust matrices are of low-rank. As a result, many recommendation system strategies can be applied here. In particular, trace norm minimization, which uses matrix's trace norm to approximate its rank, is especially appealing. However, recent articles cast doubts on the validity of trace norm approximation. In this paper, instead of using trace norm minimization, we propose a new robust rank-k matrix completion method, which explicitly seeks a matrix with *exact* rank. Moreover, our method is robust to noise or corrupted observations. We optimize the new objective function in an alternative manner, based on a combination of ancillary variables and Augmented Lagrangian Multiplier (ALM) Method. We perform the experiments on three real-world data sets and all empirical results demonstrate the effectiveness of our method.

1 Introduction

The number of registered users on social network web sites has dramatically increased in recent years. Together with the enrollment increases, more frequent activities happen between users. Users daily receive numerous pieces of content generated by other users, and many of these messages need to be evaluated for trustworthiness, thus the question of whom and what to trust has become an even more important challenge on the web. As a result, users resort to trust information to filter and extract external messages and build connections with other users within the community. One of the key features in these social web sites is to allow users to explicitly express their trust or distrust towards other users in some forms, such as accept others' connection request, block malicious users they dislike. These tags (directed edges), together with these users (nodes), can be represented as a trust graph. Here trust graph is a matrix, if user i trusts user j, then the corresponding matrix entry is 1, otherwise 0. Trust prediction is to predict the missing values in the trust graph and can be viewed as a special case of the general link prediction in social networks. The most important differences between trust graphs and general social network graphs are two structure properties, trust graphs could have symmetry property between certain users and the trust links tend to propagate along users. The symmetry comes from the mutual status of the trust relationship, one user tends to trust the other if the other party grants him/her trust. The trust propagation is also easy to see, two users are easier to become friends and cast trust towards each other given they have a common trusted peer.

Due to the lack of diligence from users' side and privacy concern, missing values in the trust graphs are inevitable. Online users seldom have the time and energy to explicitly tag their friends, the online privacy security meanwhile is becoming a serious issue. Users therefore have very limited number of explicit tags for their online friends. On the other hand, users have the desire to expand their networks and meanwhile maintain their online information and profile security, know whom to trust and whom not to. As a result, it is desirable and often necessary to predict these missing values first to facilitate subsequent mining tasks. There are many algorithms available for general prediction in graphs. In [Liben-Nowell and Kleinberg, 2003], Kleinberg *et al.* surveyed an array of trust prediction methods and classify them into three categories. The first category is based on node neighborhoods, such as Common Neighbors [Newman, 2001] and Jaccard's coefficient [Salton and McGill, 1983]. The second category is based on the ensemble of all paths, well known methods are Katz [Katz, 1953] and SimRank [Jeh and Widom, 2002]. Low-rank approximations by SVD [Billsus and Pazzani, 1998] or by trace norm minimization [Cai *et al.*, 2008] belong to the last category.

Realistic trust graphs carry less than 1% non-zeros. For instance, most Facebook users have less than 1,000 friends, which is far less than 1% considering facebook hundreds of

*Corresponding Author. This work was partially supported by NSF CCF-0830780, CCF-0917274, DMS-0915228, IIS-1117965, and NIST 70NANB9H9178, 70NANB10H168, 70NANB12H175

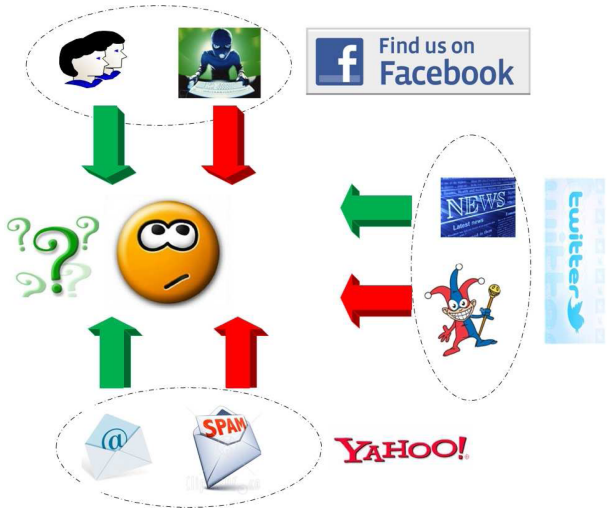


Figure 1: A demonstration for scenarios that users make trust decision in various forms, such as accepting others' adding requests from facebook, believing news from other twitter users, reading emails sent by others. Green arrow represents the sources user should trust and red arrow represents the sources user should distrust.

millions of users. This makes the first two categories difficult in practice as we will show in the experiment section. The focus of this paper is the low rank approximation method. The low rank assumption reflects the fact that social network user can be grouped into a small number of clusters based on their background or interest. Trace norm minimization [Cai *et al.*, 2008], different from classic SVD method, uses convex matrix trace norm to approximate its rank. However, there are two potential issues with this method. First, the incoherence conditions of the data matrix is often too restrictive, there is no prediction accuracy guarantee when the assumption is not satisfied. The theoretical results in [Candès and Recht, 2009; Candès and Tao, 2010] assume that the observed entries are sampled uniformly at random. Unfortunately, many real-world data sets exhibit power-law distributed samples instead [Meka *et al.*, 2009]. Furthermore, Shi et al. [Shi and Yu, 2010] pointed out that the yielded solution via trace norm minimization is often not low-rank or unique for practice applications. Second, the sparse entries are prone to the influence of outlying or corrupted observations. There are some work trying to alleviate this issue in literature [Wright *et al.*, 2009].

Users who trust each other are often in the same social circle. It has been discovered in [McPherson *et al.*, 2001], people who are in the same social circle often share similar behavior and tastes due to social influence and behavior adoption. Therefore, individual user's interests and attributes can be inferred from his or her social neighbors [Bedi *et al.*, 2007; Wen and Lin, 2010]. Furthermore, Huang *et al.* [Huang *et al.*, 2012] discovered it is possible to make accurate personalized recommendation based on trust information. These applications demonstrate the application of trust prediction.

In this paper, we introduce a novel view on robust matrix completion based on low-rank approximation and propose a Robust Rank-k Matrix Completion (RRMC) method. The main contribution of our paper can be summarized as follows.

- Instead of minimizing the trace norm, we recover the data matrix with exact rank-k (k is a low-rank number selected by user). It limits the parameter tuning within a set of integers instead of infinite possible values, which is crucial for real applications.
- Our method explicitly guarantees the low-rank of the recovery matrix and also minimizes the influence of noisy observations.
- We optimize the new objective function in an alternative, iterative way with the adoption of ancillary variables and ALM. Our algorithm has reasonable efficiency and theoretical convergence guarantee.
- Our method is robust to parameter choices and has stable performance.

The rest of the paper is organized as follows: Section 2 presents a brief review on related work and introduces our objective function. Section 3 derives the optimization method and summarizes the algorithm. Empirical experiments on different real data sets are demonstrated in Section 4, and Section 5 concludes the paper.

2 Robust Rank-k Matrix Completion

Given a sufficient number of uniform measurements from the data matrix $M \in \mathbb{R}^{n\times m}$, it is natural to recover the low-rank matrix $\mathbf{X}$ by solving the following equation:

$$\begin{aligned} &\min_{\mathbf{X}} \ rank(\mathbf{X}) \\ &s.t.\ \mathbf{X}_{ij} = \mathbf{M}_{ij},\ (i,j) \in \Omega, \end{aligned} \tag{1}$$

where Ω is the set holding all indices of observed entries in matrix M. Eq. (1) seeks the simplest explantation to fit the observed data. However, there are two fundamental drawbacks for this approach: (1) This optimization problem is NP-hard; (2) Practical data set rarely satisfy the uniqueness of $\mathbf{X}$.

If a matrix has rank r, then clearly it has exactly r nonzero singular values. As a result, the rank function in Eq. (1) is the number of nonzero singular values. In recent years, Candès et al. proposed to use trace norm $\|\mathbf{X}\|_*$ (the sum of the singular values) to seek a low-rank matrix recovery [Candès and Recht, 2009; Candès and Tao, 2010]. It is obvious that the trace norm of a matrix is a convex approximation to its rank. The heuristic optimization is as the following

$$\begin{aligned} &\min_{\mathbf{X}} \ \|\mathbf{X}\|_* \\ &s.t.\ \mathbf{X}_{ij} = \mathbf{M}_{ij},\ (i,j) \in \Omega\,. \end{aligned} \tag{2}$$

Candès *et al.* proposed Singular Value Thresholding (SVT) algorithm [Cai *et al.*, 2008] to solve Eq. (2). Meanwhile, Candès et al. further provided the assumption for a perfect recovery with high probability [Candès and Recht, 2009].

Other researchers relaxed the constraints [Fazel, 2002; Ji and Ye, 2009] to allow the recovery errors and make the matrix completion more flexible by solving:

$$\min_{\mathbf{X}} \|\mathbf{X}_\Omega - \mathbf{M}_\Omega\|_F^2 + \lambda \|\mathbf{X}\|_*\ , \tag{3}$$

where $\lambda > 0$ is the regularity parameter. Regularization by trace norm has been shown to lead to low-rank solution with certain assumptions [Fazel *et al.*, 2001]. However, Shi *et al.* [Shi and Yu, 2010] pointed out in many cases, such yielded solution is not really low-rank and there might be multiple unstable solutions. In addition to this deficiency, tuning λ in Eq. (3) for trace norm approach is often expensive and difficult.

In this paper, we tackle the rank minimization problem by a new approximation formulation. Instead of minimizing trace norm to get an approximation matrix, we explicitly require its low-rank. Different from conventional rank minimization approaches, we set exact rank as a parameter and provide users freedom to choose an appropriate value. We seek a matrix that fits the observed values appropriately, meanwhile explicitly require the approximation matrix to be low-rank. The proposed objective function is as follows:

$$\begin{array}{l} \min\limits_{\mathbf{X}} \ \|\mathbf{X}_\Omega - \mathbf{M}_\Omega\|_1 \\ s.t.\ rank(\mathbf{X}) \leq k\,, \end{array} \tag{4}$$

where k is a user-specified value. When k is not very large, our method is a good approximation to Eq. (1). The range of the parameter k can be easily decided during the practical applications. Conventional methods generally use ℓ_2 norm as discrepancy measure on observed subset, which leads to analytical convenience but often suffers from noise disturbance. Here we use ℓ_1 norm to make our method less prone to outlier influence. We call the proposed method as Robust Rank-k Matrix Completion (RRMC) approach.

3 Optimization Algorithm

As mentioned in the above section, the trace norm minimization of a matrix is a convex approximation of its rank minimization problem, therefore there is no guarantee such yielded solution is indeed low-rank when theoretical assumptions are not satisfied. This is a key difference between trace norm minimization approaches and exact rank matrix recovery methods. On the other hand, the low-rank minimization methods for Eq. (1) often involve expensive computation costs.

Our new RRMC objective function in Eq. (4) uses ℓ_1 norm to alleviate the outlier influence, meanwhile the rank constraint guarantees the low-rank of the approximation matrix. In the literature of machine learning and its closely related areas, ℓ_1 norm has been used for feature selection [Nie *et al.*, 2010; Cai *et al.*, 2011], graph based learning [Nie *et al.*, 2011], and model selection [Shao, 1996; Tibshirani, 1996]. The key advantage of using ℓ_1 norm over ℓ_2 is that each data point no longer enters the objective function as squared residual error. Optimizing objective function in Eq. (4) is difficult due to ℓ_1 norm and the exact rank constraint.

To solve our new objective function, we will derive the optimization algorithm using ancillary variables and Augmented Lagrangian Multiplier (ALM)Methods. ALM [Bertsekas, 2003] were originally proposed for convex problems and recently extended to non-separable, non-convex problems. The main idea is to eliminate equality constraints and instead add a penalty term to the cost function that assigns a very high cost to infeasible points. ALM differs from other penalty-based approaches by simultaneously estimating the optimal solution and Lagrange multipliers in an iterative fashion.

We introduce the ancillary variable $\mathbf{Y}$ that approximates $\mathbf{X}$ and re-write Eq. (4) into the following one:

$$\begin{array}{l} \min\limits_{\mathbf{X},\mathbf{Y}} \|\mathbf{X}_\Omega - \mathbf{M}_\Omega\|_1 + \frac{\mu}{2}\left\|\mathbf{X} - \mathbf{Y} + \frac{1}{\mu}\mathbf{\Lambda}\right\|_F^2 \\ s.t.\ rank(\mathbf{Y}) \leq k\,, \end{array} \tag{5}$$

where μ is the regularity coefficient that balances the first term and second term in Eq. (5). $\mathbf{\Lambda}$ is used to adjust the discrepancy between $\mathbf{X}$ and $\mathbf{Y}$. They are parameters of ALM and their update rules will be provided shortly.

Since Eq. (5) involves two variables $\mathbf{X}$ and $\mathbf{Y}$, we solve it in an alternative manner and repeat the following steps until convergence.

When $\mathbf{Y}$ is fixed, we get $\mathbf{X}_\Omega$ and $\mathbf{X}_{\Omega^c}$ separately. It is easy to get the $\mathbf{X}_{\Omega^c}$ formula

$$\mathbf{X}_{\Omega^c} = \mathbf{Y}_{\Omega^c} - \frac{1}{\mu}\mathbf{\Lambda}_{\Omega^c} \tag{6}$$

as the first term in Eq. (5) is a constant with respect to $\mathbf{X}_{\Omega^c}$.

Optimizing Eq. (5) with respect to $\mathbf{X}_\Omega$ is equivalent to optimizing

$$\min_{\mathbf{X}_\Omega} \frac{1}{2}\left\|\mathbf{X}_\Omega - (\mathbf{Y}_\Omega - \frac{1}{\mu}\mathbf{\Lambda}_\Omega)\right\|_F^2 + \frac{1}{\mu}\|\mathbf{X}_\Omega - \mathbf{M}_\Omega\|_1 \tag{7}$$

We can let $\mathbf{Z}_\Omega = \mathbf{X}_\Omega - \mathbf{M}_\Omega$ and write the above equation into the following one

$$\min_{\mathbf{Z}_\Omega} \frac{1}{2}\left\|\mathbf{Z}_\Omega - (\mathbf{Y}_\Omega - \frac{1}{\mu}\mathbf{\Lambda}_\Omega - \mathbf{M}_\Omega)\right\|_F^2 + \frac{1}{\mu}\|\mathbf{Z}_\Omega\|_1 \tag{8}$$

According to [Liu and Ye, 2009], there is a closed form solution to the following one:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{9}$$

where vectors $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$, scalar $\lambda > 0$. The solution is

$$\mathbf{x}_i = sign(\mathbf{v}_i)\max(abs(\mathbf{v}_i) - \lambda, 0) \tag{10}$$

where sign(t) is the signum function: if $t > 0$, $sign(t) = 1$; if $t < 0$,$sign(t) = -1$; and if $t = 0$, $sign(t) = 0$.

Now according to the solution Eq. (10) to Eq. (9), we solve Eq. (8) in a decouple way. The solution to Eq. (8) is

$$\mathbf{Z}_\Omega = sign(\mathbf{Y}_\Omega - \frac{1}{\mu}\mathbf{\Lambda}_\Omega - \mathbf{M}_\Omega)N, \tag{11}$$

where $N = \max(abs(\mathbf{Y}_\Omega - \frac{1}{\mu}\mathbf{\Lambda}_\Omega - \mathbf{M}_\Omega)_i - \frac{1}{\mu}, 0)$.

Then it is straightforward to get the formula

$$\mathbf{X}_\Omega = \mathbf{Z}_\Omega + \mathbf{M}_\Omega \tag{12}$$

When $\mathbf{X}$ is fixed, optimizing Eq. (5) is equivalent to

$$\begin{array}{l} \min\limits_{\mathbf{Y}} \left\|\mathbf{Y} - (\mathbf{X} + \frac{1}{\mu}\mathbf{\Lambda})\right\|_F^2 \\ s.t.\ rank(\mathbf{Y}) \leq k \end{array} \tag{13}$$

Algorithm 1 Robust Rank k Matrix Completion

Input: Available entries $\mathbf{M}_\Omega$, user specified rank value k.
Output: Prediction matrix $\mathbf{X}$
Initialize $\mathbf{Y}$, μ, ρ and $\mathbf{\Lambda}$
repeat
  Update $\mathbf{X}$ according to Eq. (6) and Eq. (12)
  Update $\mathbf{Y}$ according to Eq. (14)
  Update μ and $\mathbf{\Lambda}$ according to Eq. (15).
until converged

Assuming the SVD decomposition of $\mathbf{X}+\frac{1}{\mu}\mathbf{\Lambda}$ is $\mathbf{FSG}^T$, then the solution of $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{F}_k\mathbf{S}_k\mathbf{G}_k^T, \tag{14}$$

where $\mathbf{S}_k$ contains top k largest values and $\mathbf{F}_k$, $\mathbf{G}_k$ are the singular vector matrices corresponding to $\mathbf{S}_k$.

Finally, we update $\mathbf{\Lambda}$ and μ:

$$\begin{aligned} \mathbf{\Lambda} &= \mathbf{\Lambda} + \mu(\mathbf{X} - \mathbf{Y}) \\ \mu &= \rho\mu, \end{aligned} \tag{15}$$

where $\rho > 1$ is an ALM method parameter.

The whole algorithm is summarized in Algorithm 1. We use the relative change of objective function value becomes less than 10^{-4} as the convergence criterion. Readers interest in the detailed convergence proof of algorithm 1 may refer to [Bertsekas, 2003]. Here we provide an intuitive verification due to space constraint. As μ goes infinity, to minimize Eq. (5), we can ignore the first term and get $\mathbf{Y} = \mathbf{X}$, which implies the convergence of the algorithm. Note that our algorithm guarantees an asymptotic convergence, *i.e.*, the algorithms converges after certain number of iteration. The exact number of iterations t required clearly depends on both the data attributes and tolerance, nevertheless, it is always below 25 for all of our experiments in this paper. The computation cost for $\mathbf{X}_{\Omega^C}$ and $\mathbf{X}_\Omega$ are both of order $O(nm)$ and $\mathbf{Y}$ is of order $O(nm^2)$, therefore the total computational cost is of order $O(tnm^2)$.

Now we want to discuss the influences of the ALM parameters μ, $\mathbf{\Lambda}$ and ρ to the algorithm convergence. It is easy to see the algorithm converges when the regularity penalty μ is sufficiently large. From Eq. (15), we can see μ grows exponentially, therefore the initial value of μ has little impact given sufficient number of iterations. Similar reasoning applies to $\mathbf{\Lambda}$, when μ is large enough, $\mathbf{\Lambda}$ in the second term can be ignored. On the other hand, ρ has a significant impact on the convergence speed and objective function value accuracy. A larger ρ value would reduce the computational time but meanwhile compromise the objective function accuracy. Suggested ρ value is between 1.01 and 1.1 [Bertsekas, 2003].

4 Experiments

In this section, we empirically evaluate our method RRMC for the matrix completion effects on a few real data sets. As mentioned in the introduction, matrix completion problem has applications in different disciplines. The remainder part of this section consists of several subsections. In each subsection, we present the necessary background information for each experiment data set and then the corresponding result. To better demonstrate the impressive effect of our method, we also include a few classical competitive methods: SVD, singular value projection [1](SVP) [Jain *et al.*, 2010], robust PCA [2](RPCA) [Wright *et al.*, 2009], singular value thresholding [3](SVT) [Cai *et al.*, 2008], OPTSpace [4] [Keshavan and Montanari, 2010].

Unless specified otherwise, we select the ground truth rank of the recovery matrix from 5 to 20, with increment of 5. Regularity parameter for SVT are tuned from the list $\{0.001, 0.01, 0.1, 1\}$. Since the default step size for SVT and SVP would result in divergence for some our data sets, we set them at conservative values. Assuming p is the ratio between observations and number of elements, for SVT, we set $\tau = \frac{5}{\sqrt{mn}}, \delta = \frac{1}{20p}$,tol$= 10^{-4}$, while for SVP, we set tol$= 10^{-4}$, vtol$= 10^{-3}$,verbosity$= 1$, $\delta = \frac{1}{20p}$. For SVD, we always initialize the missing entries with 0s, set $\mathbf{Y}$ the initialized $\mathbf{M}$ for RRMC. Note that RRMC converges within 20 iterations on all of the following data sets. We report results below the average of 20 runs.

4.1 Experiment on Synthetic Data Set

We first evaluate our method against other methods for random low-rank matrices and uniform samples. We generate a random rank 2 matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ and generate random Bernoulli samples with probability 0.1.

we introduce two evaluation metrics to measure the prediction error, mean average error (MAE) and root mean square error (RMSE), where sd represents the standard deviation. These two metrics are commonly used for matrix completion evaluation.

$$\begin{aligned} \text{MAE} &= \text{mean}\,|\mathbf{X}_{\text{ij}} - \mathbf{M}_{\text{ij}}| \\ \text{RMSE} &= \frac{\sqrt{\text{mean}(\mathbf{X}_{\text{ij}} - \mathbf{M}_{\text{ij}})^2}}{\text{sd}(\mathbf{M}_{\text{ij}})}, \quad (\text{i},\text{j}) \notin \Omega \end{aligned}$$

We add approximately 10% Gaussian noise and conduct matrix completion experiment as n increases from 1000 to 5000. In Fig. 2(a), it can be observed that SVD is very sensitive to the moderate noise while all other competitive methods are relatively robust. RRMC shows significant improvement in RMSE when n is larger than 2000 nevertheless.

Next we fix the matrix size at 2000×2000 but vary the noise level from 5% to 25%. With the increased noise, all methods are more prone to outlier influence. In Fig. 2(b), we can observe RRMC method shows best RMSE at noise level 15% and thereafter. With the increased level noise, the matrix has violated the structure assumptions many methods assumed. In contrast, since RRMC has no requirement on the matrix structure, it shows the most robust performance.

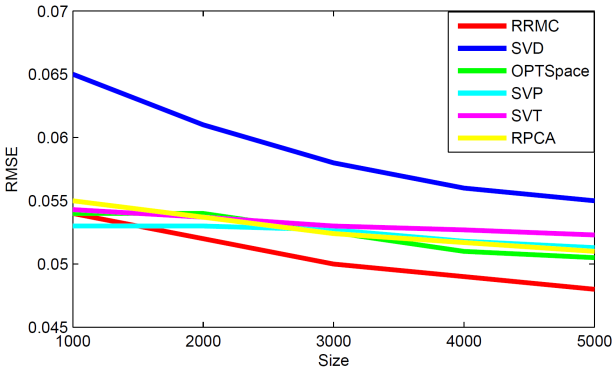
So far, we have completed all experiments on the synthetic data. These experiments show that RRMC is empirically more robust to noise against other methods, also its performance is stable. We will now shift focus on 3 real-world data sets.

[1] www.cs.utexas.edu/ pjain/svp/

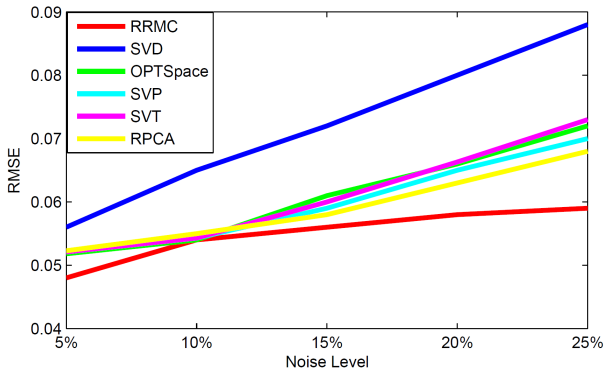
[2] perception.csl.uiuc.edu/matrix-rank/samplecode.html

[3] www-stat.stanford.edu/ candes/software.html

[4] www.stanford.edu/ raghuram/optspace/code.html



(a) Matrix Completion with Moderate Noise



(b) Matrix Completion with Large Noise

Figure 2: Matrix completion on synthetic data set with various matrix sizes and noise levels. (a) RMSE by various methods for matrix completion with p=0.1, k=2 and around 10% known entries are corrupted. (b) RMSE with increased levels of noise from 5% to 25%.

Table 1: Description of 3 Data Sets

Data set	Epinions	Wikipedia	Slashdot
# of Nodes	2000	2000	2000
# of Trust Links	149,146	81,232	74,561
% among All Links	3.73	2.03	1.86

4.2 Experiment on Real Data Sets

The trust graph data sets we use are Epinions [Massa and Avesani, 2006], Wikipedia [Leskovec *et al.*, 2010] and Slashdot [Leskovec *et al.*, 2009]. All above data sets contain directed edges with two edges labels, described as trust link (1) and distrust link (0). One thing needs to point out is that the original data sets only record trust links, therefore most of the distrust links are indeed un-tagged ones. Epinions is a data set that users tag trust/distrust votes based on their opinions on others' reviews on items. Wikipedia records a series of elections that contributors from Wikipedia vote on each other to become administrators. Slashdot carries the users' tag on each other as friends or foes based on their submitted technology news. In summary, Epinions contains about 50,000 users and 490,000 trust links, Wikipedia contains about 7,000 users and 103,000 trust links, Slashdot contains about 80,000 users and 900,000 trust links.

It can be observed that the distributions of links in these data sets are very skewed due to the domination of distrust links. To alleviate the data skewness for fair comparison and keep the computation manageable, we select top 2,000 highest degree users from each data set. Table 1 gives a summary description about the subsets used in our experiment. Note that the subsets still carry a skewed distribution in trust and distrust links.

4.3 Matrix Completion Result on Trust Graphs

To evaluate the matrix completion result for all methods, we have to mask a portion of ground truth values in each trust graph for test purpose. We randomly hide 90% of the true value of links and make prediction based on 10% available entries. We choose 90% missing here to simulate the severe data sparsity real social graphs generally suffer from. The reason is that most users from the online community only explicitly express trust and distrust to a *very small* fraction of peer users considering total number of users. For each run, after we mask these values, we initiate these positions with random values between 0 and 1, then all methods predict these values with the specified parameters above.

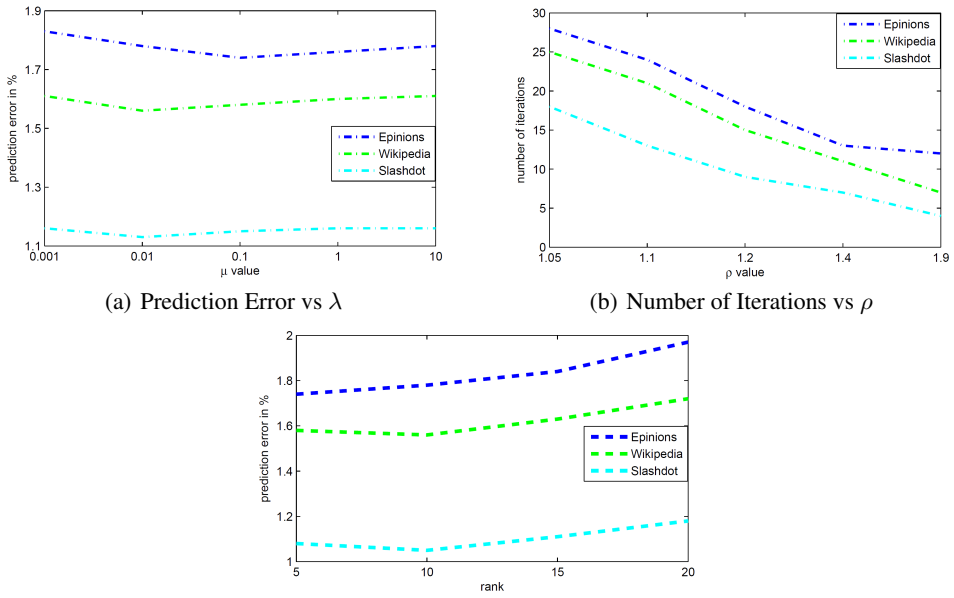
From Table 2, we can observe that RRMC outperforms all other methods in terms of MAE and RMSE on three data sets. There are a few additional discoveries. First, we can see most competitive methods get quite close result except SVD. This is not so surprising since these methods all try to get a low-rank solution. The relatively poor performance for SVD could be due to its vulnerability to extreme sparse observations and the noise. Second, we can see RPCA also gets a relatively better performance due to its adoption of ℓ_1 norm.

4.4 Experiment on Method Parameters

In this part, we show the performance of our method regarding to the parameters in Fig. (3). Specifically, we show prediction error is robust to the choice of λ, the relationship between ρ and number of iterations needed for convergence. It can be concluded that our method is not so sensitive to the choice of parameters. In addition to the above two figures, we also include the performance of RRMC with respect to rank k. RRMC gets optimal performance when k=5 for Epinions, and k=10 for Wikipedia and Slashdot. This confirms our assumption that the trust graph between users could be dominated by a few latent factors.

5 Conclusion

Trust and distrust prediction plays a key role in privacy protection for online users. Such prediction can be modeled as predicting missing elements in the trust matrices. Such matrices are sparse and could have local region symmetric and transitive properties. Conventional matrix completion methods don't work very well on those special link prediction problem.



(a) Prediction Error vs λ

(b) Number of Iterations vs ρ

(c) Prediction Error vs k

Figure 3: Method Parameters Test. (a) shows the relation between prediction error and λ. (b) demonstrates how ρ affects number of iterations. (c) displays the prediction error change with respect to rank.

(a) Epinions

methods	MAE(%)	RMSE
RRMC	$\mathbf{1.74 \pm 0.13}$	$\mathbf{0.93 \pm 0.11}$
SVD	2.14 ± 0.34	1.02 ± 0.14
OPTspace	1.86 ± 0.18	0.98 ± 0.13
SVP	1.89 ± 0.21	1.01 ± 0.17
SVT	1.92 ± 0.19	1.03 ± 0.15
RPCA	1.78 ± 0.12	0.96 ± 0.14

(b) Wikipedia

methods	MAE(%)	RMSE
RRMC	$\mathbf{1.56 \pm 0.14}$	$\mathbf{0.81 \pm 0.05}$
SVD	1.87 ± 0.18	0.92 ± 0.11
OPTspace	1.81 ± 0.15	0.87 ± 0.09
SVP	1.74 ± 0.18	0.89 ± 0.11
SVT	1.73 ± 0.16	0.88 ± 0.12
RPCA	1.61 ± 0.15	0.83 ± 0.07

(c) Slashdot

methods	MAE(%)	RMSE
RRMC	$\mathbf{1.05 \pm 0.09}$	$\mathbf{0.69 \pm 0.04}$
SVD	1.27 ± 0.17	0.83 ± 0.09
OPTspace	1.18 ± 0.13	0.78 ± 0.07
SVP	1.13 ± 0.12	0.76 ± 0.08
SVT	1.11 ± 0.11	0.74 ± 0.07
RPCA	1.09 ± 0.10	0.71 ± 0.05

Table 2: Matrix Completion Result on Different Data Sets

In this paper, we propose a robust rank-k matrix completion (RRMC) prediction method that yields prediction matrix with exact rank. Our framework seeks a low-rank matrix and is robust to data noise and potential outliers. Different from trace norm minimization, our method explicitly seeks a low-rank stable matrix. We solve the difficult integer programming problem via introducing an ancillary variable and decomposing the difficult problem into two manageable pieces. The empirical experiments on three trust social graph data sets demonstrate the effectiveness and robustness of our method.

In the future, we will look into pursuing trust prediction from a different perspective. In this paper, we generate trust prediction in the continuous domain. However, for many user, providing explicit instruction whether to trust or not is preferable. In other words, we may seek low-rank matrix completion in the discrete domain.

References

[Bedi *et al.*, 2007] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2677–2682, 2007.

[Bertsekas, 2003] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, New York,NY, 2003.

[Billsus and Pazzani, 1998] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *ICML*, pages 46–54, 1998.

[Cai *et al.*, 2008] F. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2008.

[Cai *et al.*, 2011] Xiao Cai, Feiping Nie, Heng Huang, and Chris Ding. Multi-class $l_{2,1}$-norm support vector machine. In *ICDM*, pages 91–100, 2011.

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

[Candès and Tao, 2010] E.J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transaction on Information Theory*, 56(5):2053–2080, 2010.

[Fazel *et al.*, 2001] M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference*, volume 6, pages 4734–4739, 2001.

[Fazel, 2002] M. Fazel. *Matrix Rank Minimization with Applications*. Phd thesis, Stanford University, 2002.

[Huang *et al.*, 2012] J. Huang, F. Nie, H. Huang, and Y. Tu. Trust prediction via aggregating heterogeneous social networks. In *21st ACM international Conference on Information and Knowledge Management*, pages 1774–1778, 2012.

[Jain *et al.*, 2010] P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

[Jeh and Widom, 2002] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *KDD*, pages 538–543, 2002.

[Ji and Ye, 2009] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, pages 457–464, 2009.

[Katz, 1953] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[Keshavan and Montanari, 2010] R.H. Keshavan and A. Montanari. Regularization for matrix completion, 2010.

[Leskovec *et al.*, 2009] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[Leskovec *et al.*, 2010] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, 2010.

[Liben-Nowell and Kleinberg, 2003] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[Liu and Ye, 2009] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *ICML*, pages 83–90, 2009.

[Massa and Avesani, 2006] P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, 2006.

[McPherson *et al.*, 2001] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27, 2001.

[Meka *et al.*, 2009] R. Meka, P. Jain, and I.S. Dhillon. Matrix completion from power-law distributed samples. In *NIPS*, pages 1258–1266, 2009.

[Newman, 2001] M. Newman. Clustering and preferential attachment in growing networks. *Physics Review Letter E*, 64, 2001.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In *NIPS*, 2010.

[Nie *et al.*, 2011] Feiping Nie, Hua Wang, Heng Huang, and Chris Ding. Unsupervised and semi-supervised learning via l_1-norm graph. In *ICCV*, pages 2268–2273, 2011.

[Salton and McGill, 1983] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[Shao, 1996] J. Shao. Bootstrap model selection. *Journal of the American Statistical Association*, 91:655–665, 1996.

[Shi and Yu, 2010] X. Shi and P. Yu. Limitations of matrix completion via trace norm minimization. *SIGKDD Explorations*, 12(2):16–20, 2010.

[Tibshirani, 1996] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B:267–288, 1996.

[Wen and Lin, 2010] Z. Wen and C.Y. Lin. On the quality of inferring interests from social neighbors. In *Proceedings of 16th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 373–382, 2010.

[Wright *et al.*, 2009] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*, pages 2080–2088, 2009.