# Multi-Label Ranking Loss Minimization for Matrix Completion

**Jiaxuan Li, Xiaoyan Zhu*, Hongrui Wang, Yu Zhang, Xin Lai, Jiayin Wang**
School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
lijiaxuan@stu.xjtu.edu.cn, zhu.xy@xjtu.edu.cn, wanghongrui@stu.xjtu.edu.cn, yuzhang990825@stu.xjtu.edu.cn, laixin@xjtu.edu.cn, wangjiayin@mail.xjtu.edu.cn

**Abstract**

The common matrix completion methods minimize the rank of the matrix to be completed in addition to the Hamming loss between the incomplete and completed matrices. The rank of matrix measures the linear relation among the vectors of matrix, which may introduce ambiguity for data recovery. To cope with this issue, we extend multi-label ranking loss into matrix completion, and employ multi-label ranking loss minimization (MLRM) in this paper to exploit the relative correlation among matrix vectors. In MLRM, the original incomplete matrix is converted into a pairwise ranking matrix, and the approximation on this newly generated matrix can be viewed as a surrogate of multi-label ranking loss to replace the Hamming loss pattern in the existing methods. Extensive experiments demonstrate that MLRM outperforms the state-of-the-art matrix completion methods in varies of applications, including movie recommendation, drug-target interaction prediction and multi-label learning.

## 1 Introduction

Matrix completion addresses the data recovery problem where only a few entries can be observed in a low-rank matrix. It has wide applications in recommendation systems (Koren, Bell, and Volinsky 2009; Jannach et al. 2016), computer vision (Cabral et al. 2011), signal processing (Chiang, Hsieh, and Dhillon 2015), bioinformatics (Cai, Zhang, and Wan 2011; Luo et al. 2018), etc.

The common matrix completion methods (Recht 2011; Goldberg et al. 2010; Cai, Candès, and Shen 2010; Xu, Jin, and Zhou 2013) focus on the assumptions of consistency and dependency of the matrix to be completed. Matrix consistency assumes that the completed matrix $\boldsymbol{X}$ should be as close as possible to the original incomplete matrix $\boldsymbol{Y}$ in positions those are observed. Such element-wise pattern can be formulated as the Hamming loss between $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\min_{\boldsymbol{X}} \sum_{ij} \mathbb{I}(\mathcal{R}_\Omega(\boldsymbol{X}_{ij}) \neq \mathcal{R}_\Omega(\boldsymbol{Y}_{ij})) \tag{1}$$

where $\Omega \subset \{(1, \cdots, n) \times (1, \cdots, m)\}$ is the index set of known entries, and $\mathcal{R}_\Omega(\cdot)$ is an observation operator:

$$\mathcal{R}_\Omega(\boldsymbol{Y}_{ij}) = \begin{cases} \boldsymbol{Y}_{ij}, & (i,j) \in \Omega \\ 0, & (i,j) \notin \Omega \end{cases} \tag{2}$$

and $\mathbb{I}(\cdot)$ is an indicator function returning 1 if true and 0 otherwise.

Matrix dependency implies that the completed matrix $\boldsymbol{X}$ should be low-rank. Simply put, matrices in the real world are redundant, meaning that the vectors in the completed matrix should be linearly correlated. And such matrix-wise pattern can be formulated as:

$$\min_{\boldsymbol{X}} r(\boldsymbol{X}) \tag{3}$$

where $r(\cdot)$ is the rank function of the given matrix.

Matrix consistency assumption fundamentally ensures the accuracy of matrix completion, and matrix dependency assumption provides the feasible manner for matrix completion. Unfortunately, given an incomplete matrix, different completed matrices may have the same rank; that is, minimization on matrix rank can cause data recovery ambiguity, and the completed matrix with low-rank property inevitably loses information. An example of this phenomenon is shown in Tab. 1, where 3 users contributes their willingness to 4 movies with $1, -1, 0, *$ representing positive, negative, neutral and unknown attitudes, respectively. To fill the unknown

Table 1: An example for matrix completion, where $u_i, m_j$ represent $i$-th user and $j$-th movie, and $\boldsymbol{X}_{ij}$ denotes the favorability of $u_i$ towards $m_j$.

| | $m_1$ (action) | $m_2$ (drama) | $m_3$ (comedy) | $m_4$ (fiction) |
|---|---|---|---|---|
| $u_1$ | 1 | -1 | -1 | 1 |
| $u_2$ | -1 | 1 | 0 | -1 |
| $u_3$ | 1 | -1 | $e_1$ | $e_2$ |

entries, matrix completion methods based on low-rank assumption consider two strategies for filling the missing values, i.e., $f_1 : u_3 = u_1$ or $f_2 : u_3 = -u_2$. Both strategies complete $e_2 = 1$; but conflict occurs that $e_1 = -1$ under $f_1$ while $e_1 = 0$ under $f_2$. This is the matrix completion ambiguity under low-rank assumption. To cope with this issue, the ranking of user's preference on different movies can be

*indicates the corresponding author.

employed. In the given example, it is obvious that users have similar preferences for same type of movies; users who are interested in action often do not reject fiction and are less willing to watch drama or comedy films, vice versa. Therefore, the preference ranking of $u_3$ for 4 movies can be considered as $\boldsymbol{X}_{31} \geq \boldsymbol{X}_{34} > \boldsymbol{X}_{33} \geq \boldsymbol{X}_{32}$. In this ranking assumption, the matrix rank is avoided from direct minimization, yet the relative ranking among columns also measures the correlation and provides potential generalization on recovering unknown entries.

In this paper, we take multi-label ranking loss minimization (MLRM) to measure the relative correlation among matrix vectors and deal with matrix completion. In multi-label learning (Gibaja and Ventura 2015; Zhang and Zhou 2013), a data example can be associated with multiple labels, that is, multi-label learner induction can be viewed as the fitting and intrinsic dependency exploiting on label matrix, which is consistent with the assumptions of matrix completion. Multi-label ranking loss (Schapire and Singer 2000) measures the pairwise label correlation; it cares nothing about forecast accuracy, and instead focuses on whether the predicted values of positive labels are greater than those of negative ones. According to this ranking strategy, in MLRM, the incomplete matrix is first converted to a pairwise ranking matrix according to the comparisons between matrix vector pairs. Then, the approximation on pairwise ranking matrix can be incorporated into the objective function, where the optimization techniques in existing methods can be employed, including matrix consistency and dependency, side information (Xu, Jin, and Zhou 2013), matrix factorization (Yang, Li, and Wang 2020; Xie and Huang 2021), etc. To summarize, MLRM has three aspects of advantages:

- It avoids the potential ambiguity in simplex low-rank assumption;
- It incorporates the relative relation in addition to linear relation among matrix vectors;
- It incorporates the pairwise correlation (local correlation, multi-label ranking loss minimization) in addition to matrix-wise correlation (global correlation, matrix rank minimization) and element-wise correlation (non-correlation, Hamming loss minimization).

To verify the effectiveness of MLRM, extensive experiments are conducted on diverse matrix completion scenarios including movie recommendation, drug-target interaction prediction and multi-label learning. The experimental results demonstrate that MLRM outperforms the state-of-the-art comparison methods.

## 2 The Related Work

In matrix completion (Goldberg et al. 2010), the matrix to be completed $\boldsymbol{Y} \in \mathbb{R}^{n \times m}$ is sparse and have only a few of entries known, and a set $\Omega$ can be taken to record the indices of these observed entries. Matrix completion methods can be divided into mainly two categories, transductive methods (Goldberg et al. 2010; Cai, Candès, and Shen 2010; Ma, Goldfarb, and Chen 2011; Gu et al. 2014) and inductive methods (Xu, Jin, and Zhou 2013; Natarajan and Dhillon 2014; Lu et al. 2016; Yang, Li, and Wang 2020).

The transductive matrix completion methods assume the low-rank property of $\boldsymbol{Y}$, and hence minimize the rank of the completed matrix $r(\boldsymbol{X})$. In addition, the Hamming loss between completed and incomplete matrices in terms of the observation positions should also be minimized. To facilitate optimization, the rank function is usually replaced with a derivable surrogate, and the Hamming loss is generally replaced with the squared Frobenius norm (Ma, Goldfarb, and Chen 2011; Gu et al. 2014). The objective function of the transductive methods can be summarized as:

$$\min_{\boldsymbol{X}} \lambda \|\boldsymbol{X}\|_* + \|\mathcal{R}_\Omega(\boldsymbol{X} - \boldsymbol{Y})\|_F^2 \tag{4}$$

where $\|\boldsymbol{X}\|_*$ represents the surrogate of the rank of $\boldsymbol{X}$. As the optimization on matrix rank is a NP-hard problem, the nuclear norm (Cai, Candès, and Shen 2010; Bach 2008) of the matrix is often taken as the surrogate loss function, and some non-convex substitutes (Gu et al. 2014; Guo et al. 2023) are also proposed to better approximate the matrix rank.

However, the transductive methods suffer from the cold-start problem, that is, the matrix completion algorithms have difficulty in recovering sparse matrix without any previously known features. Actually, the incomplete matrices in real world applications are usually not isolated, and side information matrices may provide some auxiliary information. For example, in movie recommendation, there are many information feature the users and movies in addition to the rating matrix from users to movies. To utilize such side information, the inductive matrix completion methods propose to minimize the difference between $\boldsymbol{Y}$ and the product of side matrix and the optimized matrix (Xu, Jin, and Zhou 2013; Natarajan and Dhillon 2014; Lu et al. 2016; Yang, Li, and Wang 2020):

$$\min_{\boldsymbol{X}} \lambda \|\boldsymbol{X}\|_* + \|\mathcal{R}_\Omega(\boldsymbol{A}\boldsymbol{X}\boldsymbol{C}^T - \boldsymbol{Y})\|_F^2 \tag{5}$$

where $\boldsymbol{A}, \boldsymbol{C}$ are the left and right side matrices. Generally, inductive methods are able to achieve more improvement, but the performance is influenced by the quality of side information.

In addition, matrix factorization techniques can also be employed to improve matrix completion. For example, FNNM (Yang, Li, and Wang 2020) takes $\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{C}^T$ to approximate the incomplete matrix,

$$\min_{\boldsymbol{M},\boldsymbol{N}} \lambda_1 \|\boldsymbol{M}\|_* + \lambda_2 \|\boldsymbol{N}\|_1 + \frac{1}{2} \|\mathcal{R}_\Omega(\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{C}^T - \boldsymbol{Y})\|_F^2 \tag{6}$$

where $\boldsymbol{M}$ aims at maintaining low-rank property and $\boldsymbol{N}$ is a sparse interaction matrix associating with the side information.

## 3 The Proposed Method

In this section, we first introduce ranking loss in multi-label binary classification, and extend it to general matrix formulation. Then, objective function based on multi-label ranking loss minimization is established for matrix completion. Last, we give a strategy to correct the output values of the ranking-based model, and the optimization method is also provided.

## 3.1 Multi-Label Ranking Loss Formulation

In multi-label learning, each instance is associated with a set of labels and these labels are inherently correlated. To measure such correlation, ranking loss (Schapire and Singer 2000) is proposed to evaluate the proportion of label pairs that are inversely ranked, as shown in Eq. 7. The label space of multi-label classification can be expressed as a binary matrix $\boldsymbol{Y} \in \{-1,1\}^{n\times m}$, where $n, m$ are the numbers instances and labels, $-1, 1$ represent negative and positive labels, and the output of the multi-label algorithm can be defined as matrix $\hat{\boldsymbol{Y}} \in [-1,1]^{n\times m}$, where the larger an absolute value, the higher its confidence of being a negative or positive label.

$$RL = \frac{1}{n}\sum_{i=1}^{n} \frac{\left|\left\{(p,q)\middle|\boldsymbol{Y}_{ip}=1, \boldsymbol{Y}_{iq}=-1, \hat{\boldsymbol{Y}}_{ip} < \hat{\boldsymbol{Y}}_{iq}\right\}\right|}{|\boldsymbol{Y}_i|\cdot\left|\bar{\boldsymbol{Y}}_i\right|} \tag{7}$$

where $|\boldsymbol{Y}_i|, \left|\bar{\boldsymbol{Y}}_i\right|$ defines the number of positive and negative labels corresponding to the $i$-th label vector, respectively.

Essentially, ranking loss measures the ranking relations between positive labels and negative labels for each instance. It is hard to directly optimize the ranking loss between $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$, due to different instances being differently labeled. In fact, as a concept introduced early on, ranking loss requires annotated labels to model tasks such as recommendation systems (Rendle et al. 2012) and multi-label learning (Xie and Huang 2018), which is not directly suitable for general matrix completion. To handle matrix completion problem, we would like to formalize ranking loss into matrix form. For matrix without a prior, the ranking loss can be measured among row or column vectors. Without loss of generality, we concentrate on the ranking loss among matrix columns in this paper.

First, we relax the positive-negative label correlation w.r.t. distinct instance to general pairwise correlation on the top of matrix columns. Given matrix $\boldsymbol{Y}_{n\times m}$, the relaxed multi-label ranking loss can be defined as the pairwise comparisons among $m$ column vectors, which obviously covers the measurement of ranking loss in Eq. 7. Obviously, there are $\binom{m}{2} = \frac{m(m-1)}{2}$ nonredundant column pairs, and the indices set of column pairs is $S = \{(p,q)|1 \leq p < q \leq m\}$. To make a clear explanation, $S$ can be unfolded as $\{S^1; S^2; \cdots; S^{m-1}\}$ where $S^p$ represents the indices of column pairs $(p,q), q > p$.

$$S^p = \{(p,p+1), (p,p+2)\cdots,(p,m)\}, |S^p| = m-p \tag{8}$$

Second, we conduct comparisons between column pairs to generate a pairwise ranking matrix. Given the original matrix $\boldsymbol{Y}_{n\times m}$, the pairwise ranking matrix $\boldsymbol{R}_{n\times m(m-1)/2}$ can be defined as:

$$\boldsymbol{R}_{ij} = \begin{cases} 1, & \boldsymbol{Y}_{ip} > \boldsymbol{Y}_{iq} \\ -1, & \boldsymbol{Y}_{ip} < \boldsymbol{Y}_{iq} \\ 0, & \boldsymbol{Y}_{ip} = \boldsymbol{Y}_{iq} \end{cases} \tag{9}$$

where $(p,q) = S_j$ is the $j$-th element of $S$. That is, the $j$-th column of $\boldsymbol{R}$ implies the comparison between column pair $S_j = (p,q)$ of $\boldsymbol{Y}$, and $\boldsymbol{R}$ implies all relative correlations among all column pairs corresponding to $S$.

Third, we extend the relaxed ranking loss based on matrix with binary values to that with continuous values. For $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$, each entry of the pairwise ranking matrix can be rewritten as the difference of two entries in the corresponding column pairs, $\boldsymbol{R}_{ij} = \boldsymbol{Y}_{ip} - \boldsymbol{Y}_{iq}, (p,q) = S_j$, which can also be straightforwardly defined as a matrix transformation:

$$\boldsymbol{R} = \boldsymbol{Y}\boldsymbol{L} \tag{10}$$

where $\boldsymbol{L} \in \{-1,0,1\}^{m\times m(m-1)/2} = [\boldsymbol{L}^1; \boldsymbol{L}^2; \cdots; \boldsymbol{L}^{c-1}]$ is the convert matrix corresponding to $S$, and specifically, $\boldsymbol{L}^p_{(m-p+1)\times(m-p)}$ corresponding to $S^p$ is:

$$\boldsymbol{L}^p = [\boldsymbol{0}_{(i-1)\times(m-i)}; [\boldsymbol{1}_{m-i}; -\boldsymbol{I}_{m-i}]^T] \tag{11}$$

where $\boldsymbol{I}$ is identity matrix.

To summarize, $\boldsymbol{R}$ is able to measure the pairwise column ranking of $\boldsymbol{Y}$. And the minimization on multi-label ranking loss can be relaxed as the approximation on $\boldsymbol{R}$.

## 3.2 Objective Function

As discussed in Eq. 9, pairwise ranking matrix $\boldsymbol{R}$ is able to represent the ranking correlation among entry pairs in the matrix with binary values. For incomplete matrix, the values of unknown entries are difficult to be directly recovered, but the confidence score of an unknown entry $\boldsymbol{Y}_u$ can be seen as no smaller than the negative entry $\boldsymbol{Y}_n$ and no greater than that positive one $\boldsymbol{Y}_p$, i.e., $\boldsymbol{Y}_p \geq \boldsymbol{Y}_u \geq \boldsymbol{Y}_n$. Therefore, we can initially set the unknown entries as neutral values $\boldsymbol{Y}_u = 0$. For general continuous data, we normalize the incomplete matrix as $\tilde{\boldsymbol{Y}} \in [0\pm0.5]^{n\times m}$, and the pairwise ranking matrix $\boldsymbol{R}$ can be reformulated as:

$$\boldsymbol{R} = \mathcal{R}_\Omega(\tilde{\boldsymbol{Y}})\boldsymbol{L} \tag{12}$$

For the sake of convenience, we still record $\boldsymbol{Y} = \tilde{\boldsymbol{Y}}$.

As the minimization on multi-label ranking loss is able to be converted to the approximation on $\boldsymbol{R}$, the transductive matrix completion strategy on the top of multi-label ranking loss can be formulated as:

$$\min_{\boldsymbol{X}} \lambda\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{R}\|_F^2 \tag{13}$$

where $\|\boldsymbol{X}\|_*$ is the nuclear norm of $\boldsymbol{X}$, representing the sum of singular values of $\boldsymbol{X}$.

Next, to overcome cold-start problem, we incorporate side information into the objective function and Eq. 13 can be reformulated as:

$$\min_{\boldsymbol{X}} \lambda\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^T - \boldsymbol{R}\|_F^2 \tag{14}$$

where $\boldsymbol{A} = \mathbb{R}^{n\times d_1}$ is the left side information matrix, $\boldsymbol{B} = \boldsymbol{L}^T\boldsymbol{C}, \boldsymbol{C} = \mathbb{R}^{m\times d_2}$ is the right side information matrix, and $d_1, d_2$ are the number of features corresponding to the rows and columns of the matrix to be completed, respectively.

Inspired by FNNM (Yang, Li, and Wang 2020), both simple transductive and inductive models are difficult to recognize all patterns of the matrix to be completed. One solution is to decompose the incomplete matrix into the sum of a low-rank matrix $\boldsymbol{M}$ and a sparse matrix $\boldsymbol{N}$, where $\boldsymbol{M}$ denotes the transductive pattern with low-rank property and $\boldsymbol{N}$ denotes the inductive pattern need to be learned from side information. Eventually, the proposed MLRM model can be formulated as:

$$\min_{\boldsymbol{M},\boldsymbol{N}} \lambda_1\|\boldsymbol{M}\|_* + \lambda_2\|\boldsymbol{N}\|_1 + \frac{1}{2}\|\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{B}^T - \boldsymbol{R}\|_F^2 \tag{15}$$

To emphasize, although MLRM may seem to be viewed as a variant of FNNM (Yang, Li, and Wang 2020), there are two fundamental difference between the proposed method and previous methods shown in Eq. 5 and Eq. 6. First, the Frobenius norm between $\boldsymbol{X}$ and $\boldsymbol{R}$ is equivalent to the minimization on multi-label ranking loss, while optimization of previous methods is based on Hamming loss. Second, in MLRM, there is no need to employ observation operator $\mathcal{R}_\Omega$ in the process of optimization. Since the loss function focuses on the pairwise correlation among matrix columns, the missing entries can be viewed as neutral values to initiate pairwise ranking relations, and an initial matrix $\boldsymbol{R}$ without unknown entries can be provided.

In addition, there is still one issue worth noting. In MLRM, the dimension of the matrix to be approximated $\boldsymbol{R}$ is about $nm^2/2$, while such figure in conventional inductive methods is $nm$. To tackle the curse of dimensionality, $\boldsymbol{L}$ can be partially sampled in column, as MLRM measures all the pairwise correlations among vectors and some of them are indirectly related and redundant. Suppose that $p, 0 < p < 1$ ratio of columns are sampled, the dimension of $\boldsymbol{R}$ can be reduced into $p \cdot nm^2/2$, namely MLRM$_p$.

### 3.3 Prediction and Result Correction

After solving the objective function Eq. 15, the coefficient matrices can be obtained, i.e., $\hat{\boldsymbol{M}}, \hat{\boldsymbol{N}}$. And the decision function can be written as:

$$\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{M}} + \boldsymbol{A}\hat{\boldsymbol{N}}\boldsymbol{B}^T)\boldsymbol{L}^- \tag{16}$$

where $\boldsymbol{L}^- = \frac{1}{m-1}\boldsymbol{L}^T$ can be viewed as the pseudoinverse of $\boldsymbol{L}$.

As MLRM is established on ranking assumption, it tends to provide a good performance on the relative size between known and unknown entries, but may limit on forecasting the absolute values of missing entries. To correct such gap, we add an adjustment value to each row of the completed matrix $\hat{\boldsymbol{Y}}$ to minimize the mean square error from the original observed entries, specifically, for the $i$-th row,

$$\min_{g_i} \|\mathcal{R}_\Omega(\hat{\boldsymbol{Y}}_i - g_i - \boldsymbol{Y}_i)\|_F^2 \tag{17}$$

The solution of Eq.17 is:

$$g_i = \frac{1}{|\Omega_i|}\sum_j \mathcal{R}_\Omega(\hat{\boldsymbol{Y}}_{ij} - \boldsymbol{Y}_{ij}) \tag{18}$$

where $|\Omega_i|$ denotes the number of observed entries in the $i$-th row of $\boldsymbol{Y}$.

Eventually, the corrected result can be formulated as:

$$\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{M}} + \boldsymbol{A}\hat{\boldsymbol{N}}\boldsymbol{B}^T)\boldsymbol{L}^- + \boldsymbol{g}_{n\times 1} \cdot \mathbf{1}_{1\times m} \tag{19}$$

where $\boldsymbol{g} = [g_1, \cdots, g_n]^T$.

### 3.4 Optimization

The objective function in Eq. 15 involves the optimization of $\ell_1$-norm and nuclear norm. In this paper, an alternating minimization algorithm (Beck 2015) is designed to solve the combination of $\ell_1$-norm and nuclear norm. In each iteration, we fix $\boldsymbol{N}$ to update $\boldsymbol{M}$, and then fix $\boldsymbol{M}$ to update $\boldsymbol{N}$. To make a clear description, we define $f(\boldsymbol{N}) = \frac{1}{2}\|\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{B}^T - \boldsymbol{R}\|_F^2$.

For nuclear norm minimization, singular value threshold (SVT) algorithm (Cai, Candès, and Shen 2010) can be employed. Define $\boldsymbol{E} = \boldsymbol{R} - \boldsymbol{A}\boldsymbol{N}\boldsymbol{B}^T$ and the optimal solution of

$$\min_{\boldsymbol{M}} \lambda_1\|\boldsymbol{M}\|_* + \frac{1}{2}\|\boldsymbol{M} - \boldsymbol{E}\|_F^2 \tag{20}$$

can be defined as the singular value shrinkage operator:

$$\boldsymbol{M} = \mathcal{D}_{\lambda_1}(\boldsymbol{E}) = \boldsymbol{U}\mathcal{D}_{\lambda_1}(\boldsymbol{\Sigma})\boldsymbol{V}^T = \sum_i^{\sigma_i \geq \lambda_1} (\sigma_i - \lambda_1)u_i v_i^T \tag{21}$$

where $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the singular value decomposition (SVD) of $\boldsymbol{E}$, and $\sigma_i, u_i, v_i$ are the $i$-th singular value, left singular vector, right singular vector of $\boldsymbol{E}$. $\mathcal{D}_{\lambda_1}(\boldsymbol{\Sigma})$ can also be expressed as:

$$\mathcal{D}_{\lambda_1}(\boldsymbol{\Sigma}) = \boldsymbol{\Lambda}(\sigma_i - \lambda_1)_+ \tag{22}$$

where $\boldsymbol{\Lambda}(\sigma_i - \lambda_1)_+$ is the diagonal matrix of $\{\sigma_i - \lambda_1, \sigma_i > \lambda_1\}$, shrinking the singular values with threshold and retaining the positive ones.

For $\ell_1$-norm minimization, proximal gradient descent (PGD) (Combettes and Wajs 2005) can be taken. And the objective function

$$\min_{\boldsymbol{N}} \lambda_2\|\boldsymbol{N}\|_1 + f(\boldsymbol{N}) \tag{23}$$

can be iteratively solved due to:

$$\boldsymbol{N} = \mathcal{S}_{\lambda_2}(\boldsymbol{N} - 1/L_p \nabla f(\boldsymbol{N})) \tag{24}$$

where $\mathcal{S}_{\lambda_2}(\boldsymbol{F})$ is the least absolute shrinkage operator on $\boldsymbol{F}$ with threshold $\lambda_2$:

$$\mathcal{S}_{\lambda_2}(\boldsymbol{F})_{ij} = \max(|\boldsymbol{F}_{ij}| - \lambda_2, 0) \cdot sign(\boldsymbol{F}_{ij}) \tag{25}$$

and $L_p$ is the Lipschitz constant that satisfies $\|f(\boldsymbol{N}_1) - f(\boldsymbol{N}_2)\|_F^2 \leq L_p^2\|\boldsymbol{N}_1 - \boldsymbol{N}_2\|_F^2$:

$$L_p = \sqrt{\|\boldsymbol{A}^T\boldsymbol{A}\|_F^2\|\boldsymbol{B}^T\boldsymbol{B}\|_F^2} \tag{26}$$

To summarize, the details of the whole optimization procedure is organized in Alg. 1.

## 4 Experiments

### 4.1 Comparison Methods and Datasets

The proposed method is applied in 3 types of matrix completion data, including movie recommendation (MovieRec), drug-target interaction prediction (DTI) and multi-label learning (MLL). The statistics of benchmark datasets from different sources are listed in Tab. 2, where $n, m, d_1, d_2$ represent the numbers of matrix rows, matrix columns, left features, right features, respectively; - in the table represents

Algorithm 1: Framework of MLRM.

**Input**: Side matrices $\boldsymbol{A}, \boldsymbol{C}$, the matrix to be completed $\boldsymbol{Y}$.
**Parameter**: Hyper-parameters $\lambda_1, \lambda_2$.
**Output**: $\boldsymbol{M}, \boldsymbol{N}$.

```
Calculate convert matrix L according to Eq. 11.
Calculate pairwise ranking matrix R = R_Ω(Y)L.
Update right side information matrix B = L^T C.
while not converged do
    E_k ← R − A N_k B^T;
    M_{k+1} ← D_{λ1}(E_k);
    F_k ← N_k − (1/L_p) A^T (M_{k+1} + A N_k B^T − R) B;
    N_{k+1} ← S_{λ2}(F_k);
    k ← k + 1;
end while
M ← M_k, N ← N_k;
return M, N.
```

that the corresponding value is unable to be obtained. The MovieLens, DTI, MLL datasets can be publicly obtained in (Harper and Konstan 2015), (Yamanishi et al. 2008) and the website of KDIS (http://www.uco.es/kdis/mllresources/), respectively.

We compare MLRM with 5 state-of-the-art methods, including 1 transductive matrix completion method (FPCA (Ma, Goldfarb, and Chen 2011)), 3 inductive methods (SIMC (Lu et al. 2016), FNNM (Yang, Li, and Wang 2020), InMC (Bertsimas and Li 2023)) and 1 multi-label learning method (SMDkM (Qian et al. 2023)). Note that in MovieRec and DTI, 4 matrix completion methods are taken, and in MLL, FPCA is removed and SMDkM (Qian et al. 2023) is taken. In MLL, the number of features is much larger than that of labels ($d_1 \gg m$), which makes it not advisable to recover the label matrix without features. Specifically, we take SMDkM to induce a multi-label model on the top of incomplete label matrix and then forecast a completed label matrix with feature information.

To conduct fair experiments, the defaults settings of comparison methods are recommended by corresponding papers. For MLRM, we grid search $10^{\{-1,\cdots,3\}}$ for hyper-parameters, and set $\lambda_1 = 10^2, \lambda_2 = 10^3$ for MovieRec, $\lambda_1 = \lambda_2 = 10$ for MLL and DTI. Besides, for all methods, tolerance to iteration termination is 0.01, maximum number of iterations is 1000.

| Scenario | Dataset | $n$ | $m$ | $d_1$ | $d_2$ |
|---|---|---|---|---|---|
| MovieRec | MovieLens | 943 | 1,682 | 23 | 20 |
| DTI | Enzymes | 445 | 664 | 445 | 664 |
| | GPCRs | 223 | 95 | 223 | 95 |
| MLL | Arts | 7,484 | 26 | 23,150 | - |
| | Business | 11,214 | 30 | 21,920 | - |
| | Computers | 12,444 | 33 | 34,100 | - |
| | Education | 12,030 | 33 | 27,530 | - |
| | Health | 9,205 | 32 | 30,610 | - |
| | Recreation | 12,828 | 22 | 30,320 | - |
| | Society | 14,512 | 27 | 31,800 | - |

Table 2: Benchmark datasets.

For MovieRec, we employ the famous MovieLens dataset (100K), where $n = 943$ users provide 100,000 ratings for $m = 1682$ movies, and $d_1 = 23$ user features and $d_2 = 20$ movie features are also given.

For DTI, 2 standard datasets are employed, where $n, m$ in above table are the numbers of drugs and targets. And the side information is drawn by drug similarity and target similarity, according to the inherent chemical structures in distinct drugs with SIMCOMP algorithm (Hattori et al. 2003).

For MLL, 7 widely used datasets are employed, where $n, m, d_1$ represent the numbers of instances, labels, features, respectively. The left side matrix is generated from the feature matrix $\boldsymbol{A} = qr(\boldsymbol{F})$, and the right one is set as identity matrix $\boldsymbol{C} = \boldsymbol{I}$ as no side information is provided for labels.

In each scenario, we set the proportion of the training data to $\omega\%$ and take 5-fold cross validation to determine the hyper-parameters. In the experiments, $\omega = \{10, 30, 50, 70, 90\}$ is set to simulate different observation rates.

Well-established metrics are used to measure the performance of MLRM and comparison methods. For MovieRec, we employ Mean Absolute Error (MAE) to evaluate the average absolute difference between the recovery values and the ground truth ones. For DTI, we use Area Under Curve (AUC) to measure the extremely imbalanced binary matrix. For MLL, Average Precision (AP) is employed to evaluate the correlation within labels (Zhang and Zhou 2007).

## 4.2 Experimental Results

We compare MLRM with baseline algorithms in this section. To conduct fair experiments, the random partition of training and testing sets for each benchmark dataset is performed 10 times, and the average values and standard deviations are reported. The comparison results for MovieRec, DTI and MLL are shown in Tab. 3 to 5. In MovieRec, $\text{MLRM}_p$ is taken to avoid the negative impact of the large scale matrix, and $p = 0.1$. In each condition of the displayed results, the algorithm ranked first is bolded, and the second ranked algorithm is underlined. Based on the experimental results, several conclusions can be drawn:

Based on the experimental results, several conclusions can be drawn:

- In summary, inductive methods outperform the transductive one (FPCA) in MovieRec and DTI, highlighting the effectiveness of side information learning.
- In MovieRec, MLRM ranks first in 2 sampling rates and secures the second position in the remaining 3 cases, and FNNM achieves the opposite record. It is noteworthy that movie recommendation often suffers from significant data missing. Consequently, it can be thus considered that MLRM is more practical than FNNM in real applications.
- In DTI, MLRM ranks first for 4 times, which is inferior to FNNM. Yet, MLRM ranks second on other 6 times and ranks 1.6 on average, which outperforms other 3 inductive methods, including 3.2 for SIMC, 1.9 for FNNM and 3.4 for InMC.

| $\omega\%$ | FPCA | SIMC | FNNM | InMC | $MLRM_p$ |
|---|---|---|---|---|---|
| 0.1 | 1.2809±0.0028 | 0.9982±0.0008 | <u>0.9502</u>±0.0016 | 1.0017±0.0008 | **0.8552**±0.0022 |
| 0.3 | 1.0912±0.0024 | 0.9894±0.0014 | <u>0.8799</u>±0.0019 | 1.0008±0.0014 | **0.8465**±0.0019 |
| 0.5 | 1.0081±0.0035 | 0.9784±0.0016 | **0.8453**±0.0027 | 1.0026±0.0016 | <u>0.8530</u>±0.0017 |
| 0.7 | 0.9665±0.0050 | 0.9625±0.0026 | **0.8201**±0.0032 | 1.0027±0.0031 | <u>0.8546</u>±0.0038 |
| 0.9 | 0.9454±0.0059 | 0.9414±0.0071 | **0.8046**±0.0039 | 1.0050±0.0057 | <u>0.8520</u>±0.0048 |

Table 3: MAE scores of MLRM and comparison methods in MovieLens.

| Dataset | $\omega\%$ | FPCA | SIMC | FNNM | InMC | MLRM |
|---|---|---|---|---|---|---|
| Enzymes | 0.1 | 0.4616±0.0179 | 0.5399±0.0045 | **0.6422**±0.0072 | 0.5339±0.0016 | <u>0.5836</u>±0.0068 |
| | 0.3 | 0.6064±0.0154 | 0.6047±0.0069 | **0.7439**±0.0053 | 0.6345±0.0032 | <u>0.6550</u>±0.0076 |
| | 0.5 | 0.6664±0.0097 | 0.6949±0.0070 | **0.8120**±0.0038 | 0.7345±0.0048 | <u>0.7528</u>±0.0078 |
| | 0.7 | 0.6991±0.0090 | 0.7927±0.0077 | **0.8517**±0.0076 | 0.7853±0.0056 | <u>0.8504</u>±0.0081 |
| | 0.9 | 0.7216±0.0095 | <u>0.8988</u>±0.0191 | 0.8957±0.0153 | 0.8221±0.0165 | **0.9257**±0.0089 |
| GPCRs | 0.1 | 0.4311±0.0246 | 0.5503±0.0122 | 0.5999±0.0135 | **0.6486**±0.0039 | <u>0.6459</u>±0.0192 |
| | 0.3 | 0.5089±0.0228 | 0.6416±0.0117 | **0.6903**±0.0052 | 0.6709±0.0057 | <u>0.6812</u>±0.0096 |
| | 0.5 | 0.6049±0.0192 | 0.7286±0.0163 | <u>0.7419</u>±0.0116 | 0.7017±0.0081 | **0.7466**±0.0093 |
| | 0.7 | 0.6592±0.0167 | <u>0.8136</u>±0.0161 | 0.7898±0.0098 | 0.7610±0.0169 | **0.8238**±0.0149 |
| | 0.9 | 0.6949±0.0248 | <u>0.8368</u>±0.0287 | 0.8269±0.0270 | 0.8069±0.0163 | **0.8720**±0.0315 |
| AvgRank | | 4.9 | 3.2 | 1.9 | 3.4 | 1.6 |

Table 4: AUC scores of MLRM and comparison methods in DTI.

| Dataset | $\omega\%$ | SMDkM | SIMC | FNNM | InMC | MLRM |
|---|---|---|---|---|---|---|
| Arts | 0.1 | 0.2824±0.0200 | 0.2917±0.0072 | 0.2796±0.0045 | <u>0.3022</u>±0.0047 | **0.3053**±0.0110 |
| | 0.3 | 0.4700±0.0057 | <u>0.6048</u>±0.0047 | 0.4838±0.0050 | 0.5080±0.0074 | **0.6066**±0.0059 |
| | 0.5 | 0.5853±0.0024 | <u>0.7345</u>±0.0037 | 0.6766±0.0033 | 0.6963±0.0058 | **0.7988**±0.0017 |
| | 0.7 | 0.6673±0.0018 | <u>0.8651</u>±0.0027 | 0.8483±0.0027 | 0.8528±0.0073 | **0.9174**±0.0020 |
| | 0.9 | 0.7213±0.0019 | <u>0.9782</u>±0.0011 | 0.9748±0.0016 | 0.9697±0.0038 | **0.9889**±0.0007 |
| Business | 0.1 | 0.4175±0.0191 | <u>0.5686</u>±0.0025 | 0.2839±0.0035 | 0.3040±0.0121 | **0.5895**±0.0209 |
| | 0.3 | 0.6137±0.0055 | <u>0.7710</u>±0.0019 | 0.5179±0.0029 | 0.5096±0.0123 | **0.8832**±0.0038 |
| | 0.5 | 0.7415±0.0027 | <u>0.8089</u>±0.0025 | 0.7222±0.0043 | 0.6917±0.0126 | **0.9526**±0.0010 |
| | 0.7 | 0.8102±0.0019 | <u>0.8834</u>±0.0021 | 0.8830±0.0022 | 0.8480±0.0090 | **0.9812**±0.0006 |
| | 0.9 | 0.8463±0.0017 | 0.9744±0.0008 | <u>0.9832</u>±0.0006 | 0.9608±0.0084 | **0.9975**±0.0003 |
| Computers | 0.1 | 0.3204±0.0124 | **0.4146**±0.0046 | 0.2495±0.0026 | 0.2620±0.0078 | <u>0.3600</u>±0.0116 |
| | 0.3 | 0.5252±0.0047 | <u>0.6914</u>±0.0049 | 0.4677±0.0041 | 0.4764±0.0160 | **0.7313**±0.0062 |
| | 0.5 | 0.6532±0.0023 | <u>0.7476</u>±0.0037 | 0.6683±0.0032 | 0.6684±0.0101 | **0.8656**±0.0020 |
| | 0.7 | 0.7358±0.0012 | <u>0.8557</u>±0.0015 | 0.8439±0.0022 | 0.8377±0.0076 | **0.9423**±0.0016 |
| | 0.9 | 0.7857±0.0012 | 0.9724±0.0011 | <u>0.9741</u>±0.0011 | 0.9636±0.0035 | **0.9916**±0.0007 |
| Education | 0.1 | 0.2853±0.0171 | **0.3229**±0.0057 | 0.2472±0.0029 | 0.2536±0.0091 | <u>0.2906</u>±0.0066 |
| | 0.3 | 0.5044±0.0047 | <u>0.5581</u>±0.0029 | 0.4586±0.0047 | 0.4635±0.0125 | **0.6425**±0.0056 |
| | 0.5 | 0.6260±0.0026 | <u>0.6775</u>±0.0026 | 0.6578±0.0051 | 0.6624±0.0124 | **0.8149**±0.0020 |
| | 0.7 | 0.7150±0.0026 | 0.8280±0.0027 | <u>0.8362</u>±0.0014 | 0.8348±0.0096 | **0.9225**±0.0013 |
| | 0.9 | 0.7630±0.0009 | 0.9692±0.0009 | <u>0.9719</u>±0.0007 | 0.9616±0.0087 | **0.9894**±0.0010 |
| Health | 0.1 | 0.3324±0.0120 | **0.4153**±0.0033 | 0.2597±0.0028 | 0.3024±0.0135 | <u>0.3564</u>±0.0147 |
| | 0.3 | 0.5700±0.0042 | <u>0.6699</u>±0.0036 | 0.4790±0.0022 | 0.4950±0.0305 | **0.7345**±0.0051 |
| | 0.5 | 0.7005±0.0024 | <u>0.7485</u>±0.0038 | 0.6846±0.0028 | 0.6815±0.0149 | **0.8762**±0.0018 |
| | 0.7 | 0.7908±0.0021 | <u>0.8645</u>±0.0026 | 0.8558±0.0019 | 0.8474±0.0176 | **0.9517**±0.0013 |
| | 0.9 | 0.8402±0.0006 | 0.9760±0.0010 | <u>0.9771</u>±0.0014 | 0.9642±0.0093 | **0.9937**±0.0003 |
| Recreation | 0.1 | 0.2812±0.0134 | <u>0.3013</u>±0.0061 | 0.2930±0.0024 | 0.2899±0.0043 | **0.3229**±0.0080 |
| | 0.3 | 0.4417±0.0053 | <u>0.5250</u>±0.0044 | 0.4980±0.0045 | 0.5006±0.0045 | **0.5834**±0.0050 |
| | 0.5 | 0.5499±0.0038 | 0.6709±0.0033 | 0.6909±0.0020 | <u>0.6913</u>±0.0038 | **0.7735**±0.0025 |
| | 0.7 | 0.6323±0.0026 | 0.8294±0.0026 | <u>0.8571</u>±0.0028 | 0.8497±0.0038 | **0.9030**±0.0018 |
| | 0.9 | 0.6881±0.0009 | 0.9702±0.0011 | <u>0.9776</u>±0.0014 | 0.9702±0.0019 | **0.9864**±0.0006 |
| Society | 0.1 | 0.2883±0.0104 | **0.3684**±0.0023 | 0.2810±0.0023 | 0.3036±0.0048 | <u>0.3659</u>±0.0105 |
| | 0.3 | 0.4705±0.0026 | <u>0.6250</u>±0.0038 | 0.4956±0.0018 | 0.5120±0.0049 | **0.6988**±0.0035 |
| | 0.5 | 0.5874±0.0031 | 0.6908±0.0036 | 0.6913±0.0029 | <u>0.6941</u>±0.0066 | **0.8315**±0.0031 |
| | 0.7 | 0.6679±0.0015 | 0.8328±0.0020 | <u>0.8577</u>±0.0026 | 0.8508±0.0048 | **0.9265**±0.0010 |
| | 0.9 | 0.7278±0.0007 | 0.9676±0.0009 | <u>0.9771</u>±0.0008 | 0.9707±0.0019 | **0.9890**±0.0005 |
| AvgRank | | 4.37 | 2.43 | 3.54 | 3.54 | 1.11 |

Table 5: AP scores of MLRM and comparison methods in MLL.

- In MLL, MLRM dominantly ranks first in 31 cases and ranks second in 4 cases among the total 35 ones over 7 datasets and 5 different sampling rates. And the shortcomings of MLRM are concentrated in the cases with low sampling rates. Fortunately, as a supervised learning paradigm, multi-label learning can generally collect the majority of labels and rarely faces the 90% missing labels.

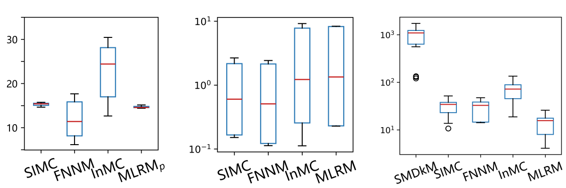
### 4.3 Runtime Comparison

MLRM employs alternating minimization algorithm to solve the model Eq. 15, and the procedure is organized as Alg. 1. Apparently, the computational complexity of MLRM depends on the number of iterations $T$ and the cost of each iteration.

In each iteration, the time complexity of nuclear norm minimization is $O(min(m^6, nm^4))$, which is determined by the SVD of $\boldsymbol{E} = \boldsymbol{R} - \boldsymbol{ANB}^T \in \mathbb{R}^{n\times\frac{m(m-1)}{2}}$, and the time complexity of $\ell_1$-norm minimization is $O(nm^2d_1 + m^2d_1d_2)$, which is determined by the matrix operation on $\boldsymbol{A}^T(\boldsymbol{M}+\boldsymbol{ANB}^T-\boldsymbol{R})\boldsymbol{B}$. Besides, in prediction and result correction, the time complexity is $O(nm^3)$ due to the matrix operations in Eq. 18 and Eq. 19. To sum up, the time complexity of MLRM is $O(T\cdot min(m^6, nm^4))$. And the space complexity of MLRM is $O(nm^2)$, limited by the scale of $\boldsymbol{R} \in \mathbb{R}^{n\times\frac{m(m-1)}{2}}$. Similarly for the existing inductive matrix completion methods summarized as Eq. 5, their time complexity and space complexity are $O(T\cdot min(m^3, nm^2))$ and $O(nm)$, respectively.

Fortunately, MLRM obtains less iteration rounds. MLRM converts the incomplete matrix $\boldsymbol{Y}$ into the pairwise matrix $\boldsymbol{R} = \mathcal{R}_\Omega(\boldsymbol{Y})\boldsymbol{L}$, and thus is able to directly handle the $\ell_1$ and nuclear norm as organized in Alg. 1. Conversely, in the traditional inductive matrix completion methods, the observation operator $\mathcal{R}_\Omega(\cdot)$ should be calculated in each iteration, and several learning rates should be incorporated. To ensure iterative convergence, these predefined learning rates cannot be too large, and the iteration rate is inevitably limited. To sum up, MLRM obtains longer strides and fewer iterations.

We compare the running time of MLRM and other inductive baselines, and the results are shown in the box charts in Fig. 1. To ensure experimental fairness, all programs are executed on Intel(R) Core(TM) i7-13650HX CPU @ 2.60GHz.



(a) MovieRec (b) DTI (c) MLL

Figure 1: Running time comparison among inductive methods (seconds).

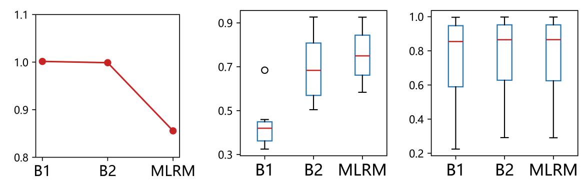
MLRM runs fastest in slender matrices ($n \gg m$, MLL), and obtains a competitive efficiency in small datasets (the scale of $m$ is limited, DTI) and large-scale matrix (MovieRec).

### 4.4 Ablation Study

In addition to multi-label ranking loss, MLRM introduces 2 manners to enhance the model performance, i.e., inductive learning on side information matrices and result correction. To verify the effectiveness of each part, an ablation study is conducted to compare MLRM with two baselines:

- B2: Removing the result correction based on MLRM;
- B1: Removing the inductive learning based on B2.

The ablation study result is shown as Fig. 2, where the ablation study on DTI and MLL is illustrated with box plot, and the ablation study on MovieRec is illustrated with the line chart of average value as only a single dataset with 6 sampling rates is revolved.



(a) MovieRec (b) DTI (c) MLL

Figure 2: Ablation study results of MLRM.

The comparison results in Fig. 2 demonstrates the performance ranking MLRM≻B2≻B1. It can also be found that the inductive learning manner has a limited improvement on MovieRec, while the result correction manner has a smaller impact on the performance of DTI and MLL. First, for MovieRec with few side features, the effectiveness of inductive learning manner is limited. Second, the evaluation metrics of DTI and MLL, i.e., AUC and AP, mainly concentrate on the ranking relation between true and predicted values rather than the absolute difference, while MAE (the evaluation metric for MovieRec) is the opposite. Therefore, the result correction manner plays an important role in MovieRec.

## 5 Limitations

MLRM assumes the ranking relation between the entries in matrices, which may not be applicable to the nominal values.

## 6 Conclusions

In this study, a multi-label ranking loss minimization (MLRM) method is proposed for matrix completion. It extends the ranking loss in multi-label classification into matrix completion, and effectively learns the relative and pairwise correlation among the vectors of the matrix to be completed. The experiments demonstrate the outstanding performance of MLRM comparing to state-of-the-art matrix completion methods in diverse scenarios. And more details of MLRM, including model implements, model convergence, convert matrix, supplementary experiments, can be referred to Codes and Appendix in https://github.com/JiaxuanGood/MLRM.git

## Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (No. xzy022023058), the National Natural Science Foundation of China (No. 72274152) and the China Scholarship Council program (No. 202406280214).

## References

Bach, F. R. 2008. Consistency of trace norm minimization. *The Journal of Machine Learning Research*, 9: 1019–1048.

Beck, A. 2015. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1): 185–209.

Bertsimas, D.; and Li, M. L. 2023. Interpretable matrix completion: A discrete optimization approach. *INFORMS Journal on Computing*.

Cabral, R.; Torre, F.; Costeira, J. P.; and Bernardino, A. 2011. Matrix completion for multi-label image classification. *Advances in neural information processing systems*, 24.

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.

Cai, Z.; Zhang, T.; and Wan, X.-F. 2011. Concepts and applications for influenza antigenic cartography. *Influenza and other respiratory viruses*, 5(Suppl 1): 204.

Chiang, K.-Y.; Hsieh, C.-J.; and Dhillon, I. S. 2015. Matrix completion with noisy side information. *Advances in neural information processing systems*, 28.

Combettes, P. L.; and Wajs, V. R. 2005. Signal recovery by proximal forward-backward splitting. *Multiscale modeling & simulation*, 4(4): 1168–1200.

Gibaja, E.; and Ventura, S. 2015. A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, 47(3): 1–38.

Goldberg, A.; Recht, B.; Xu, J.; Nowak, R.; and Zhu, J. 2010. Transduction with matrix completion: Three birds with one stone. *Advances in neural information processing systems*, 23.

Gu, S.; Zhang, L.; Zuo, W.; and Feng, X. 2014. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2862–2869.

Guo, H.; Yu, Q.; Zhang, X.; and Cheng, L. 2023. Low rank matrix minimization with a truncated difference of nuclear norm and frobenius norm regularization. *Journal of Industrial and Management Optimization*, 19(4): 2354–2366.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.

Hattori, M.; Okuno, Y.; Goto, S.; and Kanehisa, M. 2003. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39): 11853–11865.

Jannach, D.; Resnick, P.; Tuzhilin, A.; and Zanker, M. 2016. Recommender systems—beyond matrix completion. *Communications of the ACM*, 59(11): 94–102.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Lu, J.; Liang, G.; Sun, J.; and Bi, J. 2016. A sparse interactive model for matrix completion with side information. *Advances in neural information processing systems*, 29.

Luo, H.; Li, M.; Wang, S.; Liu, Q.; Li, Y.; and Wang, J. 2018. Computational drug repositioning using low-rank matrix approximation and randomized algorithms. *Bioinformatics*, 34(11): 1904–1912.

Ma, S.; Goldfarb, D.; and Chen, L. 2011. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2): 321–353.

Natarajan, N.; and Dhillon, I. S. 2014. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12): i60–i68.

Qian, K.; Min, X.-Y.; Cheng, Y.; Song, G.; and Min, F. 2023. Self-dependence multi-label learning with double k for missing labels. *Artificial Intelligence Review*, 56(6): 5057–5094.

Recht, B. 2011. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12).

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Schapire, R. E.; and Singer, Y. 2000. BoosTexter: A boosting-based system for text categorization. *Machine learning*, 39: 135–168.

Xie, M.-K.; and Huang, S.-J. 2018. Partial multi-label learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Xie, M.-K.; and Huang, S.-J. 2021. Partial multi-label learning with noisy label identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3676–3687.

Xu, M.; Jin, R.; and Zhou, Z.-H. 2013. Speedup matrix completion with side information: Application to multi-label learning. *Advances in neural information processing systems*, 26.

Yamanishi, Y.; Araki, M.; Gutteridge, A.; Honda, W.; and Kanehisa, M. 2008. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13): i232–i240.

Yang, M.; Li, Y.; and Wang, J. 2020. Feature and nuclear norm minimization for matrix completion. *IEEE Transactions on Knowledge and Data Engineering*, 34(5): 2190–2199.

Zhang, M.-L.; and Zhou, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7): 2038–2048.

Zhang, M.-L.; and Zhou, Z.-H. 2013. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8): 1819–1837.