Appendix of "Multi-Label Ranking Loss Minimization for Matrix Completion"

Jiaxuan Li,[1] Xiaoyan Zhu,[1,*] Hongrui Wang,[1] Yu Zhang,[1] Xin Lai, [1] Jiayin Wang[1]

[1] 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

MLRM

To ensure the integrity of Appendix, the objective function of the proposed MLRM is provided here,

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

The optimization algorithm of MLRM is,

Algorithm 1: Framework of MLRM.

Input: Side matrices $\boldsymbol{A}, \boldsymbol{C}$, the matrix to be completed $\boldsymbol{Y}$.
Parameter: Hyper-parameters λ_1, λ_2.
Output: $\boldsymbol{M}, \boldsymbol{N}$.

Calculate convert matrix $\boldsymbol{L}$.
Calculate pairwise ranking matrix $\boldsymbol{R} = \mathcal{R}_\Omega(\boldsymbol{Y})\boldsymbol{L}$.
Update right side information matrix $\boldsymbol{B} = \boldsymbol{L}^T\boldsymbol{C}$.
while not converged **do**
  $\boldsymbol{E}_k \leftarrow \boldsymbol{R} - \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T$;
  $\boldsymbol{M}_{k+1} \leftarrow \mathcal{D}_{\lambda_1}(\boldsymbol{E}_k)$;
  $\boldsymbol{F}_k \leftarrow \boldsymbol{N}_k - \frac{1}{L_p}\boldsymbol{A}^T(\boldsymbol{M}_{k+1} + \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T - \boldsymbol{R})\boldsymbol{B}$;
  $\boldsymbol{N}_{k+1} \leftarrow \mathcal{S}_{\lambda_2}(\boldsymbol{F}_k)$;
  $k \leftarrow k + 1$;
end while
$\boldsymbol{M} \leftarrow \boldsymbol{M}_k, \boldsymbol{N} \leftarrow \boldsymbol{N}_k$;
return $\boldsymbol{M}, \boldsymbol{N}$.

The Convergence of MLRM

Alternating minimization (AM) (Beck 2015) algorithm gives the condition of its convergence on objective function Eq. 2: firstly, $f(\boldsymbol{M}, \boldsymbol{N})$ is differentiable, and secondly, $g_1(\boldsymbol{M}), g_2(\boldsymbol{N})$ can be nondifferentiable but with subgradient.

$$\min_{\boldsymbol{M},\boldsymbol{N}} J(\boldsymbol{M}, \boldsymbol{N}) = f(\boldsymbol{M}, \boldsymbol{N}) + g_1(\boldsymbol{M}) + g_2(\boldsymbol{N}) \tag{2}$$

As shown in Eq. 1, the proposed MLRM method fully satisfies the above two conditions, with $f(\boldsymbol{M}, \boldsymbol{N}) = \frac{1}{2}\|\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{C}^T - \boldsymbol{R}\|_F^2, g_1(\boldsymbol{M}) = \lambda_1\|\boldsymbol{M}\|_*, g_2(\boldsymbol{N}) = \lambda_2\|\boldsymbol{N}\|_1$.

*indicates the corresponding author.

The Details on Convert Matrix

Convert Matrix

In this paper, the multi-label ranking loss is measured with the pairwise relation among columns. And the nonrepetitive column pairs is able to be represented by an ordered index pair set $S = \{(p, q)|1 \le p < q \le m\}$ and we take $S_j, 1 \le j \le \frac{m(m-1)}{2}$ to represent the j-th element of S. To make a clear description, let define S ordered, consisting of $m-1$ subsets, $S = \{S^1; S^2; \cdots; S^p; \cdots; S^{m-1}\}$,

$$S^1 = \{(1,2), (1,3), \cdots, (1,m)\}, |S^1| = m - 1 \tag{4a}$$
$$S^2 = \{(2,3), (2,4), \cdots, (2,m)\}, |S^1| = m - 2 \tag{4b}$$
$$\cdots = \cdots$$
$$S^p = \{(p,p+1), (p,p+2), \cdots, (p,m)\}, |S^p| = m - p \tag{4c}$$
$$\cdots = \cdots$$
$$S^{m-1} = \{(m-1, m)\}, |S^{m-1}| = 1 \tag{4d}$$

where $|S^p|$ is the capacity of S.

The pairwise ranking matrix $\boldsymbol{R}$ is defined due to the operation on $\boldsymbol{Y}$, the matrix to be completed. For instance, corresponding to S^1, the first $m-1$ columns of $\boldsymbol{R}$ can be represented as,

$$\boldsymbol{R}_{i1} = \boldsymbol{Y}_{i1} - \boldsymbol{Y}_{i2} \tag{5a}$$
$$\boldsymbol{R}_{i2} = \boldsymbol{Y}_{i1} - \boldsymbol{Y}_{i3} \tag{5b}$$
$$\cdots = \cdots$$
$$\boldsymbol{R}_{i,m-1} = \boldsymbol{Y}_{i1} - \boldsymbol{Y}_{im} \tag{5c}$$

Broadly, $\boldsymbol{R}_{ij}$ can be defined with the j-th element of S,

$$\boldsymbol{R}_{ij} = \boldsymbol{Y}_{ip} - \boldsymbol{Y}_{iq}, S_j = (p, q) \tag{6}$$

Formally, $\boldsymbol{R} = \mathcal{R}_\Omega(\boldsymbol{Y})\boldsymbol{L}$ transform the original matrix to the pairwise ranking matrix, where $\boldsymbol{L} \in \{-1, 0, 1\}^{m \times \frac{m(m-1)}{2}}$ is the convert matrix shown in Eq. 3, where elements that are not displayed represent 0.

To correspond to S, $\boldsymbol{L}$ can also be represented as $m-1$ submatrices, $\boldsymbol{L} = [\boldsymbol{L}^1; \boldsymbol{L}^2; \cdots; \boldsymbol{L}^p; \cdots; \boldsymbol{L}^{m-1}], \boldsymbol{L}^p \in \{-1, 0, 1\}^{m \times (m-p)}$ can be organized with three kinds of matrices, i.e., zero matrix or vector $\boldsymbol{0}$, row vector $\boldsymbol{1}$ and iden-

$$\boldsymbol{L} = \begin{array}{cc} & \begin{array}{ccccccccc} (1,2) & (1,3) & \cdots & (1,m) & (2,3) & \cdots & (2,m) & \cdots\cdots & (m-1,m) \end{array} \\ \begin{array}{c} 1\\ 2\\ 3\\ \vdots \\ m-1 \\ m \end{array} & \begin{pmatrix} 1 & 1 & \cdots & 1 & & & & & \\ -1 & & & & 1 & \cdots & 1 & & \\ & -1 & & & -1 & & & & \\ & & \ddots & & & \ddots & & & \\ & & & & & & & & 1 \\ & & & -1 & & & -1 & & -1 \end{pmatrix} \end{array} \tag{3}$$

tity matrix $\boldsymbol{I}$.

$$\boldsymbol{L} = \begin{array}{c} \begin{array}{ccccccc} S^1 & S^2 & \cdots & S^p & \cdots & S^{m-1} \end{array} \\ \begin{pmatrix} \phi & \mathbf{0}_{1,m-2} & & \mathbf{0}_{p-1,m-p} & & \mathbf{0}_{m-2,1} \\ \mathbf{1}_{1,m-1} & \mathbf{1}_{1,m-2} & & \mathbf{1}_{1,m-p} & & 1 \\ -\boldsymbol{I}_{m-1} & -\boldsymbol{I}_{m-2} & & -\boldsymbol{I}_{m-p} & & -1 \end{pmatrix} \end{array} \tag{7}$$

Pseudoinverse of Convert Matrix

As the pairwise ranking matrix $\boldsymbol{R}$ is approximated in the objective function, the final output is factually its predicted value $\hat{\boldsymbol{R}}$. To finally obtain the completed matrix $\hat{\boldsymbol{Y}}$, a new rule must be established as $\hat{\boldsymbol{Y}} = \hat{\boldsymbol{R}}\boldsymbol{L}^{-}$.

Recall the definition on convert matrix $\boldsymbol{R}$, its j-th column encodes the comparison between p-th and q-th columns of $\boldsymbol{Y}$, recording the superiority degree of $\boldsymbol{Y}_{ip}$ to $\boldsymbol{Y}_{iq}$ as a real value $\boldsymbol{R}_{ij}$. Consider that each column of $\boldsymbol{Y}$ is compared with the other $m-1$ columns, so that to decode $\hat{\boldsymbol{R}}$, we essentially should summarize $m-1$ superiority degrees in $\hat{\boldsymbol{R}}$ to 1 value in $\hat{\boldsymbol{Y}}$. Take the k-th column of $\hat{\boldsymbol{Y}}$ as an example, $m-1$ superiority degrees should be considered, corresponding to $m-1$ column pairs that include k, i.e., $(k,q),(p,k),1 \leq p < k < q \leq m$, where (k,q) denotes the superiority degree of the k-th column to the q-th one and (p,k) denotes the opposite. Formally, $\hat{\boldsymbol{Y}}_{ik}$ can be defined as,

$$\hat{\boldsymbol{Y}}_{ik} = \frac{1}{m-1}\left(\sum_{q,k<q}\left(\hat{\boldsymbol{R}}_{ik} - \hat{\boldsymbol{R}}_{iq}\right) - \sum_{p,p<k}\left(\hat{\boldsymbol{R}}_{ip} - \hat{\boldsymbol{R}}_{ik}\right)\right) \tag{8}$$

To compare Eq. 8 and Eq. 3, it can be concluded that,

$$\boldsymbol{L}^{-} = \frac{1}{m-1}\boldsymbol{L}^{T} \tag{9}$$

where $\boldsymbol{L}^{T}$ is the transpose of $\boldsymbol{L}$.

Besides, another interesting conclusion can also demonstrate the validity of $\boldsymbol{L}^{-} = \frac{1}{m-1}\boldsymbol{L}^{T}$ and $\hat{\boldsymbol{Y}} = \frac{1}{m-1}\hat{\boldsymbol{R}}\boldsymbol{L}^{T}$:

$$\begin{aligned} \frac{1}{m-1}\boldsymbol{L}\boldsymbol{L}^{T} &= \frac{1}{m-1}(m\boldsymbol{I}_m - 1) \\ &= \begin{bmatrix} 1 & -\frac{1}{m-1} & \cdots & -\frac{1}{m-1} \\ -\frac{1}{m-1} & 1 & \cdots & -\frac{1}{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{m-1} & -\frac{1}{m-1} & \cdots & 1 \end{bmatrix} \end{aligned} \tag{10}$$

That is, the main diagonal elements of $\boldsymbol{L}\boldsymbol{L}^{T}$ is 1, and the others are $-\frac{1}{m-1}$. Considering that the matrix $\boldsymbol{Y}$ to be completed is often in large scale, $-\frac{1}{m-1}$ is approximately 0, and $\frac{1}{m-1}\boldsymbol{L}^{T}$ can be viewed as the approximated pseudoinverse of $\boldsymbol{L}$. Factually in the Experiments, the error is negligible replacing $\boldsymbol{L}^{-} = \frac{1}{m-1}\boldsymbol{L}^{T}$ with $\boldsymbol{L}^{-} = pinv(\boldsymbol{L})$.

The Analysis on MLRM's Optimization Steps

To clearly explain the superiority of MLRM in runtime, there are a comparison between the optimization methods of MLRM and traditional matrix completion algorithms (Xu, Jin, and Zhou 2013; Natarajan and Dhillon 2014; Lu et al. 2016; Yang, Li, and Wang 2020). Take the SOTA algorithm FNNM (Yang, Li, and Wang 2020) as an example, whose objective function and optimization procedure are Eq. 11 and Alg. 2, has to incorporate 2 learning rates to tackle the observation operator $\mathcal{R}_{\Omega}(\cdot)$. To ensure iterative convergence, the 2 predefined learning rates cannot be too large, and the iteration rate is inevitably limited. In traditional matrix completion methods, subtle differences may exist in matrix factorization skills or regularization terms, but the nuclear norm minimization with observation operator is inevitable.

$$\min_{\boldsymbol{M},\boldsymbol{N}} \lambda_1\|\boldsymbol{M}\|_* + \lambda_2\|\boldsymbol{N}\|_1 + \frac{1}{2}\|\mathcal{R}_{\Omega}(\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{C}^T - \boldsymbol{Y})\|_F^2 \tag{11}$$

Algorithm 2: Framework of FNNM.

Input: Side matrices $\boldsymbol{A}, \boldsymbol{B}$, the matrix to be completed $\boldsymbol{Y}$.
Parameter: λ_1, λ_2, learning rates $\mu_1 = 0.1, \mu_2 = 0.1$.

while not converged **do**
  $\boldsymbol{E}_k \leftarrow \boldsymbol{M}_k - \mu_1\mathcal{R}_{\Omega}(\boldsymbol{M}_k + \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{C}^T - \boldsymbol{Y})$;
  $\boldsymbol{M}_{k+1} \leftarrow \mathcal{D}_{\lambda_1\mu_1}(\boldsymbol{E}_k)$;
  $\boldsymbol{F}_k \leftarrow \boldsymbol{N}_k - \mu_2\boldsymbol{A}^T\mathcal{R}_{\Omega}(\boldsymbol{M}_{k+1} + \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{C}^T - \boldsymbol{Y})\boldsymbol{B}$;
  $\boldsymbol{N}_{k+1} \leftarrow \mathcal{S}_{\lambda_2\mu_2}(\boldsymbol{F}_k)$;
  $k \leftarrow k + 1$;
end while
return $\boldsymbol{M}, \boldsymbol{N}$.

Conversely, 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 ℓ_1 and nuclear norm as organized in Alg. 1. Therefore, MLRM is able to obtain longer strides and fewer iterations.

Supplementary Experiments

In this paper, MLRM (multi-label ranking loss minimization) essentially proposes a novel loss function for matrix completion. To verify the extensibility and effectiveness of MLRM, we adapt it from Eq. 1 into several other formulations.

Transductive MLRM

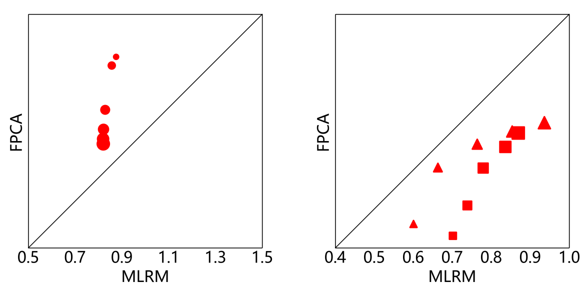
MLRM can be adapt into the situation without side information. In this case, transductive MLRM can be reformulated as:

$$\min_{\boldsymbol{M}} \frac{1}{2}\|\boldsymbol{M} - \boldsymbol{R}\|_F^2 + \lambda\|\boldsymbol{M}\|_* \tag{12}$$

The solution of Eq. 12 is

$$\boldsymbol{M} = \mathcal{D}_\lambda(\boldsymbol{R}) \tag{13}$$

The comparison between FPCA (Ma, Goldfarb, and Chen 2011) and transductive MLRM is shown in Fig. 1. In the figure, a dataset with a certain sampling rate is represented with a colored point, and the scale of a point implies the size of sampling rate. In each square, the horizontal and vertical coordinates of a point correspond to the performance of FPCA and transductive MLRM, respectively. For the scenario of movie recommendation, MLRM wins a case if the corresponding point is located to the left of the diagonal, as the smaller the MAE value, the better the algorithm performance, which is opposite for AUC in terms of DTI. In each figure, the points where MLRM performs better are marked in red, otherwise in blue. Besides, in the figure of DTI, the triangle points represent the first DTI dataset (Enzymes) and the square points represent the second one (GPCRs).



(a) MAE (MovieRec) (b) AUC (DTI)

Figure 1: The comparison between FPCA and transductive MLRM.

According to the comparison results, MLRM works well in the transductive situation.

MLRM in Global Pattern

In the assumption of FNNM, the matrix to be completed is partitioned into the combination of global and local patterns. We follow such assumption and propose MLRM in the formation of Eq. 1. Conversely, SIMC employs the assumption that the incomplete matrix is regarded as a low-rank matrix, which can also be induced via the side information learning. Such global pattern assumption is also commonly employed, and it is necessary to explore the performance of MLRM in this pattern. The objective function of such kind of methods like SIMC (Lu et al. 2016) is formulated as:

$$\begin{aligned} \min_{\boldsymbol{M},\boldsymbol{N}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{N}\boldsymbol{C}^T - \boldsymbol{M}\|_F^2 + \lambda_1\|\boldsymbol{M}\|_* + \lambda_2\|\boldsymbol{N}\|_1 \\ \text{s.t.} \quad & \mathcal{R}_\Omega(\boldsymbol{M}) = \mathcal{R}_\Omega(\boldsymbol{Y}) \end{aligned} \tag{14}$$

where $\boldsymbol{M}$ is the global pattern approximating the matrix to be completed.

MLRM is able to be simply adapted into such mode with the objective function:

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

The Eq. 15 can be solved with alternating minimization. Refer to Alg. 1, the calculation of $\boldsymbol{E}_k, \boldsymbol{F}_k$ should be changed to $\boldsymbol{E}_k \leftarrow \boldsymbol{R} + \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T$ and $\boldsymbol{F}_k \leftarrow \boldsymbol{N}_k - \frac{1}{L_p}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T - \boldsymbol{M}_{k+1})\boldsymbol{B}$, respectively.

The comparison between original MLRM and MLRM in global pattern is shown in Fig. 2. For MovieRec and DTI, we continue the legend declared in the previous section. In MLL, we divide the subfigures according to different sampling rates.

The comparison results in Fig. 2 demonstrate the availability of MLRM in global pattern. MLRM outperforms SIMC in the majority of situations.

MLRM with Noise Discrimination

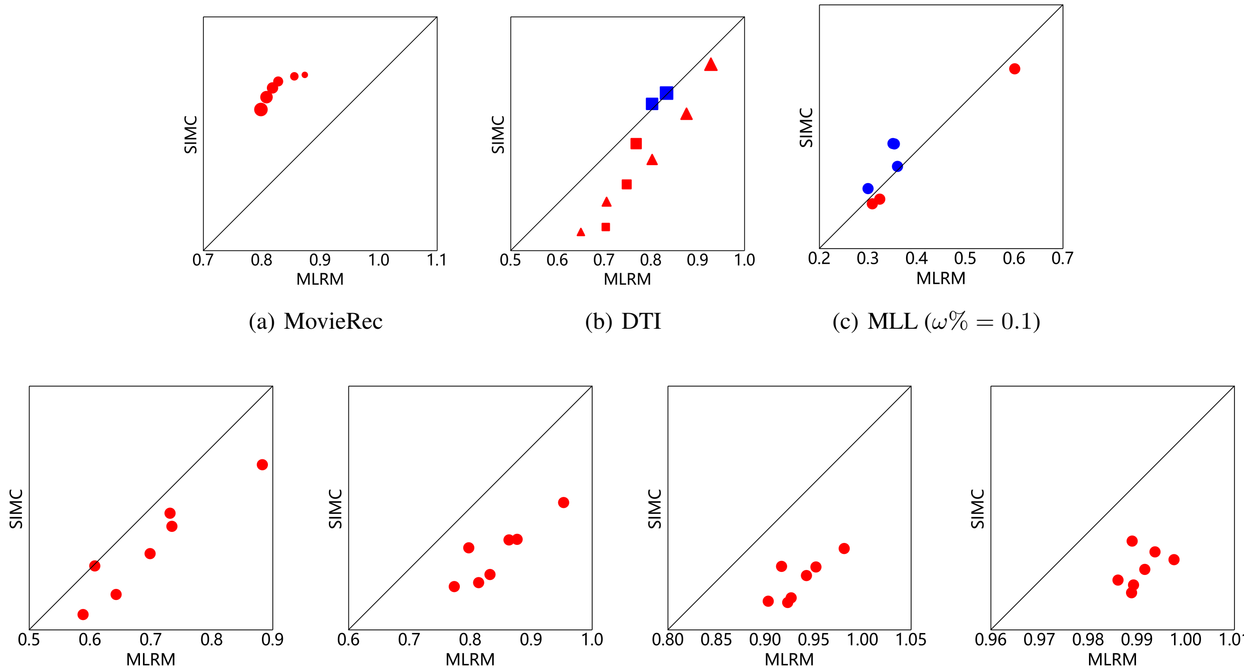
MLRM can be combined with matrix factorization techniques (Candès et al. 2011; Bao et al. 2012; Chen and Chen 2017; Sun et al. 2019, 2021; Xie and Huang 2021). For example, considering the noise in the matrix to be completed, the matrix to be approximated can be factorized as the sum of real matrix and noise matrix, i.e., $\boldsymbol{R} = \boldsymbol{R} + \boldsymbol{Z}$. In this case, the objective function can be reformulated as:

$$\begin{aligned} \min_{\boldsymbol{M},\boldsymbol{N},\boldsymbol{Z}} & \frac{1}{2}\|\boldsymbol{M} + \boldsymbol{A}\boldsymbol{N}\boldsymbol{B}^T + \boldsymbol{Z} - \boldsymbol{R}\|_F^2 \\ & + \lambda_1\|\boldsymbol{M}\|_* + \lambda_2\|\boldsymbol{N}\|_1 + \frac{\lambda_3}{2}\|\boldsymbol{Z}\|_F^2 \end{aligned} \tag{16}$$

The optimization method of Eq. 16 can also employ alternating minimization algorithm.

The comparison between original MLRM and MLRM with noise discrimination (MLRM′) is shown in Fig. 3.

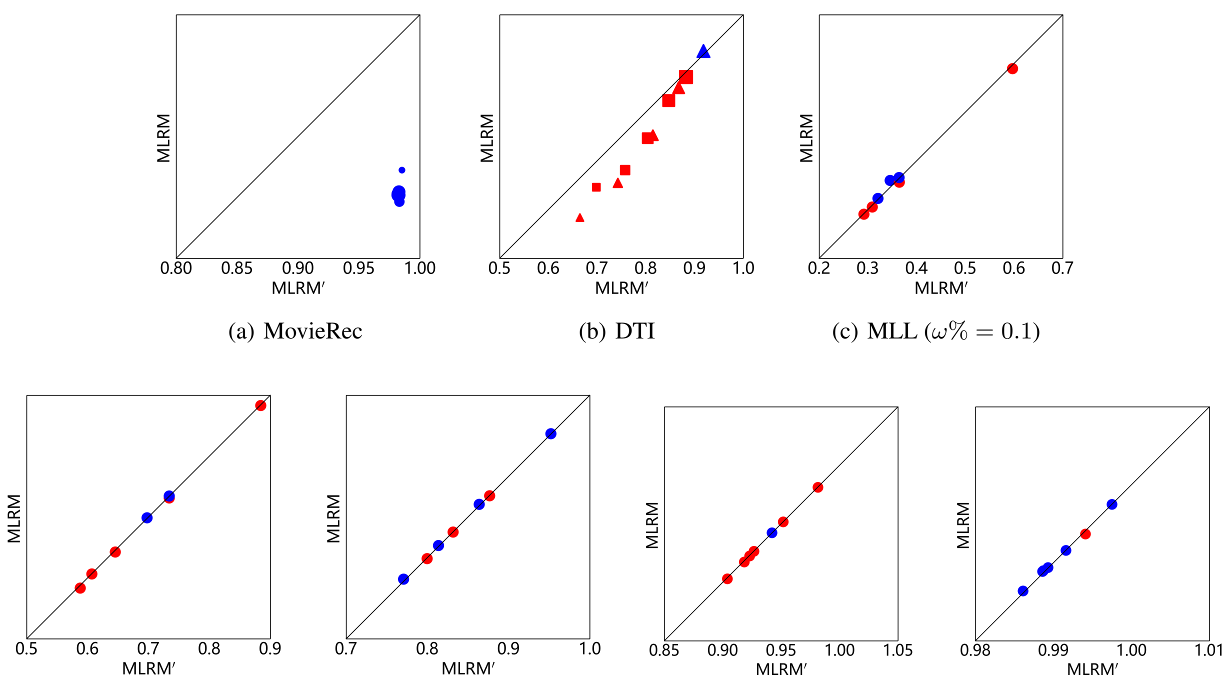
According to the comparison results, MLRM′ obtains different performance in distinct scenarios. In MovieRec, MLRM′ perform worse than MLRM because the movieLens dataset mainly faces the issue of extremely rare observed entries rather than noise data. In DTI, MLRM′ achieves a giant improvement, especially when sample rate is low. Due to the limited number of active drug target interactions, data sparsity is the main challenge to DTI, making learners tailed for negative class. The matrix noisy discrimination breaks the imbalance between positive and negative classes to some extent. In MLL, MLRM′ wins 20 cases among the all 35 ones, but the increase is not significant.



(a) MovieRec (b) DTI (c) MLL ($\omega\% = 0.1$)

(d) MLL ($\omega\% = 0.3$) (e) MLL ($\omega\% = 0.5$) (f) MLL ($\omega\% = 0.7$) (g) MLL ($\omega\% = 0.9$)

Figure 2: The comparison between original SIMC and MLRM in global pattern.



(a) MovieRec (b) DTI (c) MLL ($\omega\% = 0.1$)

(d) MLL ($\omega\% = 0.3$) (e) MLL ($\omega\% = 0.5$) (f) MLL ($\omega\% = 0.7$) (g) MLL ($\omega\% = 0.9$)

Figure 3: The comparison between original MLRM and MLRM with noise discrimination (MLRM′).

Table 1: AUC scores of MLRM′ and comparison methods in DTI.

ID	$\omega\%$	FPCA	SIMC	FNNM	InMC	MLRM′
Enzymes	0.1	0.4616±0.0179	0.5399±0.0045	0.6422±0.0072	0.5339±0.0016	**0.6647**±0.0068
	0.3	0.6064±0.0154	0.6047±0.0069	**0.7439**±0.0053	0.6345±0.0032	0.7426±0.0076
	0.5	0.6664±0.0097	0.6949±0.0070	0.8120±0.0038	0.7345±0.0048	**0.8143**±0.0078
	0.7	0.6991±0.0090	0.7927±0.0077	0.8517±0.0076	0.7853±0.0056	**0.8673**±0.0081
	0.9	0.7216±0.0095	0.8988±0.0191	0.8957±0.0153	0.8221±0.0165	**0.9183**±0.0089
GPCRs	0.1	0.4311±0.0246	0.5503±0.0122	0.5999±0.0135	0.6486±0.0039	**0.6985**±0.0192
	0.3	0.5089±0.0228	0.6416±0.0117	0.6903±0.0052	0.6709±0.0057	**0.7575**±0.0096
	0.5	0.6049±0.0192	0.7286±0.0163	0.7419±0.0116	0.7017±0.0081	**0.8040**±0.0093
	0.7	0.6592±0.0167	0.8136±0.0161	0.7898±0.0098	0.7610±0.0169	**0.8470**±0.0149
	0.9	0.6949±0.0248	0.8368±0.0287	0.8269±0.0270	0.8069±0.0163	**0.8826**±0.0315
AvgRank		4.9	3.2	2.3	3.5	1.1

Algorithm 3: Framework of MLRM with noise discrimination.

Input: Side matrices $\boldsymbol{A}, \boldsymbol{B}$, pairwise ranking matrix $\boldsymbol{R}$.
Parameter: Hyper-parameters $\lambda_1, \lambda_2, \lambda_3$.

1: **while** not converged **do**
2: $\quad \boldsymbol{Z}_{k+1} \leftarrow \frac{1}{1+\lambda_3}(\boldsymbol{R} - \boldsymbol{M}_k - \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T)$;
3: $\quad \boldsymbol{E}_k \leftarrow \boldsymbol{R} - \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T - \boldsymbol{Z}_{k+1}$;
4: $\quad \boldsymbol{M}_{k+1} \leftarrow \mathcal{D}_{\lambda_1}(\boldsymbol{E}_k)$;
5: $\quad \boldsymbol{F}_k \leftarrow \boldsymbol{N}_k - \frac{1}{L_p}\boldsymbol{A}^T(\boldsymbol{M}_{k+1} + \boldsymbol{A}\boldsymbol{N}_k\boldsymbol{B}^T + \boldsymbol{Z}_{k+1} - \boldsymbol{R})\boldsymbol{B}$;
6: $\quad \boldsymbol{N}_{k+1} \leftarrow \mathcal{S}_{\lambda_2}(\boldsymbol{F}_k)$;
7: **end while**
8: $\boldsymbol{M} \leftarrow \boldsymbol{M}_k, \boldsymbol{N} \leftarrow \boldsymbol{N}_k$;
9: **return** $\boldsymbol{M}, \boldsymbol{N}$.

Besides, in DTI, considering the significant improvement of MLRM with noise discrimination, we also gives its specific values here to compare with baselines in Tab. 1.

According to the comprehensive comparison in Tab. 1, it can be found that MLRM with noise discrimination has significantly outperformed the comparison methods.

Future Work

For the practical application, MLRM is available to the matrices with comparable elements, and is more suitable for slender matrices. In the future work, a more targeted multi-label ranking loss measurement method may need to be proposed to efficiently learn the relative correlation information in the incomplete matrix with limited amounts of pairwise ranking calculations.

References

Bao, B.-K.; Liu, G.; Xu, C.; and Yan, S. 2012. Inductive robust principal component analysis. *IEEE transactions on image processing*, 21(8): 3794–3800.

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.

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37.

Chen, L.; and Chen, S. 2017. Survey on matrix completion models and algorithms. *Journal of software*, 28(6): 1547–1564.

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.

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.

Sun, L.; Feng, S.; Liu, J.; Lyu, G.; and Lang, C. 2021. Global-local label correlation for partial multi-label learning. *IEEE Transactions on Multimedia*, 24: 581–593.

Sun, L.; Feng, S.; Wang, T.; Lang, C.; and Jin, Y. 2019. Partial multi-label learning by low-rank and sparse decomposition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5016–5023.

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.

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.