Low Rank Matrix Recovery via Robust Outlier Estimation

Xiaojie Guo, *Member, IEEE,* and Zhouchen Lin, *Fellow, IEEE*

***Abstract*—In practice, high-dimensional data are typically sampled from low-dimensional subspaces, but with intrusion of outliers and/or noises. Recovering the underlying structure and the pollution from the observations is of utmost importance to understanding the data. Besides properly modeling the subspace structure, how to handle the pollution is a core question regarding the recovery quality, the main origins of which include small dense noises and gross sparse outliers. Compared with the small noises, the outliers more likely ruin the recovery, as their arbitrary magnitudes can dominate the fidelity, and thus lead to misleading/erroneous results. Concerning the above, this paper concentrates on robust outlier estimate for low rank matrix recovery, termed as ROUTE. The principle is to classify each entry as an outlier or an inlier (with confidence). We formulate the outlier screening and the recovery into a unified framework. To seek the optimal solution to the problem, we first introduce a block coordinate descent based optimizer (ROUTE-BCD), then customize an alternating direction method of multipliers based one (ROUTE-ADMM). Through analyzing theoretical properties and practical behaviors, ROUTE-ADMM shows its superiority over ROUTE-BCD in terms of computational complexity, initialization insensitivity and recovery accuracy. Extensive experiments on both synthetic and real data are conducted to show the efficacy of our strategy and reveal its significant improvement over other state-of-the-art alternatives. Our code is publicly available at https://sites.google.com/view/xjguo/route.**

***Index Terms*—Outlier estimation, low-dimensional structure recovery, low rank matrix recovery, principal component pursuit**

I. Introduction

LOW rank matrix recovery is a process of discovering the underlying structure from given measurements, the inspiration and motivation of which are both that, in real cases, even very high-dimensional observations should be from a low-dimensional subspace but unfortunately with interference of outliers and/or noises. As a theoretic foundation in computer vision, pattern recognition and machine learning, the effectiveness of *low rank matrix recovery* (LRMR) has been witnessed by numerous fundamental tasks, such as principal component analysis [1], [2], collaborative filtering [3], [4] and subspace clustering [5], [6], as well as a wide spectrum of applications, like image denoising [7], reflection separation [8], rigid [9] and nonrigid [10] structure from motion, photometric stereo [11], [12], anomaly detection [13], [14] and super-resolution [15], to name just a few.

Suppose we are given an observation matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$, and know that it can be posed as a superposition of a low rank component $\mathbf{L} \in \mathbb{R}^{m\times n}$ and a residue component $\mathbf{E} \in \mathbb{R}^{m\times n}$. In this context, the LRMR problem can be directly or indirectly written in the following shape:

$$\min_{\mathbf{L},\mathbf{E}} \ \mathrm{rank}(\mathbf{L}) + \alpha\Psi(\mathbf{E}) \ \mathrm{s.t.} \ \mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{L}+\mathbf{E}), \quad (1)$$

where the function $\Psi(\cdot)$ is the penalty on the residual between the observed and recovered signals, $\mathrm{rank}(\cdot)$ stands for the low-rank constraint, and α is a non-negative parameter balancing the recovery fidelity and the low-rank promoting regularizer. Furthermore, $\mathcal{P}_\Omega(\cdot)$ is the orthogonal projection operator on the support $\Omega \in \{0,1\}^{m\times n}$. The binary-valued Ω_{ij} being 0 indicates that the corresponding entry is missing, and 1 otherwise. From Eq. (1), we can find that the quality of recovery depends on both the models of $\mathrm{rank}(\mathbf{L})$ and $\Psi(\mathbf{E})$.

As one of the two pivotal factors in LRMR, a proper low-rank promoting constraint on $\mathbf{L}$ is required to advocate the expected structure. It is computationally intractable (NP-hard) to directly minimize the rank function, say $\mathrm{rank}(\mathbf{L})$, due to its discreteness. A widely used scheme is employing its tightest convex proxy, *i.e.* the nuclear norm $\|\mathbf{L}\|_*$ (the sum of all the singular values) [16], [2], [17]. *Nuclear norm minimization* (NNM) based approaches can perform stably without knowing the target rank of recovery in advance. But, their applicability is often limited by the necessity of executing expensive *singular value decomposition* (SVD) for multiple times. At less expense, *bilinear factorization* (BF) [18], [19], [20], [21], [22] is an alternative by replacing $\mathbf{L}$ with $\mathbf{UV}$, where the product of $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times n}$ implicitly guarantees that the rank of $\mathbf{UV}$ is never over r, typically $r \ll \min(m,n)$. This factorization strategy, via getting rid of SVDs, can greatly relieve the pressure of computation and provide accurate results when the target rank is given. However, in some tasks, the target rank is unknown beforehand. In such a situation, the performance of BF would sharply degrade because of its sensitivity to the guess of target rank, especially when the data are severely contaminated. For connecting NNM and BF, and inheriting their respective merits, some bridges are recently built [23], [24]. One representative is adding $\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2$ into the objective of the BF [25]. Although the techniques above have made great progresses, the tolerance to dirty data is expected to be further improved.

In practical scenarios, acquiring perfect data is never the case. Furthermore, "*a little gall spoils a great deal of honey*" is

X. Guo (xj.max.guo@gmail.com) is with School of Computer Software, Tianjin University, Tianjin 300350, China. X. Guo is supported by National Natural Science Foundation (NSF) of China (grant no. 61772512).

Z. Lin (zlin@pku.edu.cn) is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, China, and Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China. Z. Lin is supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), National Natural Science Foundation (NSF) of China (grant nos. 61625301 and 61731018), and Qualcomm.

quite a common issue. This indicates that, without an effective strategy to reduce the negative effect from outliers and/or noises, the low rank matrix recovery is very likely prevented from reasonable solutions. Hence, besides properly modeling the low-rank structure, how to handle the pollution, especially gross outliers, is core to the performance of recovery. Arguably, the square loss (*a.k.a.* ℓ^2 loss) is the most commonly used penalty, which is optimal to Gaussian noises, like in *principal component analysis* (PCA) [1]. But, the square loss is brittle to outliers that are not unusual to find in real data. To be robust against gross corruptions, the ℓ^1 loss as the tightest convex surrogate of the ℓ_0 one becomes popular, *e.g.* in *robust PCA* (RPCA) [2]. In parallel, there are a few works carried out from probabilistic standpoints. *Probabilistic matrix factorization* (PMF) [21] and *probabilistic robust matrix factorization* (PRMF) [24] are two representatives, the residual penalties of which are equivalent to the ℓ^2 (PCA) and ℓ^1 (*principal component pursuit*, PCP) losses, respectively. For better fitting the residual, combining the ℓ^1 loss for the outlier component and the ℓ^2 for the small noise one is considered in [26]. Zhao *et al.* [27] proposed two loss functions to promote robustness against outliers. One is derived from Student's t distribution, while the other one is a smoothed ℓ^1 loss. Although the ℓ^1 loss can perform better than the ℓ^2 in dealing with the outliers, it still suffers from the scale issue. Moreover, the outliers may have a physical meaning in a specific task, *e.g.* foreground objects in surveillance videos. In this situation, the residuals cannot act as the "outliers" per se. Among others, the ideal option to model outliers is the ℓ^0 loss, due to its scale invariance. The non-convexity and discontinuity of the ℓ^0 penalty make it not so preferred by the community, although many works have proven its improvement on different tasks, like [28] for image deblurring, [17] for foreground detection and [29] for video editing.

Aside from the above entry-wise pollution modelings, a number of sample-wise outlier detection models have been proposed over the past years. A popular scheme is to employ the group sparsity to identify outlier samples. This kind of methods is typically specified to the task of subspace segmentation such as [6][14]. The samples with large reconstruction errors are viewed as the outlier samples, which are then excluded from the reconstruction basis set. The mentioned methods are different from our task, *i.e.* LRMR, that needs to identify outliers entry-wise. Further, some deep learning works recognized the low-dimensional structure can improve the performance. They introduced low-rank layers/filters to regularize intermediate results/extract desired features. Specifically, PCANet adopts PCA to learn multistage filter banks [30], while LRRNet first extracts the low rank part from polluted input using an off-the-shell method, and then uses the extracted low rank components as filters [31]. These designs are finally applied to classification/recognition tasks, which do not require precise matrix recoveries. Similar ideas go to [32][33].

Back to the general formulation (1), if the support of both outliers and missing elements is given, the problem turns out to be a simpler version, *i.e.* the *low rank matrix completion* (LRMC). Comparing with LRMR, the difficulty of LRMC, because of the known support, significantly decreases, which corroborates the intuition and theoretical fact that knowing the corruption location is beneficial. Therefore, it is natural to ask that *if we can connect the LRMR to the LRMC via robustly estimating outliers,* since by doing so the LRMR will be conquered more easily.

Contribution To answer the above question, this paper proposes a Robust OUTlier Estimation method, called ROUTE. Concretely, the contributions can be summarized as follows:

1) We design a method to jointly estimate outliers and recover the low rank matrix, which connects the LRMR and LRMC by assigning the estimated outliers with small weights;
2) Compared with the hard binary support, our weighting scheme assigns real-valued weights $[0, 1]$, which can be viewed as classification with confidence/probability;
3) Our design employs a maximum entropy regularization term to minimize the prediction bias, which behaves like a sigmoid function;
4) To seek the optimal solution for ROUTE, we provide a *block coordinate descent* based optimizer and an *alternating direction method of multipliers* based one, together with analysis on their theoretical properties and practical behaviors;
5) Extensive experimental results on both synthetic and real data are provided to show the efficacy of our ROUTE and reveal its superiority over other state-of-the-arts.

A preliminary version of this manuscript appeared in [34]. Compared with [34], this journal version presents the model design and the solver in more theoretical details. More experiments are conducted to verify the advances of our ROUTE over other state-of-the-art alternatives on LRMR.

II. Methodology

A. Problem Statement and Formulation

In the simplest case, the support of observed elements is at hand, the intrinsic rank r is given, and the data are clean or just with slight noises. The optimal recovery can be obtained via conquering the following BF problem:

$$\min_{\mathbf{U},\mathbf{V}} \quad \|\Omega \odot (\mathbf{Y} - \mathbf{U}\mathbf{V})\|_F^2, \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times n}$, and $\odot$ is the Hadamard product operator. However, in many real-world cases, the intrinsic rank is not available. In such a situation, an option for recovering the low rank component (LRMC) is to optimize the following NNM problem:

$$\min_{\mathbf{L}} \quad \|\mathbf{L}\|_* + \frac{\alpha}{2}\|\Omega \odot (\mathbf{Y} - \mathbf{L})\|_F^2. \tag{3}$$

As mentioned, the nuclear norm minimization requires to execute expensive SVDs on the full size data. To mitigate the computational pressure, Theorem 1 builds a bridge between the NNM and BF models.

Theorem 1. *For any matrix* $\mathbf{L} \in \mathbb{R}^{m\times n}$*, the following relationship holds [35]:*

$$\|\mathbf{L}\|_* = \min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 \quad \text{s.t.} \quad \mathbf{L} = \mathbf{U}\mathbf{V}.$$

If $\mathrm{rank}(\mathbf{L}) = r \leq \min(m,n)$*, then the minimum solution above is attained at a factor decomposition* $\mathbf{L} = \mathbf{UV}$*, where* $\mathbf{U} \in \mathbb{R}^{m\times r}$ *and* $\mathbf{V} \in \mathbb{R}^{r\times n}$.

As a result, applying Theorem 1 on (3) reads:

$$\min_{\mathbf{U},\mathbf{V}} \ \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \frac{\alpha}{2}\|\Omega \odot (\mathbf{Y} - \mathbf{UV})\|_F^2. \quad (4)$$

Compared with directly minimizing $\|\Omega \odot (\mathbf{Y} - \mathbf{UV})\|_F^2$, the model (4) inherits the advantage of (3), which avoids overfitting when r is larger than the intrinsic rank.

From a Bayesian perspective, the model (4) corresponds to a *maximum a posteriori* (MAP) problem. Consider the following probabilistic models:

$$\begin{aligned} p(y_{ij}|[\mathbf{UV}]_{ij},\lambda) &\sim \mathcal{N}(y_{ij}|[\mathbf{UV}]_{ij},\lambda^{-1}) \quad \forall (i,j) \in \Omega; \\ p(u_{ik}|\tilde{\lambda}) &\sim \mathcal{N}(u_{ik}|0,\tilde{\lambda}^{-1}); \\ p(v_{kj}|\tilde{\lambda}) &\sim \mathcal{N}(v_{kj}|0,\tilde{\lambda}^{-1}), \end{aligned} \quad (5)$$

where $\mathcal{N}(x|\mu,\sigma^2)$ stands for the Gaussian distribution whose *probability density function* (PDF) is $\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$. Taking $\mathbf{U}$ and $\mathbf{V}$ as model parameters, as well as λ and $\tilde{\lambda}$ as hyper-parameters with fixed values, we can seek the optimal parameters, according to the Bayes' rule, through maximizing the posterior probability:

$$\begin{aligned} p(\mathbf{U},\mathbf{V}|\mathbf{Y},\lambda,\tilde{\lambda},\Omega) \propto \Bigg(& \underbrace{\prod_{(i,j)\in\Omega} \mathcal{N}(y_{ij}|[\mathbf{UV}]_{ij},\lambda^{-1})}_{\text{likelihood}} \\ & \underbrace{\prod_{i,k} \mathcal{N}(u_{ik}|0,\tilde{\lambda}^{-1})}_{\text{prior on } \mathbf{U}} \underbrace{\prod_{k,j} \mathcal{N}(v_{kj}|0,\tilde{\lambda}^{-1})}_{\text{prior on } \mathbf{V}} \Bigg). \end{aligned} \quad (6)$$

Maximizing the above posterior is equivalent to minimizing its negative log, that is

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} -\ln p(\mathbf{U},\mathbf{V}|\mathbf{Y},\lambda,\tilde{\lambda},\Omega) = \\ &\min_{\mathbf{U},\mathbf{V}} \frac{\tilde{\lambda}}{2}\|\mathbf{U}\|_F^2 + \frac{\tilde{\lambda}}{2}\|\mathbf{V}\|_F^2 + \frac{\lambda}{2}\|\Omega \odot (\mathbf{Y} - \mathbf{UV})\|_F^2, \end{aligned} \quad (7)$$

which is in the same form with (4) via setting α to $\lambda/\tilde{\lambda}$.

In real world tasks, unfortunately, the data are polluted by, besides small noises, gross corruptions, which may prevent the recovery from reasonable results. Hence, some steps should be taken for reducing the negative effect of such pollution. Let us simply distinguish that the elements are contaminated by either small noises or gross outliers according to the magnitudes of residual. To achieve the goal, an indicator is required to tell which elements are polluted by small noises ($\mathbf{W}$) and which by gross corruptions ($\overline{\mathbf{W}}$). As for the outlier entries, due to their arbitrary magnitudes, the ℓ^0 loss is ideal to host them. While for the other entries, the ℓ^2 loss can take care of. Based on the above, we have:

$$\begin{aligned} \min \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \frac{\alpha}{2}\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2 + \beta\|\overline{\mathbf{W}}\|_1 \\ \text{s.t.} \ \ \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}; \ \ \mathbf{W} \text{ and } \overline{\mathbf{W}} \in \{0,1\}^{m\times n}, \end{aligned} \quad (8)$$

where β is a weight to the corresponding term and $\mathbf{1}$ represents an all-one matrix with compatible size. We can see from Eq. (8) that, the support Ω is replaced by a weight matrix $\mathbf{W}$ containing both the given support and the estimated outlier support. Please note that, under the binary weighting, $\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2 = \sum_{i,j} w_{ij}[\mathbf{Y} - \mathbf{UV}]_{ij}^2$, and $\|\overline{\mathbf{W}}\|_1 = \|\overline{\mathbf{W}}\|_0$ for imposing the sparsity on the outliers.

Consequently, the likelihood in (6) should be modified accordingly as follows:

$$\prod_{(i,j)\in\Omega} \Big(w_{ij}\mathcal{N}(y_{ij}|[\mathbf{UV}]_{ij},\lambda^{-1}) + \bar{w}_{ij}(1/\epsilon) \Big), \quad (9)$$

where the outliers are assumed to follow a uniform distribution, due to its arbitrariness, with the PDF $\frac{1}{\epsilon}$. We note that, w_{ij} and $\bar{w}_{ij}$ are binary, satisfying $w_{ij} + \bar{w}_{ij} = 1$, which can be treated as the hard mixing coefficients of the mixture of a Gaussian and a uniform distributions. Adopting the likelihood (9) results in a "long tailed" distribution with the PDF $\max(\mathcal{N}(y_{ij}|[\mathbf{UV}]_{ij},\lambda^{-1}), 1/\epsilon)$. This is desired to better fit the residual. Minimizing the negative log of the posterior by replacing the likelihood in (6) with (9) shows the equivalence with (8) by setting β to $(\ln\frac{\epsilon}{\sqrt{2\pi\lambda^{-1}}})/\tilde{\lambda}$.

The hard weighting, for one thing, frequently leads the optimization to be stuck into bad local minima. For another thing, the pollution in data is often non-homogeneously distributed. To address the discreteness issue and reflect the importance of elements more faithfully, we relax the value range of $\mathbf{W}$ and $\overline{\mathbf{W}}$ from binary $\{0,1\}^{m\times n}$ into real-valued $[0,1]^{m\times n}$, and employ an entropy term. The definition of entropy is $-\sum_{c=1}^{k} p_c \log p_c$ with $\sum_{c=1}^{k} p_c = 1$. The principle of maximum entropy tells that, the probability distribution which best represents the current state of knowledge is the one with largest entropy subject to accurately stated prior data. In other words, it is able to minimize the prediction bias. Return to our problem, the weighting variable w_{ij} can be equally viewed as the probability of the corresponding entry being classified as an outlier. It is instructive to note that maximizing the entropy (concave) is equivalent to minimizing its negative (convex). Consequently, we have the final formulation of ROUTE-LRMR as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{W}} \ & \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2 \\ & + \beta\|\overline{\mathbf{W}}\|_1 + \gamma\sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log\bar{w}_{ij}) \\ & \text{s.t.} \ \ \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}; \ \ \mathbf{W} \text{ and } \overline{\mathbf{W}} \in [0,1]^{m\times n}, \end{aligned} \quad (10)$$

where γ is a non-negative coefficient controlling the importance of the corresponding term. Further, due to the relaxation, $\sqrt{\mathbf{W}}$ with entries $\sqrt{w_{ij}}$ is used to hold the equivalence: $\sum_{i,j} w_{ij}[\mathbf{Y} - \mathbf{UV}]_{ij}^2 = \|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2$. For (8), $\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2 = \|\mathbf{W} \odot (\mathbf{Y} - \mathbf{UV})\|_F^2$. As shown in (10), it has embraced all the aforementioned concerns for simultaneously pursuing outliers and recovering the low rank matrix. In the next two subsections, we will customize two algorithms for solving (10).

B. A BCD Optimizer

Intuitively, the *block coordinate descent* (BCD) strategy [36] seems to be a natural choice to conquer the problem (10), which iteratively finds the optimal solution to one of involved variables with other ones fixed until convergence. In the following, we provide the procedure step by step, including $\mathbf{V}$, $\mathbf{U}$ and $\mathbf{W}$-$\overline{\mathbf{W}}$ sub-problems.

$\mathbf{V}$ sub-problem – With $\mathbf{U}^{(t)}$, $\mathbf{W}^{(t)}$ and $\overline{\mathbf{W}}^{(t)}$ fixed as the estimation of the previous (t-th) iteration, the target problem turns out to be:

$$\min_{\mathbf{V}} \quad \|\mathbf{V}\|_F^2 + \alpha\|\sqrt{\mathbf{W}^{(t)}} \odot (\mathbf{Y} - \mathbf{U}^{(t)}\mathbf{V})\|_F^2, \tag{11}$$

which can be divided into a set of independent column-wise problems. We resolve them one by one as below:

$$\begin{aligned}\mathbf{v}_j^{(t+1)} &\leftarrow \min_{\mathbf{v}_j} \ \|\mathbf{v}_j\|_2^2 + \alpha\|\sqrt{\mathbf{w}_j^{(t)}} \odot (\mathbf{y}_j - \mathbf{U}^{(t)}\mathbf{v}_j)\|_2^2 \\ &= \alpha(\mathbf{I} + \alpha\mathbf{U}^{(t)T}\mathbf{\Lambda}_j\mathbf{U}^{(t)})^{-1}(\mathbf{U}^{(t)T}\mathbf{\Lambda}_j\mathbf{y}_j),\end{aligned} \tag{12}$$

where $\mathbf{I}$ means the identity matrix with proper size, and $\mathbf{\Lambda}_j \in \mathbb{R}^{m\times m}$ denotes the diagonal matrix composed by $\mathbf{w}_j^{(t)}$.

$\mathbf{U}$ sub-problem – Similarly, $\mathbf{U}$ can be updated via optimizing the following:

$$\min_{\mathbf{U}} \quad \|\mathbf{U}\|_F^2 + \alpha\|\sqrt{\mathbf{W}^{(t)T}} \odot (\mathbf{Y}^T - \mathbf{V}^{(t+1)T}\mathbf{U}^T)\|_F^2. \tag{13}$$

Again, the (13) can be decomposed into a group of row-wise problems with respect to $\mathbf{U}$. Each row $\mathbf{u}_{i\cdot}$ has a closed form solution like:

$$\mathbf{u}_{i\cdot}^{(t+1)} \leftarrow [\alpha(\mathbf{I} + \alpha\mathbf{V}^{(t+1)}\mathbf{\Gamma}_i\mathbf{V}^{(t+1)T})^{-1}(\mathbf{V}^{(t+1)}\mathbf{\Gamma}_i\mathbf{y}_{i\cdot}^T)]^T, \tag{14}$$

where $\mathbf{\Gamma}_i \equiv \mathrm{Diag}(w_{i1}^{(t)}, ..., w_{in}^{(t)})$.

$\mathbf{W}$-$\overline{\mathbf{W}}$ sub-problem – Picking out the terms relevant to $\mathbf{W}$ and $\overline{\mathbf{W}}$ results in the following optimization problem:

$$\begin{aligned}\min_{\mathbf{W},\overline{\mathbf{W}}} \quad & \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)})\|_F^2 + \beta\|\overline{\mathbf{W}}\|_1 \\ & + \gamma\sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log\bar{w}_{ij}) \\ \text{s.t.} \quad & \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}; \quad \mathbf{W} \text{ and } \overline{\mathbf{W}} \in [0,1]^{m\times n}.\end{aligned} \tag{15}$$

From the objective of (15), we find that the problem is also separable. Without any loss of generality, let us take the (i,j)-th element for example. Casting the problem into the Lagrange multiplier framework gives the following Lagrange function:

$$\begin{aligned}\mathcal{Q}(w_i, \overline{w}_i, \eta_i) \equiv \frac{\alpha}{2}w_{ij}[\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}]_{ij}^2 + \beta\overline{w}_{ij} + \\ \gamma(w_{ij}\log w_{ij} + \overline{w}_{ij}\log\overline{w}_{ij}) + \eta_i(w_{ij} + \overline{w}_{ij} - 1),\end{aligned} \tag{16}$$

where η_i is a Lagrange multiplier. Taking the derivative of $\mathcal{Q}(w_i, \overline{w}_i, \eta_i)$ to w_i, $\overline{w}_i$ and η_i respectively and setting them to zero lead to the following:

$$\begin{aligned}\partial\mathcal{Q}_{w_i} &= \frac{\alpha}{2}[\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}]_{ij}^2 + \gamma\log w_i + \eta_i + \gamma = 0; \\ \partial\mathcal{Q}_{\overline{w}_i} &= \beta + \gamma\log\overline{w}_i + \eta_i + \gamma = 0; \\ \partial\mathcal{Q}_{\eta_i} &= w_i + \overline{w}_i - 1 = 0.\end{aligned} \tag{17}$$

Algorithm 1: ROUTE-LRMR (BCD)

Input: Observation matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$; support $\Omega \in \{0,1\}^{m\times n}$; a guess/target rank r; non-negative parameters α, β and γ.
Init.: $\mathbf{W}^{(0)} \in \mathbb{R}^{m\times n} \leftarrow \Omega \odot \mathbf{1}$; $\mathbf{U}^{(0)} \in \mathbb{R}^{m\times r}$ and $\mathbf{V}^{(0)} \in \mathbb{R}^{r\times n}$ are all initialized randomly; $t \leftarrow 0$.
while *not converged* **do**
  for $j = 1 : n$ **do**
    Update $\mathbf{v}_j^{(t+1)}$ via (12);
  end
  for $i = 1 : m$ **do**
    Update $\mathbf{u}_{i\cdot}^{(t+1)}$ via (14);
  end
  for $\forall\ (i,j)\ \ \&\ \ \Omega_{ij}$ **do**
    Update $w_{ij}^{(t+1)}$ via (18);
  end
  $t \leftarrow t + 1$;
end
Output: Optimal $\mathbf{W}^*$, $\mathbf{U}^*$ and $\mathbf{V}^*$

The optimal solutions to w_i and $\overline{w}_i$ can be obtained by solving the equation system in (17) as:

$$\begin{aligned}w_{ij}^{(t+1)} &\leftarrow \frac{\exp(-\alpha[\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}]_{ij}^2/2\gamma)}{\exp(-\alpha[\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}]_{ij}^2/2\gamma) + \exp(-\beta/\gamma)} \\ &= \frac{1}{1 + \exp((\alpha[\mathbf{Y} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}]_{ij}^2/2 - \beta)/\gamma)},\end{aligned} \tag{18}$$

which is in a sigmoid form. And its complementary is $\overline{w}_{ij}^{(t+1)} \leftarrow 1 - w_{ij}^{(t+1)}$.

Remarks (a) When $w_{ij} \in \{0,1\}$ adopted and the entropy term disabled (hard weighting), the solution to Eq. (15) is: if $\frac{\alpha}{2}[\mathbf{Y} - \mathbf{UV}]_{ij}^2 < \beta$, then $w_{ij} \leftarrow 1$; if $\frac{\alpha}{2}[\mathbf{Y} - \mathbf{UV}]_{ij}^2 = \beta$, then w_{ij} could be either of $\{0,1\}$; otherwise $w_{ij} \leftarrow 0$. (b) When $w_{ij} \in [0,1]$ adopted and the entropy term disabled (relaxed version), the solution to Eq. (15) is: if $\frac{\alpha}{2}[\mathbf{Y} - \mathbf{UV}]_{ij}^2 < \beta$, then $w_{ij} \leftarrow 1$; if $\frac{\alpha}{2}[\mathbf{Y} - \mathbf{UV}]_{ij}^2 = \beta$, then w_{ij} could be any value in $[0,1]$; otherwise $w_{ij} \leftarrow 0$.

Algorithm 1 has summarized the proposed BCD optimizer. The procedure stops when $\|\mathbf{W}^{(t)} - \mathbf{W}^{(t-1)}\|_F^2 < \delta\|\mathbf{W}^{(0)}\|_F^2$ (δ adopts $1e{-}7$ in this paper) or the maximum iteration number is reached. The algorithm can theoretically ensure that the energy of the objective (10) will monotonically decrease as the iteration goes (Sec. III), but its performance is inferior to the optimizer proposed in the next sub-section, in computational complexity, initialization insensitivity and recovery accuracy.

C. An ADMM Optimizer

Alternatively, the *alternating direction method of multipliers* (ADMM) scheme [37] can be adopted to solve the problem (10). To apply ADMM on our problem, the objective is required to be separable. To this end, an auxiliary variable $\mathbf{L}$ is introduced to replace $\mathbf{UV}$ in the third term. Accordingly,

$\mathbf{L} = \mathbf{UV}$ acts as a constraint. Subsequently, the associated problem becomes:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{L},\mathbf{W}} \ & \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{L})\|_F^2 + \\ & \beta\|\overline{\mathbf{W}}\|_1 + \gamma \sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log \bar{w}_{ij}) \\ \text{s.t. } & \mathbf{L} = \mathbf{UV}, \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1};\ \mathbf{W} \text{ and } \overline{\mathbf{W}} \in [0,1]^{m\times n}.\end{aligned} \tag{19}$$

The augmented Lagrangian function of (19) is defined as:

$$\begin{aligned}&\mathcal{L}^{\mu}_{\{\mathbf{W}+\overline{\mathbf{W}}=\mathbf{1};\mathbf{W},\overline{\mathbf{W}}\in[0,1]^{m\times n}\}}(\mathbf{U},\mathbf{V},\mathbf{L},\mathbf{W},\mathbf{Z}) \equiv \\ &\frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{L})\|_F^2 + \\ &\beta\|\overline{\mathbf{W}}\|_1 + \gamma \sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log \bar{w}_{ij}) + \\ &\frac{\mu}{2}\|\mathbf{L} - \mathbf{UV}\|_F^2 + \langle \mathbf{Z}, \mathbf{L} - \mathbf{UV}\rangle,\end{aligned} \tag{20}$$

where $\langle\cdot,\cdot\rangle$ designates the inner product, μ is a positive penalty and $\mathbf{Z}$ is a Lagrangian multiplier. Notice that the constraints on $\mathbf{W}$ and $\overline{\mathbf{W}}$ are enforced as hard constraints. The solver updates the variables in an iterative manner.

For $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{L}$, their solutions in closed-form are calculated via equating the derivatives of (20) with respect to $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{L}$ to zero respectively with the other variables fixed:

$$\begin{aligned}&\mathbf{V}^{(t+1)} \leftarrow (\mathbf{I} + \mu^{(t)}\mathbf{U}^{(t)T}\mathbf{U}^{(t)})^{-1}\mathbf{U}^{(t)^T}(\mu^{(t)}\mathbf{L}^{(t)} + \mathbf{Z}^{(t)}); \\ &\mathbf{U}^{(t+1)} \leftarrow (\mu^{(t)}\mathbf{L}^{(t)} + \mathbf{Z}^{(t)})\mathbf{V}^{(t+1)T}\mathbf{K}^{(t+1)-1}; \\ &\mathbf{L}^{(t+1)} \leftarrow \frac{\alpha\mathbf{W}^{(t)} \odot \mathbf{Y} + \mu^{(t)}\mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)} - \mathbf{Z}^{(t)}}{\alpha\mathbf{W}^{(t)} + \mu^{(t)}\mathbf{1}},\end{aligned} \tag{21}$$

where $\mathbf{K}^{(t+1)}$ denotes $\mathbf{I} + \mu^{(t)}\mathbf{V}^{(t+1)}\mathbf{V}^{(t+1)T}$, and the division in updating $\mathbf{L}$ is element-wise.

As regards the $\mathbf{W}$-$\overline{\mathbf{W}}$ sub-problem, it is similar with that shown in (15), only replacing $\mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}$ by $\mathbf{L}^{(t+1)}$, as:

$$\begin{aligned}\min_{\mathbf{W},\overline{\mathbf{W}}} \ & \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{L}^{(t+1)})\|_F^2 + \beta\|\overline{\mathbf{W}}\|_1 \\ & + \gamma \sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log \bar{w}_{ij}) \\ \text{s.t. } & \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1};\ \ \mathbf{W} \text{ and } \overline{\mathbf{W}} \in [0,1]^{m\times n}.\end{aligned} \tag{22}$$

Accordingly, the closed form solution is

$$w_{ij}^{(t+1)} \leftarrow \frac{1}{1 + \exp((\alpha[\mathbf{Y} - \mathbf{L}^{(t+1)}]_{ij}^2/2 - \beta)/\gamma)}. \tag{23}$$

Besides, the Lagrange multiplier $\mathbf{Z}$ and μ are updated via:

$$\begin{aligned}&\mathbf{Z}^{(t+1)} \leftarrow \mathbf{Z}^{(t)} + \mu^{(t)}(\mathbf{L}^{(t+1)} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)}); \\ &\mu^{(t+1)} \leftarrow \mu^{(t)}\rho, \quad \rho > 1.\end{aligned} \tag{24}$$

The parameter μ is monotonically increased by ρ during iterations, gradually leading the solution to the feasible region.

For clarity and completeness, the customized ADMM solver to the problem (10) is outlined in Algorithm 2. The procedure should not be terminated until the equality constraint $\mathbf{L} = \mathbf{UV}$ is satisfied up to a given tolerance, that is $\|\mathbf{L} - \mathbf{UV}\|_F \leq \varsigma\|\mathbf{Y}\|_F$, or the maximal number of iterations is reached. In all our experiments, the tolerance factor ς is chosen as $1e{-}7$. Please refer to the complete Algorithm 2 for other details. We will compare the proposed BCD and ADMM optimizers both theoretically and experimentally in Section III and V.

Algorithm 2: ROUTE-LRMR (ADMM)

Input: Observation matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$; support $\Omega \in \{0,1\}^{m\times n}$; a guess/target rank r; non-negative parameters α, β and γ.
Init.: $\mu^{(0)} \leftarrow 1$ and $\rho \leftarrow 1.1$; $\mathbf{W}^{(0)} \in \mathbb{R}^{m\times n} \leftarrow \Omega \odot \mathbf{1}$; $\mathbf{L}^{(0)} \in \mathbb{R}^{m\times n}$, $\mathbf{U}^{(0)} \in \mathbb{R}^{m\times r}$ and $\mathbf{V}^{(0)} \in \mathbb{R}^{r\times n}$ are all initialized randomly; $\mathbf{Z}^{(0)} \in \mathbb{R}^{m\times n} \leftarrow \mathbf{0}$; $t \leftarrow 0$.
while *not converged* **do**
  Update $\mathbf{V}^{(t+1)}$, $\mathbf{U}_{(t+1)}$ and $\mathbf{L}^{(t+1)}$ via (21);
  for $\forall\ (i,j)\ \ \&\ \ \Omega_{ij}$ **do**
    Update $w_{ij}^{(t+1)}$ via (23);
  end
  Update $\mathbf{Z}^{(t+1)}$ and $\mu^{(t+1)}$ via (24);
  $t \leftarrow t + 1$
end
Output: Optimal $\mathbf{W}^*$ and $\mathbf{L}^*$

III. Theoretical Analysis

We first provide some useful theoretical results, including Lemma 1 and Proposition 1, for the $\mathbf{W}$-$\overline{\mathbf{W}}$ sub-problem.

Lemma 1. *At stage t with $\mathbf{U}^{(t)}\mathbf{V}^{(t)}$ fixed for the BCD optimizer or with $\mathbf{L}^{(t)}$ fixed for the ADMM optimizer, the solutions, i.e. w_{ij} given in Eqs.* (18) *and* (23)*, are global optimal to the corresponding intermediary problems, respectively.*

Proof. Taking the problem (15) for example, having $\mathbf{U}^{(t)}$ and $\mathbf{V}^{(t)}$ (thus $\mathbf{U}^{(t)}\mathbf{V}^{(t)}$) fixed, the objective function in (15) is convex with respect to $w_{ij} \in [0,1]$. The solution in Eq. (18) is computed by the Lagrange multiplier method, which guarantees that the obtained solution is feasible and satisfies the KKT conditions for (15). For the ADMM optimizer, the conclusion can be reached analogously. □

Proposition 1. *The function defined in Eq.* (15)*, containing three parameters including $\tilde{\beta} \equiv \beta/\alpha$, $\tilde{\gamma} \equiv \gamma/\alpha$, and $\varepsilon_{ij} \equiv [\mathbf{Y} - \mathbf{UV}]_{ij}^2/2$ for the BCD optimizer or $\varepsilon_{ij} \equiv [\mathbf{Y} - \mathbf{L}]_{ij}^2/2$ for the ADMM optimizer, has the following properties:*

1) *$w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij})$ is monotonically decreasing with respect to ε_{ij}, which holds $\lim_{\varepsilon_{ij}\to 0} w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij}) = \frac{1}{1+\exp(-\tilde{\beta}/\tilde{\gamma})}$ and $\lim_{\varepsilon_{ij}\to+\infty} w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij}) = 0$;*
2) *$w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij})$ is monotonically increasing with respect to $\tilde{\beta}$, which holds that $\lim_{\tilde{\beta}\to 0} w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij}) = \frac{1}{1+\exp(\varepsilon_{ij}/\tilde{\gamma})}$ and $\lim_{\tilde{\beta}\to+\infty} w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij}) = 1$;*
3) *$w_{ij}(\tilde{\beta},\tilde{\gamma},\varepsilon_{ij})$ is an inverse-'S' shaped function, which approaches a binary function when $\tilde{\gamma} \to 0$ and the constant $1/2$ when $\tilde{\gamma} \to +\infty$.*

Each statement takes care of one target parameter with the others fixed to be constants.

Proof. It can be easily verified by the definition. □

Next, we concentrate on the convergence of the proposed naive BCD algorithm.

Theorem 2. *The sequence of* $\{\mathcal{J}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{W}^{(t)})\}$*, i.e. the energy of the objective in* (10)*, generated by the proposed BCD optimizer (Algorithm 1) converges monotonically.*

Proof. In terms of energy, the optimization nature of BCD ensures that:

$$\mathcal{J}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{W}^{(t)}) \geq \mathcal{J}(\mathbf{V}^{(t+1)},\mathbf{U}^{(t)},\mathbf{W}^{(t)}) \geq \mathcal{J}(\mathbf{V}^{(t)},\mathbf{U}^{(t+1)},\mathbf{W}^{(t)}) \geq \mathcal{J}(\mathbf{V}^{(t+1)},\mathbf{U}^{(t+1)},\mathbf{W}^{(t+1)}).$$

In other words, the energy gradually decreases as the involved three steps iterate. Further, the whole objective function (10) has a lower bound. Therefore, Algorithm 1 is guaranteed to converge monotonically. □

We notice that Theorem 2 provides a convergence guarantee for the energy of the objective function $\mathcal{J}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{W}^{(t)})$. However, the convergence of $\{\mathcal{J}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{W}^{(t)})\}$ cannot ensure the convergence of the variables.

In what follows, we shall consider the lemmas required by analysis on convergence and optimality of the designed ADMM-based ROUTE-LRMR algorithm.

Lemma 2. *Let* $\{(\mathbf{U}^{(t)},\mathbf{V}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)})\}$ *be a sequence generated by Algorithm 2. Then the sequence approaches to a feasible solution.*

Proof. First, we prove the boundedness of $\{\mathbf{Z}^{(t)}\}$. According to Theorem 1 and the optimality condition for (19) with respect to $\hat{\mathbf{L}} \equiv \mathbf{UV}$, we have:

$$\mathbf{Z}^{(t-1)} + \mu^{(t-1)}(\mathbf{L}^{(t)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)}) = \mathbf{Z}^{(t)} \in \partial\|\hat{\mathbf{L}}^{(t)}\|_*.$$

Through applying Lemma 3 on the above:

Lemma 3. *[38] Let* $\mathcal{H}$ *be a real Hilbert space endowed with an inner product* $\langle\cdot,\cdot\rangle$ *and a corresponding norm* $\|\cdot\|$*, and any* $\mathbf{y} \in \partial\|\mathbf{x}\|$*, where* $\partial\|\cdot\|$ *denotes the subgradient. Then* $\|\mathbf{y}\|^* = 1$ *if* $\mathbf{x} \neq 0$*, and* $\|\mathbf{y}\|^* \leq 1$ *if* $\mathbf{x} = 0$*, where* $\|\cdot\|^*$ *is the dual norm of the norm* $\|\cdot\|$*.*

we obtain that the sequence $\{\mathbf{Z}^{(t)}\}$ is bounded via observing the fact that the dual norm of $\|\cdot\|_*$ is the spectral norm. Together with the boundedness of $\{\mathbf{Z}^{(t)}\}$ and $\lim_{t\to\infty}\mu^{(t)} = \infty$, the relationship $\mathbf{L}^{(t)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)} = \frac{\mathbf{Z}^{(t)}-\mathbf{Z}^{(t-1)}}{\mu^{(t-1)}}$ gives $\lim_{t\to\infty}\mathbf{L}^{(t)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)} = \mathbf{0}$. Further, the constraints of $\mathbf{W}+\overline{\mathbf{W}} = \mathbf{1}$ and $\mathbf{W},\overline{\mathbf{W}} \in [0,1]^{m\times n}$ are immediately satisfied at each update, please see Lemma 1. Thus the claim holds. □

Lemma 4. *Let* $\{(\mathbf{U}^{(t)},\mathbf{V}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)})\}$ *be a sequence generated by Algorithm 2. Then, we have two claims:*

1) *All of the sequences* $\{\mathbf{U}^{(t)}\}$*,* $\{\mathbf{V}^{(t)}\}$*,* $\{\mathbf{U}^{(t)}\mathbf{V}^{(t)}\}$*,* $\{\mathbf{L}^{(t)}\}$*, and* $\{\mathbf{W}^{(t)}\}$ *are bounded.*
2) *The sequences* $\{\mathbf{U}^{(t)}\mathbf{V}^{(t)}\}$*,* $\{\mathbf{L}^{(t)}\}$*, and* $\{\mathbf{W}^{(t)}\}$ *are Cauchy sequences.*

Proof. We here prove the first claim. By the nature of the iterative procedure of Alg. 2, the following relationship holds:

$$\begin{aligned}
&\mathcal{L}^{\mu^{(t)}}(\mathbf{V}^{(t+1)},\mathbf{U}^{(t+1)},\mathbf{L}^{(t+1)},\mathbf{W}^{(t+1)},\mathbf{Z}^{(t)})\\
\leq&\mathcal{L}^{\mu^{(t)}}(\mathbf{V}^{(t+1)},\mathbf{U}^{(t+1)},\mathbf{L}^{(t+1)},\mathbf{W}^{(t)},\mathbf{Z}^{(t)})\\
\leq&\ldots \leq \mathcal{L}^{\mu^{(t)}}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)},\mathbf{Z}^{(t)})\\
=&\mathcal{L}^{\mu^{(t-1)}}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)},\mathbf{Z}^{(t-1)})\\
&+\frac{\mu^{(t-1)}+\mu^{(t)}}{2\mu^{(t-1)2}}\|\mathbf{Z}^{(t)}-\mathbf{Z}^{(t-1)}\|_F^2.
\end{aligned} \tag{25}$$

Because of the boundedness of the sequence $\{\mathbf{Z}^{(t)}\}$ and $\sum_{t=1}^{\infty}\frac{\mu^{(t-1)}+\mu^{(t)}}{2\mu^{(t-1)2}} = \frac{\rho(1+\rho)}{2\mu^{(0)}(\rho-1)} < \infty$, it is ready to draw that the sequence $\{\mathcal{L}^{\mu^{(t-1)}}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)},\mathbf{Z}^{(t-1)})\}$ is upper bounded. Moreover, we have:

$$\begin{aligned}
&\frac{1}{2}\|\mathbf{U}^{(t)}\|_F^2+\frac{1}{2}\|\mathbf{V}^{(t)}\|_F^2+\frac{\alpha}{2}\|\sqrt{\mathbf{W}^{(t)}}\odot(\mathbf{Y}-\mathbf{L}^{(t)})\|_F^2\\
&+\beta\|\overline{\mathbf{W}}^{(t)}\|_1+\gamma\sum_{i,j}(w_{ij}^{(t)}\log w_{ij}^{(t)}+\bar{w}_{ij}^{(t)}\log\bar{w}_{ij}^{(t)})\\
=&\mathcal{L}^{\mu^{(t-1)}}(\mathbf{V}^{(t)},\mathbf{U}^{(t)},\mathbf{L}^{(t)},\mathbf{W}^{(t)},\mathbf{Z}^{(t-1)})+\\
&\frac{\|\mathbf{Z}^{(t-1)}\|_F^2-\|\mathbf{Z}^{(t)}\|_F^2}{2\mu^{(t-1)}},
\end{aligned} \tag{26}$$

is upper bounded. Due to the property of the weight matrix $\{\mathbf{W}^{(t)}\}$, its boundedness is fulfilled naturally. As for $\{\mathbf{V}^{(t)}\}$, $\{\mathbf{U}^{(t)}\}$, $\{\mathbf{L}^{(t)}\}$ and $\{\mathbf{U}^{(t)}\mathbf{V}^{(t)}\}$, Eq. (26) tells that they are all bounded. This establishes the proof of the first claim.

For proving the second claim, an auxiliary variables is required, which is defined as

$$\bar{\mathbf{Z}}^{(t)} \equiv \mathbf{Z}^{(t-1)} + \mu^{(t-1)}(\mathbf{L}^{(t-1)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)}). \tag{27}$$

The boundedness of $\bar{\mathbf{Z}}^{(t)}$ can be achieved in the same way with that of $\mathbf{Z}^{(t)}$ as given in the proof of Lemma 2. With $\|\mathbf{L}^{(t)} - \mathbf{L}^{(t-1)}\| = \frac{1}{\mu^{(t-1)}}\|\mathbf{Z}^{(t)} - \bar{\mathbf{Z}}^{(t)}\| = o(\frac{1}{\mu^{(t-1)}})$ and $\sum_{t=1}^{\infty}\frac{1}{\mu^{(t-1)}} = \frac{\rho}{\mu^{(0)}(\rho-1)} < \infty$, we have that $\{\mathbf{L}^{(t)}\}$ is a Cauchy sequence. Further by the feasibility of the solver as shown in Lemma 2, $\{\mathbf{U}^{(t)}\mathbf{V}^{(t)}\}$ is also a Cauchy sequence. Based on the closed-form solution of $\mathbf{W}^{(t)}$ (23), it is ready to conclude that $\{\mathbf{W}^{(t)}\}$ is a Cauchy sequence. This completes the proof of the second claim. □

Having the above theoretical results, we finally come to the convergence and optimality of Algorithm 2.

Theorem 3. *The proposed Algorithm 2 converges to a KKT point to the optimization problem* (10)*.*

Proof. By Lemmas 1, 2 and 4, the KKT conditions for the constraints and the solutions to the variables $\mathbf{UV}$, $\mathbf{L}$ and $\mathbf{W}$ are all satisfied. According to the Bolzano-Weierstrass Theorem, the sequence has at least one accumulation point. Moreover, we know from Eq. (21) that the closed-form solutions to $\mathbf{U}$ and $\mathbf{V}$ are unique when one of them is fixed because $\mathbf{I}+\mu^{(t)}\mathbf{U}^{(t)T}\mathbf{U}^{(t)}$ and $\mathbf{I}+\mu^{(t)}\mathbf{V}^{(t+1)}\mathbf{V}^{(t+1)T}$ are both positive definite. Combining all the above together, it suffices to guarantee that the ADMM-based ROUTE-LRMR algorithm converges to a KKT point to the problem (10). □

Method	PRMF	PCP	MoG	RegL1	factEN	Unifying	PSSV	KDRS	HW	ROUTE-BCD	ROUTE-ADMM
$\text{RMSE}_{s=0.3}$	0.1106	0.1009	0.7440	1.2789	0.2121	*0.0608*	1.0214	1.2463	0.2731	0.3627	**0.0523**
$\text{MAE}_{s=0.3}$	0.0573	0.0776	0.0984	0.2844	0.1069	*0.0457*	0.2813	0.2740	0.2084	0.2166	**0.0445**
$\text{RMSE}_{s=0.4}$	0.5803	0.1484	0.9796	1.6146	0.3731	*0.0762*	1.9496	1.4495	0.4089	0.4005	**0.0624**
$\text{MAE}_{s=0.4}$	0.1716	0.1101	0.1430	0.4432	0.2030	*0.0565*	0.7361	0.4035	0.3114	0.2391	**0.0480**
$\text{RMSE}_{s=0.5}$	1.0920	0.4367	1.3344	1.9255	0.5181	*0.0975*	2.7295	1.8338	0.7953	0.4629	**0.0676**
$\text{MAE}_{s=0.5}$	0.3719	0.2967	0.2201	0.6576	0.2868	*0.0719*	1.4310	0.6268	0.5944	0.2574	**0.0520**
$\text{RMSE}_{s=0.6}$	1.6075	1.3518	1.6381	2.4863	0.6861	*0.1492*	3.7988	2.4666	0.9730	0.4896	**0.1092**
$\text{MAE}_{s=0.6}$	0.6287	0.9268	0.3607	1.0438	0.4048	*0.1093*	2.3843	1.0554	0.7236	0.2624	**0.0651**
$\text{RMSE}_{s=0.7}$	2.2421	2.6602	1.9513	3.2555	0.8734	*0.3566*	4.8753	3.1769	1.7483	0.5815	**0.3294**
$\text{MAE}_{s=0.7}$	1.0660	1.9287	0.4956	1.7142	0.5677	*0.2352*	3.3894	1.6581	1.2933	0.2989	**0.2088**

TABLE I: Performance comparison in RMSE and MAE with different outlier ratios s. The numbers are averaged over 10 runs. The best results are highlighted in bold. The second best results are in italic and underlined. The third places are underlined.

$r=4\|r_{gt}=4$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	0.0576	**0.0435**	0.9	0.0334	0.0256	6.2	0.0254	0.0196	14.5	**0.0227**	**0.0175**	21.9	**0.0186**	**0.0143**	49.1
ROUTE-BCD	0.6059	0.2549	5.3	0.4353	0.2291	17.6	0.0519	0.0324	39.5	0.0233	0.0180	50.6	0.0190	0.0147	96.6
ROUTE-ADMM	**0.0489**	0.0500	**0.4**	**0.0325**	**0.0251**	**2.1**	**0.0251**	**0.0194**	**5.4**	0.0230	0.0177	**8.2**	0.0187	0.0144	**19.3**
$r=8\|r_{gt}=4$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	0.0881	0.0647	0.9	0.0486	0.0372	6.4	0.0387	0.0298	14.9	0.0350	0.0271	22.9	**0.0286**	**0.0222**	51.8
ROUTE-BCD	0.5088	0.2792	5.7	0.5536	0.2882	22.2	0.3929	0.2074	44.0	0.0743	0.0385	62.1	0.0297	0.0229	118.9
ROUTE-ADMM	**0.0665**	**0.0520**	**0.4**	**0.0462**	**0.0361**	**2.1**	**0.0380**	**0.0297**	**5.4**	**0.0345**	**0.0270**	**8.5**	**0.0286**	0.0224	**20.0**
$r=20\|r_{gt}=10$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	0.4328	0.2295	1.0	0.1057	0.0725	6.9	0.0673	0.0522	15.8	0.0590	0.0460	24.4	0.0472	0.0370	55.4
ROUTE-BCD	0.4536	0.2450	8.1	0.5549	0.3439	29.7	0.7209	0.4683	59.0	0.6972	0.4452	83.3	0.4073	0.2378	167.2
ROUTE-ADMM	**0.1773**	**0.0906**	**0.4**	**0.0667**	**0.0527**	**2.2**	**0.0558**	**0.0441**	**5.5**	**0.0511**	**0.0404**	**8.6**	**0.0432**	**0.0342**	**20.5**
$r=20\|r_{gt}=20$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	0.4586	0.2880	1.0	0.0907	0.0709	6.8	0.0631	0.0497	15.7	0.0553	0.0436	24.4	0.0434	0.0343	55.6
ROUTE-BCD	0.4964	0.2981	8.1	0.6110	0.3975	30.0	0.7799	0.5369	59.1	0.7319	0.4902	84.1	0.5017	0.3179	167.6
ROUTE-ADMM	**0.1975**	**0.1143**	**0.5**	**0.0692**	**0.0546**	**2.2**	**0.0547**	**0.0433**	**5.4**	**0.0502**	**0.0431**	**8.5**	**0.0413**	**0.0327**	**20.5**
$r=40\|r_{gt}=20$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	1.8016	1.0150	1.2	1.6310	0.4539	7.2	0.5962	0.1251	16.6	0.2650	0.0825	25.8	0.0737	0.0576	58.6
ROUTE-BCD	**0.4936**	**0.2469**	11.4	0.4073	0.2560	44.4	0.4299	0.2782	96.0	0.5438	0.3652	134.8	0.7400	0.5027	262.8
ROUTE-ADMM	0.7372	0.3812	**0.5**	**0.0908**	**0.0700**	**2.3**	**0.0735**	**0.0582**	**5.7**	**0.0675**	**0.0535**	**8.8**	**0.0576**	**0.0458**	**21.0**
$r=40\|r_{gt}=40$	$m=n=200$			$m=n=500$			$m=n=800$			$m=n=1000$			$m=n=1500$		
$s=0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
Unifying	1.8280	1.0382	1.2	1.5511	0.4861	7.2	0.4936	0.1564	16.5	0.0922	0.0723	25.7	0.0665	0.0527	59.1
ROUTE-BCD	**0.5251**	**0.2696**	11.5	0.4565	0.2952	45.6	0.4857	0.3306	91.6	0.5662	0.3885	133.0	0.8611	0.4271	278.4
ROUTE-ADMM	0.7650	0.4101	**0.5**	**0.0953**	**0.0747**	**2.3**	**0.0757**	**0.0598**	**5.6**	**0.0686**	**0.0543**	**8.6**	**0.0570**	**0.0452**	**20.7**

TABLE II: Performance comparison in terms of RMSE, MAE and Time between Unifying, ROUTE-BCD and ROUTE-ADMM

Further, based on Theorem 1, the problem (10) is equivalent to the following one:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{W}} \ & \|\mathbf{L}\|_* + \frac{\alpha}{2}\|\sqrt{\mathbf{W}} \odot (\mathbf{Y}-\mathbf{L})\|_F^2 + \beta\|\overline{\mathbf{W}}\|_1 \\ & + \gamma\sum_{i,j}(w_{ij}\log w_{ij} + \bar{w}_{ij}\log \bar{w}_{ij}) \qquad (28)\\ \text{s.t.} \quad & \mathbf{W}+\overline{\mathbf{W}} = \mathbf{1}; \quad \mathbf{W} \text{ and } \overline{\mathbf{W}} \in [0,1]^{m\times n},\end{aligned}$$

which is biconvex in $\mathbf{W}$ and $\mathbf{L}$. The convergence to a KKT point holds for the problem (28) too. Our ROUTE-LRMR is free to switch modes between NNM and BF. Concretely, instead of separately refreshing $\mathbf{U}$ and $\mathbf{V}$, the updating of $\hat{\mathbf{L}} \equiv \mathbf{UV}$ in (21) can be done by minimizing the problem:

$$\min_{\hat{\mathbf{L}}} \quad \|\hat{\mathbf{L}}\|_* + \frac{\mu^{(t)}}{2}\|\mathbf{L}_{(p)} - \hat{\mathbf{L}}\|_F^2 + \langle \mathbf{Z}^{(t)}, \mathbf{L}_{(p)} - \hat{\mathbf{L}}\rangle, \quad (29)$$

which can be solved in closed-form by the singular value thresholding [39]. Except for this step, no other changes happen in Algorithm 2.

Complexity Analysis We first discuss the complexity of Alg. 1 (the BCD optimizer). Updating each column $\mathbf{v}_j$ at each iteration spends $\mathcal{O}(r^3+r^2+r^2m+rm)$. Thus, each update of $\mathbf{V}$ costs $\mathcal{O}(n(r^3+r^2+r^2m+rm))$ in total. Similarly, each update of $\mathbf{U}$ takes $\mathcal{O}(m(r^3+r^2+r^2n+rn))$. Refreshing $\mathbf{W}$ only needs $\mathcal{O}(rmn)$. Hence, the time complexity of the BCD optimizer is $\mathcal{O}(t_{bcd}(r^3(m+n)+r^2(m+n)+2r^2mn+3rmn)))$, where t_{bcd} is the iteration number required to converge. As for the ADMM optimizer, updating $\mathbf{V}$ and $\mathbf{U}$ both demand $\mathcal{O}(r^3 + r^2 + r^2(m+n) + rmn)$. While solving the $\mathbf{W}$ sub-problem and updating the multiplier need $\mathcal{O}(mn)$ and $\mathcal{O}(rmn)$, respectively. Thus, the whole procedure of Alg. 2 spends $\mathcal{O}(t_{admm}(2r^3 + 2r^2 + 2r^2(m+n) + 3rmn + mn))$ with t_{admm} the total iteration number that Alg. 2 takes for convergence. From the above analysis, we see that Alg. 2 is much more efficient than Alg. 1 for each iteration. We will compare the two algorithms in terms of convergence speed, elapsed time and recovery accuracy in Sec. V.

IV. Related Work

So far, a large body of research about LRMR has been carried out. We briefly review classic and recent achievements closely related with ours, which are basically derived from

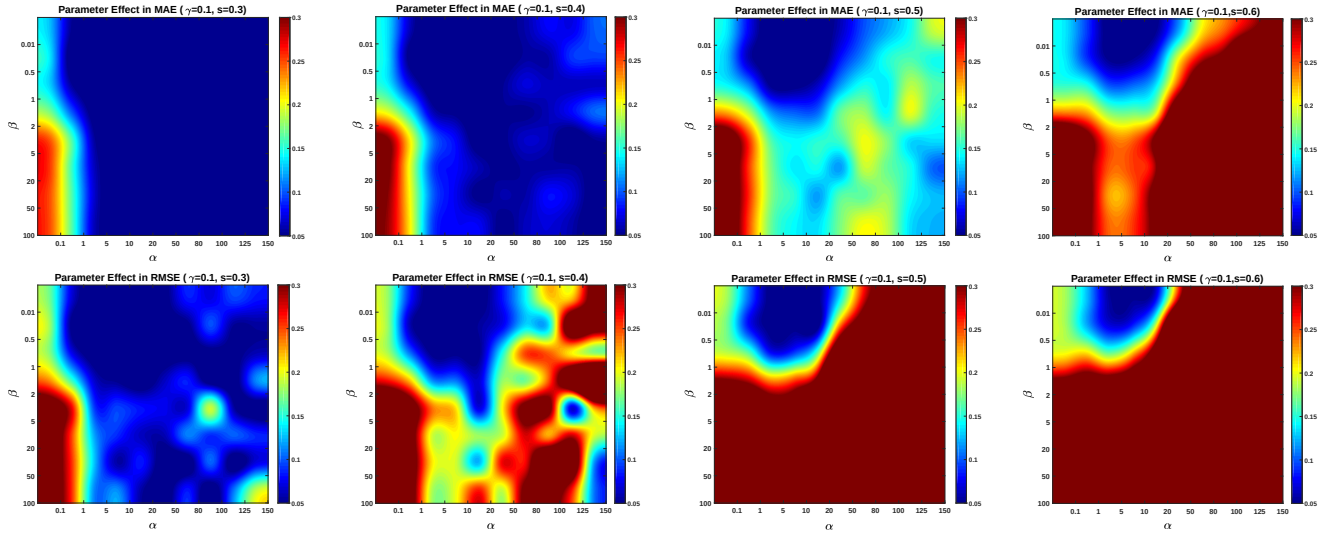

Fig. 1: Parameter effect of α and β in terms of MAE (upper row) and RMSE (lower row). The pictures correspond to the cases with fixed $\gamma = 0.1$ and different outlier ratios $s \in \{0.3, 0.4, 0.5, 0.6\}$ from left to right.

the NNM and BF models. PCA [1] follows the NNM with the ℓ^2 loss by assuming the residual existing in the observation satisfies a Gaussian distribution, while PCP [2] takes care of arbitrary outliers by adopting the ℓ^1 penalty. To accelerate PCP, Zhou and Tao developed GoDec [40] by using bilateral random projections based approximation. Recently, Oh *et al.* proposed an approximate *singular value thresholding* (SVT) method that exploits the property of iterative NNM procedures by range propagation and adaptive rank prediction [41]. Since conventional NNM based approaches do not fully utilize priori target rank information about the problems when the exact rank of clean data is given, PSSV [42] attempts to minimize partial sum of singular values in PCP, which behaves better than PCP when the number of samples is deficient. Cabral *et al.* proposed a method Unifying [25] that unifies nuclear norm and bilinear factorization as below:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V})\|_1.$$

More recently, Lin *et al.* proposed to solve the Unifying model by majorization minimization for seeking better solutions [43]. To further improve the stability of Unifying when highly corrupted data are presented, factEN [44] employs the Elastic-Net regularization on the factor matrices as:

$$\min_{\mathbf{U},\mathbf{V}} \frac{\lambda_1}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \frac{\lambda_2}{2}\|\mathbf{P}\|_F^2 + \|\mathbf{W} \odot (\mathbf{Y} - \mathbf{P})\|_1$$

with the definition $\mathbf{P} = \mathbf{U}\mathbf{V}$. As a hybrid of NNM and BF, RegL1 [23] solves the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{V}\|_* + \lambda\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V})\|_1 \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I},$$

which reduces the cost of PCP by calculating SVDs on a small matrix $\mathbf{V}$. *Robust bilinear factorization* (RBF) [45] shares the same model with RegL1 with different solving details. In parallel, there are a few works developed from probabilistic standpoints. PMF [21] and PRMF [24] are two representatives, corresponding to PCA and PCP, respectively. Meng and De la Torre [20] improved the BF via modeling the unknown noises as a *mixture of Gaussian distributions* (MoG). More recently, Bahri *et al.* proposed a Kronecker-decomposable component analysis (KDRS) [46], which combines ideas from sparse dictionary learning and PCP.

V. Experimental Verification

In this section, we assess the performance of ROUTE-LRMR in comparison with several state-of-the-art methods including RegL1 [23], PCP [2], PRMF [24], MoG [20], factEN [44], PSSV [42], Unifying [25] and KDRS [46], the codes of which are either downloaded from the authors' websites or provided by the authors. Their settings follow the suggestions by the authors or the given parameters. All the experiments are conducted on a PC running Windows 7 64bit operating system with Intel Core i7 2.5 GHz CPU and 64.0 GB RAM.

A. Synthetic Data

Data Preparation and Quantitative Metrics Similar to [2], [25], we generate a matrix $\mathbf{Y}_0$ as a product $\mathbf{Y}_0 = \mathbf{U}_0\mathbf{V}_0$. The $\mathbf{U}_0$ and $\mathbf{V}_0$ are of size $m \times r$ and $r \times n$ respectively, both of which are randomly produced by sampling each entry from the Gaussian distribution $\mathcal{N}(0, 1)$, leading to a ground truth rank-r matrix. Then we corrupt the entries via replacing a fraction s of $\mathbf{Y}_0$ with errors drawn from a uniform distribution over $[-20, 20]$ at random, and the rest entries are polluted by Gaussian noise $\mathcal{N}(0, 0.1^2)$. To quantitatively measure the recovery performance, we employ 1) *root mean square error* (RMSE): $\frac{1}{\sqrt{mn}}\|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}\|_F$ and 2) *mean absolute error* (MAE): $\frac{1}{mn}\|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}\|_1$ as our metrics. Lower values of both the metrics indicate better performance.

Parameter effect There are three parameters, including α, β and γ, involved in Eq. (10). This part experimentally evaluates how these parameters influence the performance. In this experiment, without loss of generality, square matrices of dimension $m = n = 100$ and rank $r = 4$ are considered. We

note that, ROUTE-BCD and ROUTE-ADMM solve the same problem (10). Thus, from the parameter perspective, the effect should be similar for the two solvers. In this part, we merely test the parameter effect on ROUTE-ADMM. Later we will see the performance comparison between the two solvers. We first test the parameters α and β with γ set to 0.1. Figure 1 displays four α-β testings with respect to different outlier ratios $s \in \{0.3, 0.4, 0.5, 0.6\}$ in terms of MAE (upper row) and RMSE (lower row). Each graph is generated by averaging 10 independent runs. As can be seen from the pictures, as s decreases, the work range enlarges. Even though, for all of the four cases, the top-left region (small β, small α) is in trouble. This is because of the under-penalization on the pollution. The bottom-left region (large β, small α) also reflects poor performance. The reason is that, as stated in the second claim in Proposition 1, when β/α is large, each w_{ij} approaches to 1. As a result, together with a small α, the residual component, including both noises and outliers, is less cared. Moreover, considering the top-right region (small β, large α), especially for $s \in \{0.5, 0.6\}$, the values in MAE and RMSE are high. Similarly to the bottom-left region, the pollution is under-penalized. But differently the under-penalization comes from that most $\bar{w}_{ij}$ approaching to 1 (please see Eq. (23)) and a small β. As s increases, the bottom-right region (large β, large α) turns red. This is because the pollution is over-penalized, leading to inaccurate recovery. Under this experimental setting, $\alpha \in [1, 20]$ and $\beta \in [0.01, 1]$ consistently provide reasonable results for all the involved situations. We now focus on the parameter γ that controls the entropy term, the other two parameters α and β are fixed to 50 and 1, respectively. Figure 2 depicts RMSE and MAE curves (averaged over 10 independent trials) with respect to different outlier ratios. From the plots, we see that when γ approaches to 0, the errors rapidly go up. This is because, as analyzed in Sec. III, the smaller γ is, the harder the weighting carries out, say the risk of being stuck into bad minima gets higher. It is also the evidence to prove the soft weighting is beneficial. In opposite, if γ gets too large, the performance also drops. The reason is that, in this situation, the weighting becomes almost constant (0.5 for each entry), which degenerates ROUTE-LRMR to PCA. Although the work range of γ shrinks as s grows, γ in $[0.005, 0.8]$ can perform stably and sufficiently well. For the rest experiments unless stated otherwise, we set $\alpha = 50$, $\beta = 1$, and $\gamma = 0.01$. To better reveal the advantage of our method over the competitors especially on heavily ruined data, Table I reports the numerical comparison. As can be seen from Tab. I, ROUTE-ADMM wins for all the cases, and the closest performance to ours is from Unifying. The main reason for the inferior performance of ROUTE-BCD is that it is sensitive to the initialization and has high risk of being early stuck into bad minimum, we will further confirm this in the coming part. Please note that the method HW is ROUTE-ADMM with $\gamma = 0.001$ for mimicking the hard weighting.

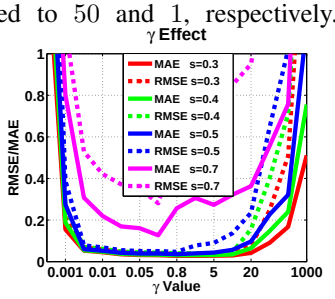

Fig. 2: Parameter effect of γ

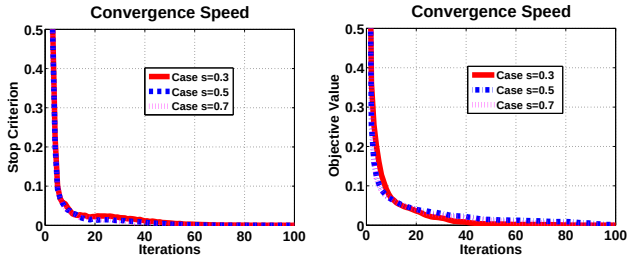

Fig. 3: Convergence speed. The left and right graphs correspond to ROUTE-ADMM and ROUTE-BCD, respectively.

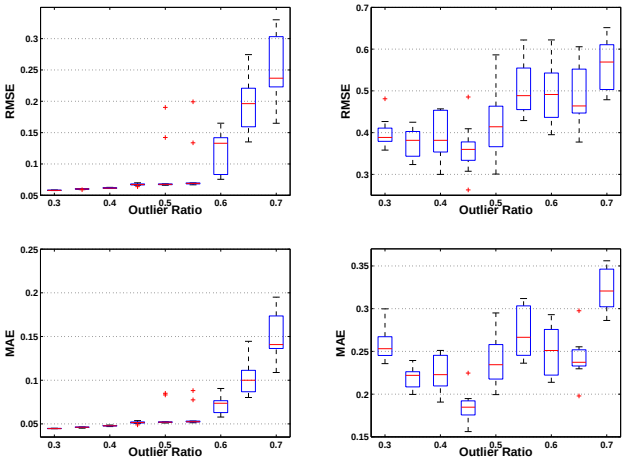

Fig. 4: Initialization sensitivity test. The two columns correspond to ROUTE-ADMM and ROUTE-BCD, respectively.

Convergence Behavior As regards convergence speed, for different cases, the left graph in Fig. 3 shows that the stop criterion of ROUTE-ADMM quickly declines within 20 iterations, while the algorithm converges within $60 \sim 80$ iterations. The right picture in Fig. 3 corresponds to ROUTE-BCD, which behaves similarly with ROUTE-ADMM in terms of the iteration number required to converge, *i.e.* $t_{admm} \simeq t_{bcd}$. But, for each iteration, ROUTE-ADMM needs much less computational resource than ROUTE-BCD does, as analyzed in Sec. III. Table II offers an empirical comparison in terms of recovery accuracy and time cost, with the outlier ratio fixed to 0.5. The numbers are averaged over 10 independent runs. From the table, we can see that ROUTE-ADMM is significantly faster than ROUTE-BCD and Unifying, the gain of which becomes conspicuous as the data size increases. It is worth noting that Unifying requires inner loops to update $\mathbf{U}$ and $\mathbf{V}$ (please refer to [25] for details, and we set the maximal inner iteration number to 10), while our ROUTE-ADMM has no such requirement. In terms of accuracy, ROUTE-ADMM wins over the other two in most cases. We notice that the accuracy margin is more obvious when the ratio of the data size versus the intrinsic rank is relatively small. The differences in RMSE and MAE between Unifying and ROUTE-ADMM shrink as the ratio increases. In addition, ROUTE-BCD loses the competition, because, as mentioned, it may easily fall into bad minimum during optimization and has

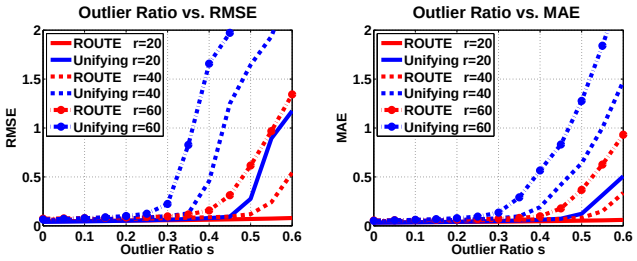

Fig. 5: Outlier ratio s versus RMSE and MAE

a higher computational complexity. Moreover, experimental findings here and follow-up tell that our ROUTE-ADMM has very stable convergence behavior even with respect to random initializations (please see the next part).

Initialization Sensitivity Directly applying BCD has a higher probability of being stuck at bad optima than using ADMM. Besides the theoretical analysis in Sec. III, we here give an intuitive explanation. The main reason comes from the $\mathbf{W}$ sub-problem. Suppose that the variables $\mathbf{U}$ and $\mathbf{V}$ are randomly initialized. After updating $\mathbf{U}$ and $\mathbf{V}$ at early iterations, if the initialization is unsatisfactory (typically not), $\mathbf{W}$ can be wrongly determined because of large residuals $(\mathbf{Y}-\mathbf{UV})$. Please see the solution in Eq. (18). This will easily lead the solver to a bad optimum or even a trivial solution. One possible strategy to mitigate the above issue is putting the update of $\mathbf{W}$ out of the loop of iterating $\mathbf{U}$ and $\mathbf{V}$. However, in this way, the ℓ^2 loss on $||\sqrt{\mathbf{W}}\odot(\mathbf{Y}-\mathbf{UV})||_F^2$ is no longer effective as it is short of ability to handle outliers. So a more robust loss is required. If simply adopting the ℓ^1 loss, as it is non-differentiable, BCD is not appropriate. In contrary, ADMM can do the job in a more faithful way. At early iterations, μ is relatively small and $\mathbf{L}$ is close to $\mathbf{Y}$. So the inaccuracy in $\mathbf{W}$ can be largely avoided. As μ grows, the solver converges. Thus, ROUTE-ADMM can perform stably to random initializations. Figure 4 provides the empirical evidence in RMSE versus outlier ratio (upper row) and MAE versus outlier ratio (lower row). The left and right columns correspond to ROUTE-ADMM and ROUTE-BCD, respectively. The box-plots are formed by 10 runs on the $m=n=100$ and $r=4$ case with random initializations. From the plots, we can see that ROUTE-ADMM performs accurately and stably even when the outlier ratio is up to 0.55, while ROUTE-BCD has large medians and wide 95% confidences. Further, the median of ROUTE-ADMM increases as the outlier ratio grows, which corroborates to the common sense, while ROUTE-BCD does not show such a property due to its initialization sensitivity. In the following experiments, we will focus on ROUTE-ADMM.

Tolerance to Outliers To more thoroughly show the tolerance to outliers, we fix $m=n=400$ and test the tendency by varying outlier ratio $s\in[0,0.6]$ and rank $r\in\{20,40,60\}$. According to the results in Tab. I, Unifying is the method chosen to compare. From the left picture of Fig. 5, we see that at the beginning, Unifying and our method are close in terms of RMSE, but as s increases, the margin between them enlarges. The second graph in Fig. 5 further confirms the first one. In the case of $r=20$, both the RMSE and MAE of ROUTE-LRMF stay very low even when s reaches 0.6. The tolerance to outliers becomes weaker when r gets larger, not just for our method and Unifying but also for all the methods. The reason is that a higher-dimensional space requires more data to accomplish the recovery.

B. Real Data

Photometric Stereo Images of a static Lambertian object sensed by a fixed camera under a varying but distant point lighting source lie in a rank-3 subspace [11]. This experiment aims to evaluate the effectiveness of the LRMR techniques on modeling the face under different illuminations. The cropped Extended YaleB-10 sequence, containing 64 faces of one subject with size 192×168, is adopted as the dataset. The light imbalance including shadows and highlights on the face significantly breaks the low-rank structure (please see the 1^{st} column in Fig. 6 for example). In this part, we set the guess rank r to 5 for all the competitors.

Comparison Figure 6 gives several comparison. We can observe that PRMF, factEN, MoG, KDRS and Unifying perform reasonably well, which are superior to PSSV and RegL1 but inferior to ours. As shown in the 2^{nd} and 4^{th} rows of Fig. 6, PSSV and RegL1 fail to remove shadows. The results by PRMF, factEN, MoG, KDRS and Unifying, although recalling some details previously hidden in the dark, look unreal in the 2^{nd} and 3^{rd} cases. Our ROUTE-LRMR[1] provides visually pleasant and real results for all the given cases, the benefit of which mainly comes from the effective outlier detection. The 2^{nd} column in Fig. 6 displays the estimated weights $\mathbf{W}$ (brighter regions indicate closer values to 1, while darker ones stand for those to 0), from which we can find our strategy successfully detects and thus eliminates outliers. Figure 7 furthers provide several results by our method. One may wonder if the weights can be formed by treating as outliers the pixels with intensity greater (highlights) or lower (shadows) than predefined thresholds like [23]. This way can reduce the problem to LRMC, but is too heuristic, at high risk of sacrificing much useful information for recovery. Taking the bottom-right original for example, the thresholding may determine all the pixels as outliers, while our strategy can finish the job wisely and nicely. Moreover, in many real-world applications, manually seeking appropriate thresholds is, if not impossible, very difficult. Being able to adaptively assign weights to data is definitely desired, which is the goal and motivation of our design.

Background Modeling The problem of background modeling for surveillance videos can be viewed as a decomposition of a video into the foreground component and the background. This experiment is carried out on the WaterSurface sequence[2], which contains 633 frames with resolution 128×160. We assume that the background of the sequence is rank-1 and use only 130 frames (frame #481-frame #610) to accomplish the

[1]In image/video data, the outliers, such as shadows and foregrounds, often appear coherently. Considering this, in this experiment, we employ a 2×2 median filter on $\mathbf{W}$.

[2]http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

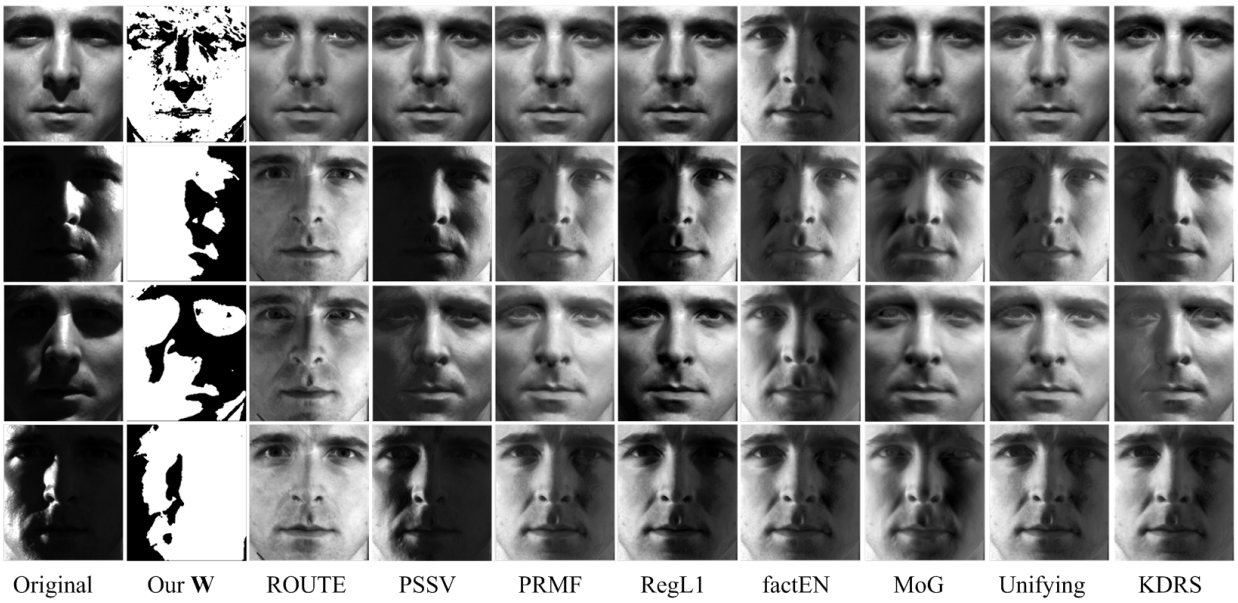

Fig. 6: Visual comparison on the task of photometric stereo. ROUTE-ADMM adopts $\gamma = 0.001$ in this experiment.

comparison. The foreground person occupies a large portion of the frames and the water surface is flowing, which ruins the rank-1 background. In addition, for making the recovery more challenging, we further introduce Gaussian white noises with variance $\sigma^2 = 0.005$ into the data and randomly discard 25% pixels as missing elements. Figure 8 gives a sample frame. The first to the third pictures in the upper row are the original frame, the noisy version and, the noisy & incomplete input, respectively.

Comparison Figure 8 (e)-(g) are the backgrounds obtained by factEN, RegL1 and Unifying, respectively, from which we can clearly find the ghosts left in the background. That is to say, all of factEN, RegL1 and Unifying are not capable to handle the data sufficiently well, because of the limited samples, the gross outliers, the noises and the missing pixels. In comparison, our proposed ROUTE-ADMM can significantly outperform the competitors, as shown in the last picture of Fig. 8, which benefits from the outlier estimation. The estimated weight $\mathbf{W}$ is given in Fig. 8 (d). Please notice that the dark regions in (d) contain the missing elements weighted by zero for unifying LRMR and LRMC.

Fig. 7: More results by ROUTE-ADMM

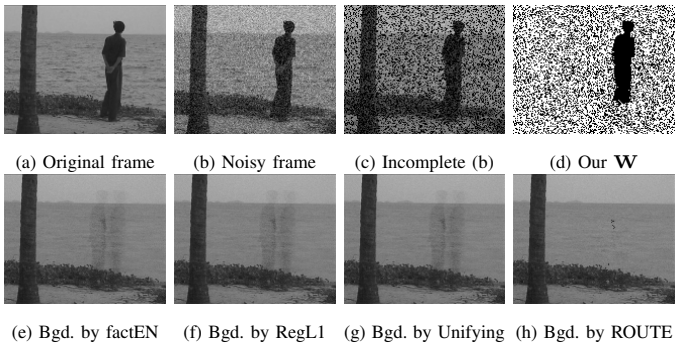

Fig. 8: Visual comparison on the task of background modeling

VI. Conclusion

This paper has shown a method for jointly detecting outliers and recovering the underlying low-rank matrix, called ROUTE-LRMR. Our weighting strategy employs an entropy regularization term to minimize the prediction bias, which behaves like a sigmoid function. To seek the optimal solution for ROUTE-LRMR, we have developed a block coordinate descent based algorithm (ROUTE-BCD) and an alternating direction method of multipliers based one (ROUTE-ADMM). The theoretical analysis and the experimental results compared to the state-of-the-arts, have demonstrated the advantages of the proposed ROUTE-LRMR, with the evidence on the superiority of ROUTE-ADMM over ROUTE-BCD. Our strategy can be applied to numerous tasks such as regression, clustering, inpainting and foreground detection. It is also ready to embrace specific domain knowledge, like graph regularizer on the weight, for boosting the performance on different applications.

REFERENCES

[1] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, 1901.

[2] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[3] Q. Zhang and H. Wang, "Collaborative filtering with generalized laplacian constraint via overlapping decomposition," in *IJCAI*, 2016.

[4] C. Chen, X. Zheng, Y. Wang, F. Hong, and Z. Lin, "Context-aware collaborative topic regression with social matrix factorization for recommender systems," in *AAAI*, 2014.

[5] F. Nie and H. Huang, "Subspace clustering via new discrete group structure constrained low-rank model," in *IJCAI*, 2016.

[6] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Yu, "Robust recovery of subspace structures by low-rank representation," *TPAMI*, vol. 35, no. 1, pp. 171–184, 2013.

[7] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with applications to image denoising," in *CVPR*, 2014.

[8] X. Guo, X. Cao, and Y. Ma, "Robust separation of reflection from multiple images," in *CVPR*, 2014.

[9] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *IJCV*, vol. 9, no. 2, pp. 137–154, 1992.

[10] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *CVPR*, 2000.

[11] H. Hayakawa, "Photometric stereo under a light source with arbitrary motion," *JOSA*, vol. 11, pp. 3079–3089, 1994.

[12] R. Basri, D. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *IJCV*, vol. 72, no. 3, pp. 239–257, 2007.

[13] S. Li, M. Shao, and Y. Fu, "Multi-view low-rank analysis for outlier detection," in *SDM*, 2015.

[14] S. Li, M. Shao, and Y. Fu, "Multi-view low-rank analysis with applications to outlier detection," *ACM Trans. on Knowledge Discovery from Data*, vol. 12, no. 3, pp. 32:1–32:22, 2018.

[15] X. Jing, X. Zhu, F. Wu, X. You, Q. Liu, D. Yue, R. Hu, and B. Xu, "Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning," in *CVPR*, 2015.

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

[17] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *TPAMI*, vol. 35, no. 3, pp. 597–610, 2013.

[18] A. Eriksson and A. van den Hengel, "Efficient computation of robust low-rank matrix approximation in the presence of missing data using the ℓ_1 norm," in *CVPR*, 2010.

[19] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *ICML*, 2003.

[20] D. Meng and F. De la Torre, "Robust matrix factorization with unknown noise," in *ICCV*, 2013.

[21] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*, 2008.

[22] B. Lakshminarayanan, G. Bouchard, and C. Archambeau, "Robust bayesian matrix factorisation," in *AISTATS*, 2011.

[23] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust l1-norm," in *CVPR*, 2012.

[24] N. Wang, T. Yao, J. Wang, and D. Yeung, "A probabilistic approach to robust matrix factorization," in *ECCV*, 2012.

[25] R. Cabral, F. De la Torre, J. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *ICCV*, 2013.

[26] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *ISIT*, 2010.

[27] L. Zhao, P. Babu, and D. P. Palomar, "Efficient algorithms on robust low-rank matrix completion against outliers," *IEEE Trans. on Signal Processing*, vol. 26, no. 18, pp. 4767–4780, 2016.

[28] J. Pan, Z. Hu, Z. Su, and M. Yang, "L0-regularized intensity and gradient prior for deblurring text images and beyond," *TPAMI*, vol. 39, no. 2, pp. 342–355, 2017.

[29] X. Guo, X. Cao, X. Chen, and Y. Ma, "Video editing with temporal, spatial and appearance consistency," in *CVPR*, 2013.

[30] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?," *TIP*, vol. 24, no. 12, pp. 5017 – 5032, 2015.

[31] J. Zhao, Y. Lv, Z. Zhou, and F. Cao, "A novel deep learning algorithm for incomplete face recognition: Low-rank-recovery network," *Neural Networks*, vol. 97, pp. 115–124, 2017.

[32] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *CVPR*, 2017.

[33] Z. Ding, M. Shao, and Y. Fu, "Deep robust encoder through locality preserving low-rank dictionary," in *ECCV*, 2016.

[34] X. Guo and Z. Lin, "Route: Robust outlier estimation for low rank matrix recovery," in *IJCAI*, 2017.

[35] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *JMLR*, vol. 99, pp. 2287–2322, 2010.

[36] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming-Theory and Algorithms*. John Wiley and Sons Inc., 1993.

[37] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *NIPS*, 2011.

[38] M. Fazel, "Matrix rank minimization with applications," *PhD Thesis, Stanford University*, 2002.

[39] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[40] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *ICML*, 2011.

[41] T. Oh, Y. Matsushita, Y. Tai, and I. Kweon, "Fast randomized singular value thresholding for nuclear norm minimization," in *CVPR*, 2015.

[42] T. Oh, Y. Tai, J. Bazin, H. Kim, and I. Kweon, "Partial sum minimization of singular values in robust pca: Algorithm and applications," *TPAMI*, vol. 4, no. 3, pp. 744–758, 2016.

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

[44] E. Kim, M. Lee, and S. Oh, "Elastic-net regularization of singular values for robust subspace learning," in *CVPR*, 2015.

[45] F. Shang, Y. Liu, H. Tong, J. Cheng, and H. Cheng, "Robust bilinear factorization with missing and grossly corrupted observations," *Information Sciences*, vol. 307, pp. 53–72, 2015.

[46] M. Bahri, Y. Panagakis, and S. Zafeiriou, "Robust kronecker-decomposable component analysis for low-rank modeling," in *ICCV*, 2017.

Xiaojie Guo received the B.E. degree in software engineering from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2008, and the M.S. and Ph.D. degrees in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2010 and 2013, respectively. He is currently an Associate Professor with tenure (Peiyang Young Scientist) at Tianjin University. Prior to joining TJU, he spent about 4 years at the Institute of Information Engineering, Chinese Academy of Sciences. He was a recipient of the Piero Zamperoni Best Student Paper Award in the International Conference on Pattern Recognition (International Association on Pattern Recognition), in 2010. He is an Associate Editor of the IEEE Access.

Zhouchen LIN received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an area chair of ACCV 2009/2018, CVPR 2014/2016, ICCV 2015, NIPS 2015/2018 and AAAI 2019, and senior program committee member of AAAI 2016/2017/2018 and IJCAI 2016/2018. He is an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and the International Journal of Computer Vision. He is an IAPR/IEEE fellow.