Robust Cost-Sensitive Learning for Recommendation with Implicit Feedback

Peng Yang * Peilin Zhao † Yong Liu ‡ Xin Gao *

Abstract

This paper aims at improvement on the effectiveness of matrix decomposition (MD) methods for implicit feedback. We highlight two critical limitations of existing works. First, due to the large number of unlabeled feedback, most existing works employ a uniform weight to the missing data to reduce computational complexity. However, such a uniform assumption may rarely hold in real-world scenarios. Second, the commonly-used bilateral loss function might be infinite if the data point is mis-classified. Outliers may have such issues and misguide the learning process. We address the above two issues by learning a robust asymmetric learning model. By leveraging the cost-sensitive learning and capped unilateral loss function, our robust MD objective function integrates them into a joint formulation, where the low-rank basis for user/item profiles can be modeled in an effective and robust way. Particularly, a novel log-determinant function is employed to refine the nuclear norm with respect to the low-rank approximation. We derive an iterative reweighted algorithm to efficiently minimize this MD objective, and also rigorously prove a lower error bound of the proposed algorithm compared to the 1-bit matrix completion method. Finally, we show the promising experimental results of our algorithm on benchmark recommendation datasets.

1 Introduction

User personalization has become increasingly important in current recommender systems. It aims to capture users' individual preferences, and increase both satisfaction for users and revenue for content providers. Among its various methods, matrix decomposition (MD) [1, 2] is the most widely used technique that characterizes users and items by vectors of latent factors. Early MD algorithms for recommendation have largely focused on *explicit feedback*, where users' ratings reflect their preference on items. However, explicit ratings are not always available in many real-world applications; more often, users interact with items through *implicit feedback*, e.g. users' viewing log and product purchase history. Compared to explicit ratings, implicit feedback is easier to collect, but more challenging to model due to the natural scarcity of negative feedback. It has been shown that modeling only the observed positive feedback leads to biased representation in user profiles [3, 4]. Moreover, users patronize the product/service they expect to like and avoid the genres they dislike, leading to a severe bias in the observed data.

To solve the problem of lacking negative feedback, one popular solution is to model all missing data as negative feedback [1]. However, this strategy misleads the learning process since some missing values are positives. Another solution is to select some negative instances from unlabeled entries [5, 6]. However, this adversely decreases the efficacy of the predictive model due to insufficient data coverage. To resolve this problem, He et al. [4] models all the missing data as negative feedback with a heuristic weight. Despite its success in dealing with implicit feedback, its performance is degraded due to the ignorance of the outlier issue. In [4], the conventional bilateral squared loss might be infinite if the data point is mis-classified. Outliers may have such issues and severely misguide the learning process. [7, 4]. To address this issue, self-paced learning [8] was applied in MD methods to reduce the impact of outliers. However, we argue that such methods cannot eliminate abnormal data points and may possibly involve the outliers in the learning process. In this work, we concern the above two challenging problems of the MD method — implicit feedback and outliers. Note that we are not the first to consider both aspects for MD models, as recent works by [9, 10] have proposed robust matrix completion methods on implicit feedback. Specifically, they assigned a uniform weight to the missing data, assuming that the missing entries are equally treated to be negative feedback. However, such an assumption rarely holds in practice, since the cost of missing a hidden positive is much higher than that of having a false-positive. In addition, the bilateral squared loss functions in their objectives is sensitive to the outliers and mis-classified data points.

In this work, we propose a novel MD model aimed at learning from implicit feedback in an effective and robust way. We develop a cost-sensitive learning method that efficiently optimizes the implicit MD model without imposing a uniform-weight restriction on missing data. In particular, we employ an asymmetric error cost for the positives and unlabeled data points [11, 12] which helps generalize label information from positive observations to the missing targets with similar learned representations. To be robust to outliers, we propose a capped unilateral loss function, which provides more robustness than existing MD models [8] by eliminat-

*KAUST, Saudi Arabia, {peng.yang.2,xin.gao}@kaust.edu.sa
†South China University of Technology, China peilinzhao@hotmail.com
‡Link Analytics Centre, NTUC Link, Singapore, liuysc@acm.org

ing the abnormal data points with large residues, such that the classification model training is robust to noise and incorrect labels. By leveraging cost-sensitive learning and capped unilateral loss function, our robust MD objective function integrates them into a joint formulation, where the low-rank basis for user/item profiles can be modeled in an effective and robust way. In previous study, the low-rank basis was formulated by the nuclear norm [13], which simply adds all nonzero singular values together instead of treating them equally as rank function does. To solve this issue, a novel *log-det* function is employed to reduce the contributions of big singular values to be close to 1 while keeping that of small singular values being 0. Based on this new MD objective, we design an iterative re-weighted algorithm to efficiently solve this problem. We show that our algorithm can be scaled up to the large-sized datasets after a relaxation. Theoretically, our method achieves a lower error bound than the state-of-the-art 1-bit matrix completion method [14]. Finally, the promising experimental results demonstrate the effectiveness of our algorithm on benchmark recommendation datasets.

2 Algorithm

A U-I rating matrix $\mathbf{Y} \in \mathbb{R}^{n\times m}$ includes n items, m users and $n \times m$ possible rating values. This matrix has a bounded value $\forall(i,j)\ Y_{ij} \in [0,1]$ and a bounded nuclear norm $\|\mathbf{Y}\|_* \leq \epsilon\ (\epsilon > 0)$, In the implicit setting, we have a 0-1 matrix $Y_{ij} = I_{(M_{ij}>q)}$, where I_π is the indicator function that outputs 1 if π holds and 0 otherwise, and $q \in [0,1]$ is a threshold. Assume that we observe only a subset of 1's of $\mathbf{Y}$. Let Ω be a subset of observed entries sampled from $\{(i,j)|Y_{ij}=1\}$, and S be the total number of 1's in $\mathbf{Y}$, then the sampling rate is $\rho = |\Omega|/S$. In particular, we define the observation matrix $\mathbf{A}$, where $A_{ij} = 1$ if $(i,j) \in \Omega$, and $A_{ij} = 0$ otherwise. The recommendation problem is defined as: given the observation U-I matrix $\mathbf{A}$, the objective is to recover $\mathbf{Y}$ based on this observed sampling. Inspired by the regularized loss minimization (RLM) [15], we aim to find a matrix $\mathbf{X}$ under some constraints to minimize the element-wise loss between $\mathbf{X}$ and $\mathbf{A}$ for each element:

$$\min_{X\in\mathcal{X}} \sum_{i=1}^{n}\sum_{j=1}^{m} \ell(X_{ij}, A_{ij}) + \lambda g(\mathbf{X}), \tag{2.1}$$

where the hypothesis space $\mathcal{X} := \{\mathbf{X} \in \mathbb{R}^{n\times m} \mid X_{ij} \leq 1, \text{if } A_{ij}=1; X_{ij} \geq 0, \text{if } A_{ij}=0\}$, $\lambda \geq 0$ is a trade-off parameter, $\ell(\mathbf{X},\mathbf{A})$ is a loss function between the matrices $\mathbf{X}$ and $\mathbf{A}$, and $g(\mathbf{X})$ is a convex regularization term that constrains $\mathbf{X}$ into simple sets, e.g. hyperplanes, balls, and bound constraints.

2.1 Robust Asymmetric Learning Framework

To solve the problem (2.1), existing MD techniques [4, 16] usually exploited squared loss based classification models. For classification tasks, if the data point x is correctly classified, i.e. $y(\mathbf{w}^\top\mathbf{x}+b)-1 \geq 0$, the loss should be zero. Such unilateral loss based classification models are more suitable for classification than the bilateral loss based classification models. However, as illustrated in Fig. 1(a), squared loss is a bilateral function, which do not meet this requirement. Motivated by the hinge loss, we define a novel loss function,

$$\xi(X_{ij}) = \max\{0, A_{ij} - [I_{(A_{ij}=+1)} - I_{(A_{ij}=0)}]X_{ij}\}.$$

We observe that $\xi(x)$ provides a unilateral loss for both positive ("+1") and unlabeled ("0") examples, which is more suitable for the implicit feedback problem.

2.1.1 Cost-Sensitive Learning

Although $\xi(x)$ is a unilateral function, it equally penalizes the mistakes on both classes. However, in the implicit feedback scenario, the cost of missing a positive target is much higher than that of having a false-positive. Thus, we study new MD techniques, which optimize a more appropriate performance metric, such as the *sum* of weighted *recall* and *specificity*,

$$sum = \mu_p \times recall + \mu_n \times specificity, \tag{2.2}$$

where $0 \leq \mu_p, \mu_n \leq 1$ and $\mu_p + \mu_n = 1$. In general, the higher the *sum* value, the better the performance. Besides, another suitable metric is the total cost of the algorithm [17]:

$$cost = c_p \times M_p + c_n \times M_n, \tag{2.3}$$

where M_p and M_n are the number of false negatives and false positives respectively, and $0 \leq c_p, c_n \leq 1$ are the cost parameters for positive and negative classes with $c_p + c_n = 1$. The lower the cost value, the better the classification performance.

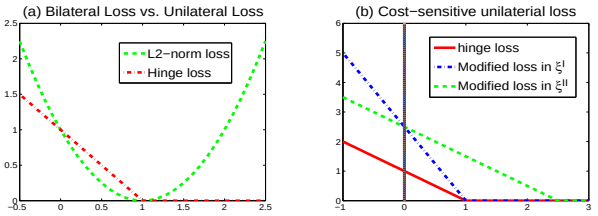
LEMMA 2.1. *The goal of maximizing the weighted sum in (2.2) or minimizing the weighted cost in (2.3) is equivalent to minimizing the following objective:*

$$\sum_{A_{ij}=+1} \alpha I_{(X_{ij}\leq q)} + \sum_{A_{ij}=0} I_{(X_{ij}>q)}, \tag{2.4}$$

where $\alpha = \frac{\mu_p T_n}{\mu_n T_p}$ for the weighted sum, T_p and T_n are the number of positive examples and negative examples, respectively; and $\alpha = \frac{c_p}{c_n}$ for the weighted misclassification cost.

Proof. By analyzing the function of the weighted sum in (2.2), we can derive the following $sum = \mu_p\frac{T_p - M_p}{T_p} + \mu_n\frac{T_n - M_n}{T_n} = 1 - \frac{\mu_n}{T_n}\{\frac{\mu_p T_n}{\mu_n T_p}\sum_{A_{ij}=+1} I_{(X_{ij}\leq q)} + \sum_{A_{ij}=0} I_{(X_{ij}>q)}\}$. Thus, maximizing *sum* is equivalent to minimizing $\frac{\mu_p T_n}{\mu_n T_p}\sum_{A_{ij}=+1} I_{(X_{ij}\leq q)} + \sum_{A_{ij}=0} I_{(X_{ij}>q)}$. Secondly, by analyzing the function of the weighted cost

Figure 1: Cost-sensitive unilateral loss



in (2.3), we can derive the following: $cost = c_p \times M_p + c_n \times M_n = c_n\{\frac{c_p}{c_n}\sum_{A_{ij}=+1} I_{(X_{ij}\leq q)} + \sum_{A_{ij}=0} I_{(X_{ij}>q)}\}$. Thus, minimizing *cost* is equivalent to minimizing $\sum_{A_{ij}=+1} \frac{c_p}{c_n} I_{(X_{ij}\leq q)} + \sum_{A_{ij}=0} I_{(X_{ij}>q)}$. Thus, the lemma holds by setting $\alpha = \frac{\mu_p T_n}{\mu_n T_p}$ for sum and $\alpha = \frac{c_p}{c_n}$ for cost. □

Remark: The values of $\frac{T_n}{T_p}$ might be unknown in advance. To alleviate this issue, an alternative way is to consider the *cost* for performance evaluation. Specifically, we set $\alpha = \frac{C_p}{C_n}$ with $C_p + C_n = 1$. We assume $0 \leq C_n \leq C_p$, since we would prefer to improve the accuracy of the positive class. When $\alpha > 1$, it tries to train a model with the positive instances first, then gradually improves it with the unlabeled instances, which is very intuitive for learning a more balanced model.

Lemma 2.1 gives the explicit objective for optimization, but the indicator function is non-convex. To address this issue, we derive two cost-sensitive loss functions by replacing Eq. (2.4) with the convex loss $\xi(x)$:

$$\xi^I_\alpha(X_{ij}) = S_{ij}\max\{0, A_{ij} - [I_{(A_{ij}=+1)} - I_{(A_{ij}=0)}]X_{ij}\},$$
$$\xi^{II}_\alpha(X_{ij}) = \max\{0, S_{ij}A_{ij} - [I_{(A_{ij}=+1)} - I_{(A_{ij}=0)}]X_{ij}\},$$

where the weighted cost $S_{ij} = \alpha I_{(A_{ij}=+1)} + I_{(A_{ij}=0)}$. Fig. 1(b) illustrates the difference of the cost-sensitive loss functions for the cases $A_{ij} = 1$ when $\alpha = 2.5$. We observe that for $\xi^I_\alpha(x)$, the slope of the loss function changes for a specific class, leading to more "aggressive" updating; for $\xi^{II}_\alpha(x)$, the required margin for specific class changes comparing to the traditional hinge loss, resulting in more "frequent" updating.

2.1.2 Robust Capped Loss The cost-sensitive loss $\xi_\alpha(x)$ might be infinite if the data point is not correctly classified. To be resistant to outliers, we propose to solve the problem with a new loss function to robustly learn a classifier:

$$\min_{\mathbf{X}\in\mathcal{X}} \sum_{i,j} \min(\xi_\alpha(X_{ij}), \varepsilon_\alpha(A_{ij})) + \lambda g(\mathbf{X}), \quad \text{s.t. } \varepsilon_\alpha(A_{ij}) = (\alpha I_{(A_{ij}=+1)} + I_{(A_{ij}=0)})\varepsilon \tag{2.5}$$

where $\varepsilon > 0$. To provide robustness, the objective (2.5) enforces a capped bound over the cost-sensitive loss $\xi_\alpha(x)$. This makes the loss function robust to outliers since their impact to the model is upper bounded by $\varepsilon_\alpha(a) > 0$.

It is nontrivial to solve the objective (2.5) since the $\min(u, \varepsilon_\alpha(a))$ is a concave function in the domain of $u = \xi_\alpha(x)$. Fortunately, Lemma 2.2 provides a way to solve this problem.

LEMMA 2.2. *Motivated by concave duality [18], the problem (2.5) can be relaxed to iteratively minimizing a weighted cost-sensitive loss formulation:*

$$\mathbf{X}^{t+1} = \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X}} \sum_{i,j} \Psi^t_{ij}\xi_\alpha(X_{ij}) + \lambda g(\mathbf{X}), \tag{2.6}$$

where $\Psi = \nabla_u \min(u, \varepsilon_\alpha(a))|_{u=\xi_\alpha(x)}$ *is the supergradient of the concave function* $\min(u, \varepsilon_\alpha(a))$ *at* $u = \xi_\alpha(x)$*:*

$$\Psi^t_{ij} = I(\xi_\alpha(X^t_{ij}) \leq \varepsilon_\alpha(A_{ij})). \tag{2.7}$$

Intuitively, the mis-classified points with the large residues $\xi_\alpha(x) > \varepsilon_\alpha(a)$ *are considered as outliers, and ignored.*

Proof. The proof is in Supplementary Material. □

THEOREM 1. *The solution* $\{\mathbf{X}^t\}$ *generated by (2.6) minimizes the upper bound of the problem* (2.5) *iteratively.*

Proof. Denoted by $h(\xi_\alpha(x)) = \min(\xi_\alpha(x), \varepsilon_\alpha(a))$ as a concave function, for any x we have

$$h(\xi_\alpha(x)) \leq h(\xi_\alpha(x^t)) + \langle \Psi^t, \xi_\alpha(x) - \xi_\alpha(x^t)\rangle,$$

where $\Psi^t = \nabla_u \min(u, \varepsilon)|_{u=\xi_\alpha(x^t)}$. Thus we obtain an upper bound of (2.5) via a linear approximation at $\xi_\alpha(\mathbf{X}^t)$:

$$\begin{aligned}&\forall \mathbf{X} \in \mathbb{R}^{n\times m} : \lambda g(\mathbf{X}) + h(\xi_\alpha(\mathbf{X}))\\ \leq &\lambda g(\mathbf{X}) + h(\xi_\alpha(\mathbf{X}^t)) + \langle \mathbf{\Psi}^t, \xi_\alpha(\mathbf{X}) - \xi_\alpha(\mathbf{X}^t)\rangle_F.\end{aligned}$$

Since $\xi_\alpha(\mathbf{X}^t)$ is constant w.r.t. $\mathbf{X}$, we have

$$\begin{aligned}&\mathbf{X}^{t+1} =\\ &\arg\min_{\mathbf{X}} \lambda g(\mathbf{X}) + h(\xi_\alpha(\mathbf{X}^t)) + \langle \mathbf{\Psi}^t, \xi_\alpha(\mathbf{X}) - \xi_\alpha(\mathbf{X}^t)\rangle_F\\ &= \arg\min_{\mathbf{X}} \lambda g(\mathbf{X}) + \langle \mathbf{\Psi}^t, \xi_\alpha(\mathbf{X})\rangle_F,\end{aligned}$$

which, as illustrated in Eq. (2.6), obtains an iterative solution to minimize the upper bound of the problem (2.5). □

2.2 Framework Instantiation to Matrix Decomposition

We intend to cooperate the capped cost-sensitive objective (2.6) into matrix decomposition to derive a low-rank representation. To achieve this goal, we propose a robust matrix decomposition, assuming that the observation $\mathbf{A}$ can be decomposed into a low-rank matrix $\mathbf{U}$ and an outlier matrix $\mathbf{V}$. We begin with a function $f(\cdot)$, $\mathbf{X} = f(\mathbf{W}) =$

$\left[\mathbf{I}_n, \mathbf{I}_n\right] \mathbf{W} = \mathbf{U} + \mathbf{V}$, where $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix and the matrix $\mathbf{W}$ is decomposed into two components: $\{\mathbf{W}|\mathbf{W} = \begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix} \in \mathbb{R}^{2n\times m}, \mathbf{U} \in \mathbb{R}^{n\times m}, \mathbf{V} \in \mathbb{R}^{n\times m}\}$. The problem can be formulated as follows: given that $\mathbf{U}$ and $\mathbf{V}$ are unknown, and $\mathbf{U}$ is known to be low-rank while $\mathbf{V}$ is known to be sparse, we recover $\mathbf{A}$ with $\mathbf{X} = \mathbf{U} + \mathbf{V}$,

$$\min_{\mathbf{W}} \ \lambda\langle \mathbf{\Psi}, \xi_\alpha(f(\mathbf{W}))\rangle_F + \text{rank}(\mathbf{U}) + \|\mathbf{V}\|_0. \tag{2.8}$$

However, the objective (2.8) is a highly nonconvex optimization problem [19]. To solve (2.8), we relax the rank function rank($\mathbf{U}$) and the L_0-norm $\|\mathbf{V}\|_0$, respectively. For the rank function, previous work tries to replace it by the nuclear norm [13]. However, as shown in Fig. 2(a), the nuclear norm simply adds all nonzero singular value together instead of treating them equally as the rank function does. Fazel *et al.* [20] proposed a nonconvex form $\log\det(\mathbf{U} + \delta\mathbf{I})$ for a rank approximation. However, it is restrictive to a positive semidefinite matrix $\mathbf{U}$ and biases the estimation due to a small parameter δ [21]. In Definition 1, we introduce a novel *logdet* function that can guarantee a more general $\mathbf{U}$ with a simple solution.

DEFINITION 1. *Let $\{\sigma_i\}_{i=1}^r$ be the singular values of the matrix $\mathbf{U}$, we propose a* logdet *function,*

$$r(\mathbf{U}) = \log\det(\mathbf{I} + \sqrt{\mathbf{U}^\top\mathbf{U}}) = \sum_{i=1}^{r}\log(1 + |\sigma_i|),$$

which, as shown in Fig. 2(b), has following traits:
a) When $\sigma_i = 0$, the term $\log(1 + |\sigma_i|) = 0$, which is the same as the true rank function;
b) When $0 < \sigma_i < 1$, $\log(1 + |\sigma_i|) < \sigma_i$, implying small singular values can be further reduced to be close to 0;
c) For a large value $\sigma_i > 1$, $\log(1 + |\sigma_i|) \ll \sigma_i$, which is a significant reduce over large singular values.

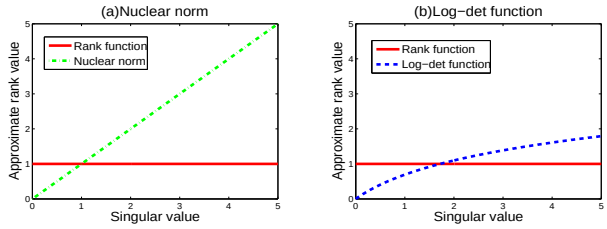
For the L_0-norm of $\mathbf{V}$, [22] tries to relax it with L_1-norm. We advocate using this L_1-norm relaxation for its computational efficiency. Equipped $r(\mathbf{U})$ and L_1-norm of $\mathbf{V}$ into Eq. (2.8), the objective solves a slightly relaxed version,

$$\min_{\mathbf{W}} \ \langle \mathbf{\Psi}, \xi_\alpha(f(\mathbf{W}))\rangle_F + \lambda_1\sum_{i=1}^{r}\log(1 + |\sigma_i|) + \lambda_2\|\mathbf{V}\|_1, \tag{2.9}$$

where λ_1 and λ_2 are trade-off parameters.

2.3 Optimization Directly optimizing the objective (2.9) can be computationally expensive [23], since it contains two high-dimensional matrices. Motivated by [24], we solve (2.9) with an accelerated proximal gradient line (APGL) method, which enjoys a convergence rate of $O(1/t^2)$, where t is the number of iterations. We begin with the first-order

Figure 2: Nuclear norm vs. Log-det function



Taylor expansion of $\langle \mathbf{\Psi}, \xi_\alpha(f(\mathbf{W}))\rangle_F$ at $\mathbf{W} = \mathbf{W}^t$:

$$\xi_{\eta,\mathbf{W}^t}(\mathbf{W}) = \langle \mathbf{\Psi}^t, \xi_\alpha(f(\mathbf{W}^t))\rangle_F + \frac{\mathcal{D}(\mathbf{W}, \mathbf{W}^t)}{2\eta} + \langle \mathbf{\Psi}^t \circ \nabla\xi_\alpha(f(\mathbf{W}^t)), \mathbf{W} - \mathbf{W}^t\rangle_F,$$

where $\circ$ is entrywise product, $\mathbf{\Psi}^t$ is the re-weighted matrix at $\mathbf{X} = \mathbf{X}^t$, $\nabla\xi_\alpha(f(\mathbf{W}^t))$ is the derivative of $\xi_\alpha(f(\mathbf{W}))$ at $\mathbf{W} = \mathbf{W}^t$, and $\mathcal{D}(\cdot,\cdot)$ measures the Euclidean distance between variables. Equipped with $\xi_{\eta,\mathbf{W}^t}(\mathbf{W})$, the objective (2.9) can be solved iteratively:

$$\mathbf{W}^{t+1} = \operatorname*{argmin}_{\mathbf{W}} \xi_{\eta,\mathbf{W}^t}(\mathbf{W}) + \lambda_1 r(\mathbf{U}) + \lambda_2\|\mathbf{V}\|_1. \tag{2.10}$$

LEMMA 2.3. *Given that $\mathbf{W} = \begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}$, the objective (2.10) can be iteratively solved with $\mathbf{U}$ and $\mathbf{V}$, respectively:*

$$\tilde{\mathbf{U}}^{t+1} = \operatorname*{argmin}_{\mathbf{U}} \frac{1}{2}\|\mathbf{U} - \hat{\mathbf{U}}^t\|_F^2 + \eta\lambda_1 r(\mathbf{U}),$$

$$\tilde{\mathbf{V}}^{t+1} = \operatorname*{argmin}_{\mathbf{V}} \frac{1}{2}\|\mathbf{V} - \hat{\mathbf{V}}^t\|_F^2 + \eta\lambda_2\|\mathbf{V}\|_1,$$

where $\hat{\mathbf{U}}^t = \mathbf{U}^t - \eta\nabla_{\mathbf{U}^t}\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t$ and $\hat{\mathbf{V}}^t = \mathbf{V}^t - \eta\nabla_{\mathbf{V}^t}\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t$.

Proof. The proof is in Supplementary Material. □

Computation of $\tilde{\mathbf{U}}$: Inspired by [25], we show that the solution to $\tilde{\mathbf{U}}$ can be obtained via solving a simple convex optimization problem. Assume the eigendecomposition of $\hat{\mathbf{U}}^t = \mathbf{P}\hat{\mathbf{\Sigma}}^t\mathbf{Q}^\top$ where r =rank($\hat{\mathbf{U}}^t$) and $\hat{\mathbf{\Sigma}}^t = \text{diag}(\hat{\sigma}_i^t)_{i=1}^r \in \mathbb{R}^{r\times r}$. Let $\mathbf{U} = \mathbf{P}\text{diag}\{\sigma_i\}_{i=1}^r\mathbf{Q}^\top$ with $\sigma_i \geq 0$ and $\rho = \eta\lambda_1$, the problem for solving $\tilde{\mathbf{U}}$ can turn into an equivalent form,

$$\min_{\{\sigma_i\}_{i=1}^r} \sum_{i=1}^{r}\left[\Theta(\sigma_i) = \frac{1}{2}(\sigma_i - \hat{\sigma}_i^t)^2 + \rho\log(1 + \sigma_i)\right]. \tag{2.11}$$

According to the first-order optimality condition, the gradient of the objective function of (2.11) w.r.t. each σ_i should vanish. For the *loget* function, we have

$$\frac{1}{\rho}\sigma_i^2 - \frac{\hat{\sigma}_i^t - 1}{\rho}\sigma_i - \frac{\hat{\sigma}_i^t - \rho}{\rho} = 0, \text{ s.t. } \sigma_i \geq 0, \ \rho = \eta\lambda_1.$$

The above equation is quadratic and gives two roots. If $\hat{\sigma}_i^t = 0$, the minimizer σ_i^* will be 0; otherwise, there exists a unique minimizer. Finally we obtain the update of $\mathbf{U}$ with

$$\tilde{\mathbf{U}}^{t+1} = \mathbf{P}\text{diag}(\sigma_i^*)_{i=1}^r \mathbf{Q}^\top. \tag{2.12}$$

Computation of $\tilde{\mathbf{V}}$: The problem of $\tilde{\mathbf{V}}$ is a Lasso problem and admits a closed-form solution for each entry,

$$\tilde{V}_{ki}^{t+1} = \mathcal{K}\left(\left[\mathbf{V}^t - \eta\nabla_{\mathbf{V}^t}\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t\right]_{ki}, \eta\lambda_2\right), \tag{2.13}$$

where $[\cdot]_{ki}$ returns the (k,i)-th element of a matrix, $\text{sgn}(\cdot)$ defines the sign function, $|\cdot|$ gives the absolute value when the argument is a scalar, and $\mathcal{K}(a,b) = [|a| - b]_+\text{sign}(a)$. Specifically, if $\mathbf{v}_i = \mathbf{0}$, the user will obey only $\mathbf{u}_i$; otherwise, non-zero entries hold in $\mathbf{v}_i$, and $\mathbf{x}_i = \mathbf{v}_i + \mathbf{u}_i$.

2.3.1 Algorithm We summarize this cost-sensitive robust recommendation model, CSRR, in Alg. 1. It optimizes a matrix $\mathbf{X}$ iteratively based on two sequences $\{\mathbf{X}^t\}$ and $\{\tilde{\mathbf{X}}^t\}$ with a coefficient $C_{t+1} = (1 + \sqrt{1 + 4C_t^2})/2$:

$$\begin{aligned} \mathbf{U}^{t+1} &= \tilde{\mathbf{U}}^{t+1} + \frac{C_t - 1}{C_{t+1}}(\tilde{\mathbf{U}}^{t+1} - \tilde{\mathbf{U}}^t), \\ \mathbf{V}^{t+1} &= \tilde{\mathbf{V}}^{t+1} + \frac{C_t - 1}{C_{t+1}}(\tilde{\mathbf{V}}^{t+1} - \tilde{\mathbf{V}}^t), \end{aligned} \tag{2.14}$$

where $\{\tilde{\mathbf{X}}^{t+1}\} = \prod_\pi(\mathbf{X}^t - \eta\nabla\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t)$ with $\prod_\pi(\mathbf{G}) = \min_{\mathbf{X}\in\pi} \frac{1}{2}\|\mathbf{X} - \mathbf{G}\|_F^2$ is the sequence of approximate solution, and $\{\mathbf{X}^{t+1}\}$ in (2.14) is the sequence of search points. Note that $\mathbf{X} = f(\mathbf{W}) = \mathbf{U} + \mathbf{V}$ and $\frac{\partial \mathbf{X}}{\partial \mathbf{U}} = \frac{\partial \mathbf{X}}{\partial \mathbf{V}} = \mathbf{I}$, we have $\nabla_{\mathbf{V}}\xi_\alpha(\mathbf{X}) = \nabla_{\mathbf{U}}\xi_\alpha(\mathbf{X}) = \nabla_{\mathbf{X}}\xi_\alpha(\mathbf{X})$. Moreover, $\xi_{\eta,\mathbf{W}^t}(\mathbf{W})$ can be an upper bound to search a feasible point (Step 13 of Alg. 1). In particular, when using the loss $\xi_\alpha^I(x)$, we refer to the algorithm above as "CSRR-I"; when using $\xi_\alpha^{II}(x)$, we refer to the algorithm as "CSRR-II".

2.4 An Efficient Optimization Though CSRR can perform stably without knowing the target rank in advance, it is limited by the necessity of executing singular value decomposition (SVD) for multiple times (Step 11 of Alg. 1). At less expense, bilinear factorization (BF) [26, 27] is an alternative by replacing $\mathbf{U}$ with $\mathbf{P}\mathbf{Q}^\top$, where the product of two factor matrices $\mathbf{P} \in \mathbb{R}^{n\times d}$ and $\mathbf{Q} \in \mathbb{R}^{m\times d}$ implicitly guarantees that the rank of $\mathbf{P}\mathbf{Q}^\top$ is never over d, typically $d \ll \min(m,n)$. Theorem 2 provides a bridge between the nuclear norm minimization and BF models.

THEOREM 2. *For any matrix $\mathbf{U} \in \mathbb{R}^{n\times m}$, the following relationship holds [28]:*

$$\|\mathbf{U}\|_* = \min_{\mathbf{P},\mathbf{Q}} \frac{1}{2}\|\mathbf{P}\|_F^2 + \frac{1}{2}\|\mathbf{Q}\|_F^2 \quad s.t. \quad \mathbf{U} = \mathbf{P}\mathbf{Q}^\top.$$

If rank$(\mathbf{U}) = d \le \min(m,n)$, then the minimum solution above is attained at a factor decomposition $\mathbf{U} = \mathbf{P}\mathbf{Q}^\top$, where $\mathbf{P} \in \mathbb{R}^{n\times d}$ and $\mathbf{Q} \in \mathbb{R}^{m\times d}$.

Algorithm 1 CSRR

1: **Input**: $\mathbf{A} \in \mathbb{R}^{n\times m}$, scalars T, λ_1, λ_2 and α.
2: **Output**: $\mathbf{X}^T$, $\mathbf{U}^T$ and $\mathbf{V}^T$;
3: **Initialize**: $\tilde{\mathbf{U}}^1 = \tilde{\mathbf{U}}^0, \tilde{\mathbf{V}}^1 = \tilde{\mathbf{V}}^0, C_0 = 0$ and η_1;
4: **for** $t = 1, \ldots, T$ **do**
5: Set $C_t = (1 + \sqrt{1 + 4C_{t-1}^2})/2$;
6: **Update**: $\mathbf{X}^t = \mathbf{U}^t + \mathbf{V}^t$ with Eq. (2.14);
7: **Update**: $\mathbf{\Psi}^t$ with Eq. (2.7);
8: **for** $k = 0, 1$ to $\ldots$ **do**
9: Set $\eta = 2^{-k}\eta_t$;
10: Compute $\xi_\alpha^{\{I,II\}}(\mathbf{X}^t)$ and $\nabla\xi_\alpha^{\{I,II\}}(\mathbf{X}^t)$;
11: $\tilde{\mathbf{U}}^{t+1} = \mathbf{P}\mathbf{\Sigma}^*\mathbf{Q}^\top$ with Eq. (2.12);
12: $\tilde{\mathbf{V}}^{t+1} = \Pi_\pi(\hat{\mathbf{V}}^t)$ as Eq. (2.13)
13: **if** $\xi_\alpha(f(\tilde{\mathbf{W}}^{t+1})) \le \xi_{\eta,\mathbf{W}^t}(\tilde{\mathbf{W}}^{t+1})$ **then**
14: $\eta_{t+1} = \eta$, break;
15: **end if**
16: **end for**
17: **end for**

Proof. Please refer to Theorem 3 in [28]. □

Such nuclear norm factorization is well established in recent work [29]. Replacing $r(\mathbf{U})$ with BF model in the objective (2.10), we solve $\mathbf{P}$ and $\mathbf{Q}$ with the APGL scheme,

$$\min_{\mathbf{P},\mathbf{Q}} \frac{1}{2}\left(\|\mathbf{P} - \hat{\mathbf{P}}^t\|_F^2 + \|\mathbf{Q} - \hat{\mathbf{Q}}^t\|_F^2\right) + \frac{\rho}{2}\left(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2\right), \tag{2.15}$$
$$\text{s.t.} \quad \mathbf{X}^t = \mathbf{P}^t\mathbf{Q}^{t\top} + \mathbf{V}^t,$$

where $\hat{\mathbf{P}}^t = \mathbf{P}^t - \eta\nabla_{\mathbf{P}^t}\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t$, $\hat{\mathbf{Q}}^t = \mathbf{Q}^t - \eta\nabla_{\mathbf{Q}^t}\xi_\alpha(\mathbf{X}^t) \circ \mathbf{\Psi}^t$, and $\rho = \eta\lambda_1$. L_2-norm regularization is used to avoid overfitting when d is larger than the intrinsic rank. We refer to the refined efficient algorithm as CSRR-e for short, and summarize it in Alg. 2. Since the objective (2.15) is biconvex, i.e. fixing $\mathbf{P}$ the problem is convex on $\mathbf{Q}$, and vice-versa, we solve Eq. (2.15) via updating $\mathbf{P}$ and $\mathbf{Q}$ iteratively until they are converged (Step 12-15 of Alg. 2). Note that CSRR-e uses $\xi_\alpha^I(x)$ as the loss function (Step 8 of Alg. 2), since it achieves better empirical results.

2.5 Theoretical Analysis We theoretically analyze the performance of the CSRR algorithms in terms of the cost-sensitive metrics. We assume that $\mathbf{X} := \{\mathbf{X} \in \mathbb{R}^{n\times m} | \|\mathbf{X}\|_* \le \epsilon, \forall(i,j)\ 0 \le X_{ij} \le 1\}$. The expected error can be formulated as $\mathbb{E}[R_h(\mathbf{X})] = \mathbb{E}[\frac{1}{mn}\sum_{i,j} h(\xi_\alpha(X_{ij}))]$, and the empirical error as $\hat{R}_h(\mathbf{X}) = \frac{1}{mn}\sum_{i,j} h(\xi_\alpha(X_{ij}))$. Inspired by the work in [7], we provide the error bound for the CSRR algorithms in the following lemma.

LEMMA 2.4. *Assume that $\mathbf{X} \in \mathcal{X}$, then with probability at*

Algorithm 2 CSRR-e

Input: $\mathbf{A} \in \mathbb{R}^{n\times m}$, scalars T, α, λ_1 and λ_2.
Output: $\mathbf{X}^T$, $\mathbf{V}^T$, $\mathbf{P}^T$ and $\mathbf{Q}^T$;
Initialize: $\tilde{\mathbf{P}}^1 = \tilde{\mathbf{P}}^0, \tilde{\mathbf{Q}}^1 = \tilde{\mathbf{Q}}^0, \tilde{\mathbf{V}}^1 = \tilde{\mathbf{V}}^0, C_0 = 0, \eta_1$;
for $t = 1, \ldots, T$ **do**
  Set $C_t = (1 + \sqrt{1 + 4C_{t-1}^2})/2$;
  Update: $\mathbf{X}^t = \mathbf{P}^t\mathbf{Q}^{t\top} + \mathbf{V}^t$ as in Eq. (2.14);
  Update: $\mathbf{\Psi}^t$ with Eq. (2.7);
  Compute $\xi_\alpha^I(\mathbf{X}^t)$ and $\nabla\xi_\alpha^I(\mathbf{X}^t)$;
  for $k = 0, 1$ to $\ldots$ **do**
    $\eta = 2^{-k}\eta_t$;
    $\mathbf{P}_{(0)}^t = \mathbf{P}^t$ and $\mathbf{Q}_{(0)}^t = \mathbf{Q}^t$;
    repeat
      $k = k + 1$;
      Update $\mathbf{P}_{(k)}^t$ and $\mathbf{Q}_{(k)}^t$ via solving Eq. (2.15);
    until convergence
    $\tilde{\mathbf{U}}^{t+1} = \mathbf{P}_{(k)}^t\mathbf{Q}_{(k)}^{t\top}$;
    $\tilde{\mathbf{V}}^{t+1} = \Pi_\pi(\hat{\mathbf{V}}^t)$ as in Eq. (2.13);
    if $\xi_\alpha(f(\tilde{\mathbf{W}}^{t+1})) \le \xi_{\eta,\mathbf{W}^t}(\tilde{\mathbf{W}}^{t+1})$ **then**
      $\eta_{t+1} = \eta$, $\tilde{\mathbf{P}}^{t+1} = \mathbf{P}_{(k)}^t$, $\tilde{\mathbf{Q}}^{t+1} = \mathbf{Q}_{(k)}^t$, break;
    end if
  end for
end for

least $1 - \delta$,

$$\begin{aligned}&\mathbb{E}[R_h(\mathbf{X})] - \min_{\mathbf{X}\in\mathcal{X}} \hat{R}_h(\mathbf{X})\\ \le& C\epsilon\alpha\frac{\sqrt{n} + \sqrt{m} + \sqrt[4]{S}}{mn} + \alpha\varepsilon\frac{\sqrt{\log(2/\delta)}}{\sqrt{mn}},\end{aligned} \tag{2.16}$$

where C is a constant and S is the total number of 1's in $\mathbf{A}$.

Proof. The proof is in Supplementary Material. □

Now we evaluate CSRR on the thresholded 0-1 matrix. With an appropriate value α, the following theorems provide a bound in terms of *weighted sum* (in (2.2)) and *weighted cost* (in (2.3)), respectively.

THEOREM 3. *By setting $\alpha = \frac{\mu_p T_n}{\mu_n T_p}$, with probability at least $1 - \delta$, the sequence $\{\mathbf{X}^t\}$ generated by CSRR is bounded,*

$$\begin{aligned}&\mathbb{E}[sum(\mathbf{X}^t)] \ge 1 - \min_{\mathbf{X}\in\mathcal{X}} \frac{\mu_n}{T_n}\mathcal{L}_\alpha(\mathbf{X})\\ &- \gamma\alpha\frac{\mu_n}{T_n}\{C\epsilon\frac{\sqrt{n} + \sqrt{m} + \sqrt[4]{S}}{mn} + \varepsilon\frac{\sqrt{\log(2/\delta)}}{\sqrt{mn}}\},\end{aligned}$$

where C is a constant and $\min(\frac{1}{q^2}, \frac{1}{(1-q)^2}) \le \gamma \le \max(\frac{1}{q^2}, \frac{1}{(1-q)^2})$, *and* $\mathcal{L}_\alpha(\mathbf{X}) = \sum_{i,j}(\alpha I_{(X_{ij}<q)}I_{(A_{ij}=1)} + I_{(X_{ij}\ge q)}I_{(A_{ij}=0)})$.

Proof. The proof is in Supplementary Material. □

THEOREM 4. *Under the same assumptions in Theorem 3, by setting $\alpha = \frac{C_p}{C_n}$, with probability at least $1 - \delta$, the sequence $\{\mathbf{X}^t\}$ generated by CSRR is bounded,*

$$\begin{aligned}&\mathbb{E}[cost(\mathbf{X}^t)] - \min_{\mathbf{X}\in\mathcal{X}} cost(\mathbf{X})\\ \le& c_n\gamma\alpha\{C\epsilon\frac{\sqrt{n} + \sqrt{m} + \sqrt[4]{S}}{mn} + \varepsilon\frac{\sqrt{\log(2/\delta)}}{\sqrt{mn}}\}.\end{aligned}$$

Proof. The proof is in Supplementary Material. □

Remark: [14] studied 1-bit matrix completion with a *unbiased estimator* of the squared loss $\ell(x, a) = (x-a)^2$, whereas we study the *biased matrix completion* problem. Due to the biased loss $\xi_\alpha(x)$, the expected error bound over the 0-1 matrix is the order of $O(\alpha/\sqrt{nm})$, which is tighter than $O(1/\sqrt{n})$ inferred by [14].

3 Experimental Results

We empirically evaluate the performance of CSRR on three real-world datasets. We start with experimental data and benchmark setup, followed by the experimental results.

3.1 Experimental Settings

3.1.1 Datasets We conducted experiments on three benchmark datasets: MovieLens-100K [30], MovieLens-1M, and EachMovie[1]. The U-I matrix of MovieLens-100K contains 943 users and 1,682 movies with 100,000 ratings. The U-I matrix of MovieLens-1M contains 6,039 users and 3,628 items with 1,000,209 ratings. EachMovie includes 61,265 users and 1,623 movies with 2,811,983 ratings. The densities of the U-I matrices from the three datasets are 6.3×10^{-2}, 4.47×10^{-2}, and 1.91×10^{-2}, respectively. Due to a small number of observed feedbacks, these datasets are suitable for the cost-sensitive learning.

3.1.2 Evaluation Metrics Given the observed matrix $\mathbf{A}$, we followed the setting of implicit feedback [31, 2] by treating the large ratings $A_{ij} \ge 3$ as observed feedbacks and evaluating the performance with two widely-used recommendation metrics, NDCG@N and F1@N: 1) NDCG@N is the normalized discounted cumulative gain, which reflects the usefulness in the ranking list. 2) F1-score@N is the weighted harmonic mean of precision and recall. We set $N = \{5, 10, 15\}$, since users usually focus on a few top-N ranked items in recommender systems. Basically, the higher these measures, the better the performance. Each observed matrix $\mathbf{A}$ was randomly divided into two non-overlapping sets for training and testing. For each user $\mathbf{a}_i$, 20% of its observed feedback $\mathbf{a}_i^+$ were randomly picked as testing data and the remaining 80% of the observed feedback were used

[1] https://grouplens.org/datasets/eachmovie/

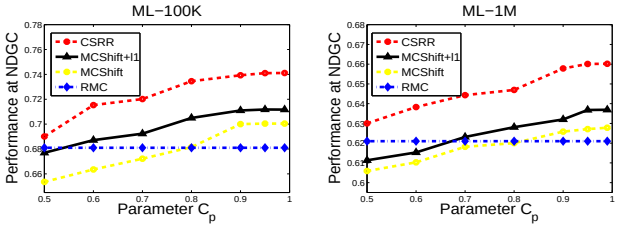
Table 1: Mean(%) and Standard Deviation(%) of Evaluation Metrics on Three Benchmark Datasets

Algorithm	ML-100K					
	F1@5	F1@10	F1@15	NDCG@5	NDCG@10	NDCG@15
PopRank	9.18±0.31	13.55±0.29	15.24±0.26	39.35±0.43	43.87±0.39	45.07±0.31
BPRMF	19.53±0.46	26.06±0.44	27.93±0.34	62.97±0.88	64.63±0.72	64.33±0.62
SPMF	18.74±0.53	25.13±0.40	27.72±0.26	68.02±0.51	68.33±0.72	68.17±0.46
RMF-MM	18.80±0.77	25.18±0.68	27.88±0.42	68.13±0.73	68.41±0.66	68.39±0.56
RMC	19.11±0.63	25.50±0.55	28.82±0.34	68.34±0.69	68.80±0.66	68.77±0.38
MCShift	19.12±0.54	25.56±0.43	29.00±0.29	70.16±0.54	69.54±0.53	69.33±0.49
MCShift+ℓ_1	19.41±0.49	25.94±0.33	29.32±0.31	70.94±0.53	70.08±0.56	69.61±0.46
CSRR-I	**22.33**±0.43	**28.81**±0.21	**31.66**±0.21	**74.32**±0.51	**74.50**±0.32	73.56±0.40
CSRR-II	21.92±0.33	28.37±0.31	31.51±0.20	73.95±0.43	74.02±0.41	**73.67**±0.21
CSRR-e	22.02±0.53	28.43±0.44	31.55±0.51	73.83±0.64	74.00±0.46	73.41±0.31
Algorithm	ML-1M					
PopRank	6.88±0.36	9.95±0.33	11.89±0.33	37.63±0.46	40.84±0.42	42.41±0.37
BPRMF	15.01±0.48	20.90±0.46	23.64±0.35	60.44±1.01	62.21±0.71	62.47±0.62
SPMF	16.63±0.46	21.44±0.42	23.42±0.44	62.03±0.62	63.61±0.40	64.05±0.46
RMF-MM	16.71±0.51	21.56±0.47	23.81±0.41	62.19±0.78	63.87±0.80	64.09±0.72
RMC	16.87±0.41	21.35±0.42	23.89±0.39	62.26±0.56	63.90±0.53	64.11±0.41
MCShift	16.52±0.42	21.78±0.37	24.25±0.31	63.07±0.60	63.97±0.63	64.03±0.47
MCShift+ℓ_1	16.90±0.40	22.12±0.39	24.72±0.39	63.40±0.44	64.13±0.61	64.21±0.62
CSRR-I	**18.73**±0.44	24.61±0.36	**27.63**±0.31	**66.02**±0.43	**67.54**±0.33	**67.56**±0.31
CSRR-II	18.41±0.32	**24.67**±0.32	27.21±0.12	65.83±0.41	67.10±0.43	67.33±0.39
CSRR-e	18.43±0.45	24.53±0.41	27.23±0.35	**66.02**±0.41	67.17±0.37	67.36±0.41
Algorithm	Each-MV					
PopRank	17.65±0.36	18.60±0.31	18.61±0.34	39.49±0.42	42.41±0.41	43.71±0.37
BPRMF	30.91±0.46	31.10±0.31	28.00±0.20	60.06±1.02	61.14±0.84	60.92±0.66
SPMF	31.10±0.53	30.62±0.43	27.77±0.34	59.65±0.55	60.85±0.71	60.58±0.47
RMF-MM	31.14±0.46	30.72±0.50	27.59±0.36	59.78±0.86	60.73±0.67	60.73±0.75
RMC	31.28±0.45	31.08±0.31	27.63±0.33	59.78±0.80	60.81±0.75	60.61±0.66
MCShift	31.31±0.38	31.12±0.30	27.32±0.30	59.65±0.51	60.81±0.47	60.67±0.45
MCShift+ℓ_1	31.52±0.40	31.32±0.35	27.77±0.31	60.17±0.55	61.25±0.44	61.02±0.40
CSRR-I	**33.24**±0.41	**32.98**±0.34	30.21±0.35	**62.61**±0.34	**63.42**±0.43	**63.34**±0.38
CSRR-II	32.77±0.33	32.54±0.33	**30.23**±0.31	62.11±0.31	63.01±0.28	63.03±0.30
CSRR-e	33.00±0.49	32.84±0.44	30.11±0.35	62.23±0.35	63.18±0.33	63.21±0.41

for training. For each evaluation metric, we first computed the performance for each user on the testing data, then reported the averaged results across all the users. For fair comparison, we repeated the random partition for 5 times and reported the average performance.

3.1.3 Baselines We compared the proposed algorithms with seven state-of-the-art baselines: 1) PopRank naively provides a recommendation based on the popularity of the items [32]. 2) BPRMF is an implicit MF technique that employs a uniform prior to missing entries [5]. 3) SPMF relies on a self-paced learning to build a robust MF model [8]. 4) RMF-MM proposes an L_1-norm based low-rank MF model and utilizes majorization minimization to solve the problem [33]. 5) RMC exploits a robust MD model where the nuclear norm is used for low-rank and L_1-norm for noise, and applies ADMM technique to minimize the objective [13]. 6) MCShift provides a biased loss based model to solve the matrix completion problem [7]. 7) MCShift+l_1 employs a robust MD framework for MCShift, where L_1-norm is introduced for outlier detection. Cross-validation is used to tune the parameters for all the baselines. For MF methods, the number of latent factors was tuned from $\{10, 15, \ldots, 50\}$. For CSRR, λ_1 and λ_2 were varied from $\{10^{-5}, \ldots, 10^2\}$ while $\alpha = \frac{C_p}{C_n}$ was varied by tuning C_p from $\{0.5, 0.55, \ldots, 0.95\}$ on the training set.

Figure 3: Parameter sensitivity analysis of α



3.2 Comparison Results Table 1 summarizes the comparison results in terms of NDCG and F1. It shows that the CSRR algorithms outperform all baselines over all metrics over the three datasets. 1) We observe that CSRR sig-

Figure 4: Parameter sensitivity analysis of η

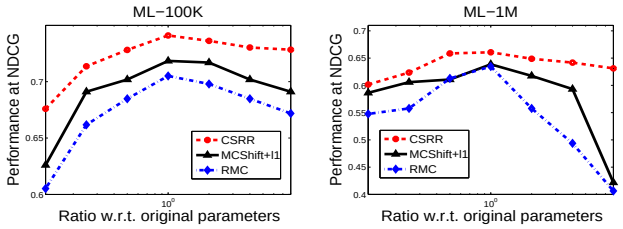
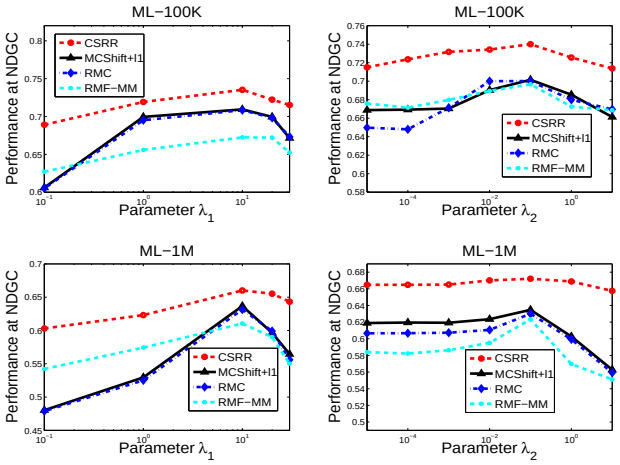


Figure 5: Parameter sensitivity analysis of λ_1 and λ_2



nificantly outperforms all robust MD methods. In terms of the averaged NDCG, CSRR outperforms RMF-MM, RMC and SPMF by relatively 7.2% on ML-100K, 5.4% on ML-1M and 4.1% on Each-MV. These baselines use a uniform weight restriction on missing data, which will cause the unsatisfactory performance when the observations are sparse. This validates the effectiveness of the proposed cost-sensitive metrics. 2) CSRR consistently outperforms MCShift in terms of both measures. The paired t-test verifies that all improvements are statistically significant with $p < 0.01$. It is clear that when there is no outlier detection, the performance of MCShift becomes poor — even lower than robust MD models. 3) With outlier detection, the performance of MCShift+ℓ_1 is improved over all datasets. However, it cannot eliminate the outliers in the training process. Our robust asymmetric learning model solves this issue and always achieves the best performance among all baselines. This highlights the importance of incorporating cost-sensitive learning and capped unilateral loss into the model, especially for the noisy labels and sparse observations in the U-I matrix.

3.3 Analysis for Cost-Sensitive Bias We compared the biased estimator based MD methods (including RMC) under various values of α. The results in Fig. 3 show the same trend that better prediction is obtained by increasing the weight of α, while a small weight of α will adversely degrade the performance. The reason is that some negative instances might be similar to the positives, updating these examples will harm the predictive power on positives. Thus, a high value of α can reduce the negative impact of such examples via more "aggressive" (ξ^I) or more "frequent" (ξ^{II}) update of the positive data.

3.4 Analysis for Learning Rate We also analyzed the parameter sensitivity of the learning rate η. In particular, we set the learning rate as a factor in $\{2^{-2}, 2^{-1}, \ldots, 2^2\}$ times the original learning rate used in the comparison results above, and reported the performance under various values of the learning rate. The results in Fig. 4 show that our algorithm performs more stably than other baselines on a broad value range of the learning rate.

3.4.1 Analysis for Robust Matrix Decomposition We studied the impact of the parameter pair (λ_1, λ_2). In particular, by fixing one parameter, we evaluated all compared methods by tuning the weights of the other one. The results in Fig. 5 show the same trend that better prediction is achieved by balancing the impact of outliers and low-rank basis, while either a large or a small value of (λ_1, λ_2) will adversely degrade the performance. The experimental results also show that our method is less sensitive to parameter tuning then other methods. Thus, our robust cost-sensitive learning model is more suitable for practical applications.

4 Conclusion

We proposed a novel recommendation model to overcome outlier and imbalance issues. By leveraging cost-sensitive learning and capped unilateral loss function, the proposed model seamlessly integrated them into a joint formulation, which solves both issues simultaneously. We incorporated this joint formulation into matrix decomposition for learning a low-rank representation. Particularly, we proposed a novel *logdet* function to refine the nuclear norm via imbalanced penalization of singular values. Moreover, our algorithm can handle large-scaled datasets after a relaxation. Theoretically, due to the cost-sensitive loss, CSRR achieves a lower error bound than the existing matrix completion method. We further validated the efficacy of our algorithm on benchmark datasets.

Acknowledgement

This work was supported by the Kind Abdullah Unviersity of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Awards No. URF/1/1976-04 and URF/1/3007-01.

References

[1] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*. Ieee, 2008, pp. 263–272.

[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW-2017*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[3] R. Devooght, N. Kourtellis, and A. Mantrach, "Dynamic matrix factorization with priors on unknown values," in *KDD*. ACM, 2015, pp. 189–198.

[4] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*. ACM, 2016, pp. 549–558.

[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[6] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008, pp. 502–511.

[7] C.-J. Hsieh, N. Natarajan, and I. S. Dhillon, "Pu learning for matrix completion," in *ICML*, 2015, pp. 2445–2453.

[8] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization." in *AAAI*, 2015, pp. 3196–3202.

[9] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4324–4337, 2013.

[10] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Robust matrix completion and corrupted columns," in *ICML-11*, 2011, pp. 873–880.

[11] C. Scott, "Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs," in *ICML-11*, 2011, pp. 153–160.

[12] P. Zhao, S. C. Hoi, R. Jin, and T. Yang, "Online auc maximization," in *ICML*, 2011, pp. 377–384.

[13] F. Shang, Y. Liu, J. Cheng, and H. Cheng, "Robust principal component analysis with missing data," in *CIKM*. ACM, 2014, pp. 1149–1158.

[14] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, "1-bit matrix completion," *Information and Inference*, vol. 3, no. 3, pp. 189–223, 2014.

[15] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l1-regularized loss minimization," *JMLR*, vol. 12, pp. 1865–1892, 2011.

[16] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 325–334.

[17] C. Elkan, "The foundations of cost-sensitive learning," in *IJCAI*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[18] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *JMLR*, vol. 11, pp. 1081–1107, 2010.

[19] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1-2, pp. 237–260, 1998.

[20] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *ACC-03*, vol. 3. IEEE, 2003, pp. 2156–2162.

[21] Z. Kang, C. Peng, and Q. Cheng, "Top-n recommender system via matrix completion." in *AAAI*, 2016, pp. 179–185.

[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *JACM*, vol. 58, no. 3, p. 11, 2011.

[23] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.

[24] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[25] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[27] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung, "A probabilistic approach to robust matrix factorization," in *ECCV*. Springer, 2012, pp. 126–139.

[28] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[29] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[30] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *TIIS*, vol. 5, no. 4, p. 19, 2016.

[31] W. Pan and L. Chen, "Gbpr: group preference based bayesian personalized ranking for one-class collaborative filtering," in *IJCAI*. AAAI Press, 2013, pp. 2691–2697.

[32] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[33] Z. Lin, C. Xu, and H. Zha, "Robust matrix factorization by majorization minimization," *PAMI*, 2017.