Chapter 2
Sparse Structure for Visual Information Sensing: Theory and Algorithms

Abstract This chapter proposes to utilize sparse structure for visual information sensing and understanding. In detail, concentered on the fundamental theory of compressive sensing, we will discuss the problem of low-rank structure learning (LRSL) from sparse outliers. Different from traditional approaches, which directly utilize convex norms to measure the sparseness, our method introduces more reasonable non-convex measurements to enhance the sparsity in both the intrinsic low-rank structure and the sparse corruptions. Although the proposed optimization is no longer convex, it still can be effectively solved by a majorization–minimization (MM)-type algorithm. From the theoretic perspective, we have proved that the MM-type algorithm can converge to a stationary point after successive iterations. The proposed model is applied to solve a number of computer vision and information processing tasks, e.g., face image enhancement, object tracking, and time series clustering.

2.1 Introduction

Learning the intrinsic data structures via matrix analysis has received wide attention in many fields, e.g., neural networks [1], learning systems [2, 3], control theory [4], computer vision [5, 6], and pattern recognition [7, 8]. There are quite a number of efficient mathematical tools for rank analysis, e.g., principal component analysis (PCA) and singular value decomposition (SVD). However, these typical approaches could only handle some preliminary and simple problems. With the recent progresses of compressive sensing [9], a new concept on nuclear norm optimization has emerged into the field of rank minimization [10] and has led to a number of interesting applications, e.g., low-rank structure learning (LRSL) from corruptions.

Corrupted matrix recovery [11] considers decomposing a low-rank matrix from sparse corruptions which can be formulated as $\mathbf{P} = \mathbf{A} + \mathbf{E}$, where $\mathbf{A}$ is a low-rank matrix, $\mathbf{E}$ is the sparse error, and $\mathbf{P}$ is the observed data from real-world devices, e.g., cameras, sensors, and other equipments. The rank of $\mathbf{P}$ is not low, in most scenarios, due to the disturbances of $\mathbf{E}$. How can we recover the low-rank structure of the matrix

Parts of this chapter are reproduced from [1] with permission number 3410110407533 @ IEEE.

Y. Deng, *High-Dimensional and Low-Quality Visual Information Processing*,
Springer Theses, DOI 10.1007/978-3-662-44526-6_2

from gross errors? This interesting topic has been discussed in a number of works, e.g., [12–14]. Wright et al. proposed the PCP (a.k.a. RPCA) to minimize the nuclear norm of a matrix by penalizing the ℓ_1 norm of errors [13].

Low-rank representation (LRR) [3] is a robust tool for subspace clustering [15], the desired task of which is to classify the mixed data in their corresponding subspaces/clusters. The general model of LRR can be formulated as $\mathbf{P} = \mathbf{PA} + \mathbf{E}$, where $\mathbf{P}$ is the original mixed data, $\mathbf{A}$ is the affine matrix that reveals the correlations between different pairs of data, and $\mathbf{E}$ is the residual of such a representation. In LRR, the affine matrix $\mathbf{A}$ is assumed to be low rank and $\mathbf{E}$ is regarded as sparse corruptions.

Without the loss of generality, the two problems described above can be formulated as LRSL. In this section, we will explicitly combine non-convex terms into the paradigm of low-rank structure learning and investigate two widely used non-convex terms in this paper, i.e., ℓ_p norm ($0 < p < 1$) and log-sum term. Accordingly, two non-convex models, i.e., ℓ_p-norm heuristic recovery (pHR) and log-sum heuristic recovery (LHR), will be proposed for corrupted matrix learning. Theoretically, we will analyze the relationship between these two models and reveal that the proposed LHR exhibits the same objective of pHR when p infinitely approaches to 0^+. Therefore, LHR owns more powerful sparseness enhancement capabilities than pHR.

For the sake of accurate solutions, the majorization–minimization (MM) algorithm will be applied to solve the non-convex heuristic model. MM algorithm is implemented in an iterative way that it first replaces the non-convex component of the objective with a convex upper bound and then to minimize the constructed surrogate, which exactly makes the non-convex problem fall into the general paradigm of the reweighted schemes. Accordingly, it is possible to solve the non-convex optimization following a sequence of convex optimizations and we will prove that with the MM framework, non-convex models finally converge to a stationary point after successive iterations.

In practical applications, LHR is a powerful tool to solve a number of data analysis tasks in the areas of image and vision analysis, such as light separation and motion analysis.

2.2 Algorithm

2.2.1 Sparse and Low-Rank Structures

In this part, we formulate the LRSL as a semi-definite programming (SDP). With the SDP formulation, it will become apparent that typical LRSL is a kind of general ℓ_1 heuristic optimizations. As stated previously, the basic optimization (P0) is non-convex and generally impossible to solve as its solution usually requires an intractable combinatorial search. In order to make the problem trackable, convex alternatives are widely used in a number of works, e.g., [12, 13]. Among these approaches, one prevalent method tries to replace the rank of a matrix by its convex envelope,

i.e., the nuclear norm, and the ℓ_0 sparsity is penalized via ℓ_1 norm. Accordingly, by convex relaxation, the problem in (2.1) can actually be recast as a semi-definite programming.

$$\begin{aligned} &\min_{(\mathbf{A},\mathbf{E})} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_{\ell_1} \\ &\text{s.t. } \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}), \end{aligned} \tag{2.1}$$

where $\|\mathbf{A}\|_* = \sum_{i=1}^{r} \sigma_i(A)$ is the nuclear norm of the matrix which is defined as the summation of the singular values of $\mathbf{A}$; and $\|\mathbf{E}\|_{\ell_1} = \sum_{ij} |E_{ij}|$ is the ℓ_1 norm of a matrix. Although the objective in (2.1) involves two norms: nuclear norm and ℓ_1 norm, its essence is based on the ℓ_1 heuristic. We will verify this point with the following lemma.

Lemma 1 *For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, its nuclear norm is equivalent to the following optimization:*

$$\|\mathbf{X}\|_* = \left\{ \begin{aligned} &\min_{(\mathbf{Y},\mathbf{Z},\mathbf{X})} \tfrac{1}{2}[\mathrm{tr}(\mathbf{Y}) + \mathrm{tr}(\mathbf{Z})] \\ &\text{s.t.} \begin{bmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \succeq 0, \end{aligned} \right\} \tag{2.2}$$

where $\mathbf{Y} \in \mathbb{R}^{m\times m}$ and $\mathbf{Z} \in \mathbb{R}^{n\times n}$ are both symmetric and positive definite. The operator $\mathrm{tr}(\cdot)$ *means the trace of a matrix, and* $\succeq$ *represents semi-positive definite.*

The proof of Lemma 1 may refer to [10, 16]. According to this lemma, we can replace the nuclear norm in (2.1) and formulate it in the form of

$$\begin{aligned} &\min_{(\mathbf{Y},\mathbf{Z},\mathbf{A},\mathbf{E})} \tfrac{1}{2}[\mathrm{tr}(\mathbf{Y}) + \mathrm{tr}(\mathbf{Z})] + \lambda \|\mathbf{E}\|_{\ell_1} \\ &\text{s.t.} \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \succeq 0 \\ &\quad \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}). \end{aligned} \tag{2.3}$$

From Lemma 1, we know that both $\mathbf{Y}$ and $\mathbf{Z}$ are symmetric and positive definite. Therefore, the trace of $\mathbf{Y}$ and $\mathbf{Z}$ can be expressed as a specific form of ℓ_1 norm, i.e., $\mathrm{tr}(\mathbf{Y}) = \|\mathrm{diag}(\mathbf{Y})\|_{\ell_1}$. $\mathrm{diag}(\mathbf{Y})$ is an operator that only keeps the entries on the diagonal position of $\mathbf{Y}$ in a vector. Therefore, the optimization in (2.3) can be expressed as follows:

$$\min_{\hat{X}\in\hat{D}} \frac{1}{2}(\|\mathrm{diag}(\mathbf{Y})\|_{\ell_1} + \|\mathrm{diag}(\mathbf{Z})\|_{\ell_1}) + \lambda \|\mathbf{E}\|_{\ell_1}, \tag{2.4}$$

where $\hat{X} = \{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$ and

$$\hat{D} = \left\{ (\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}) : \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \succeq 0, (\mathbf{A}, \mathbf{E}) \in C \right\}.$$

$(\mathbf{A}, \mathbf{E}) \in C$ stands for convex constraint.

2.2.2 Non-convex Heuristic Recovery

By Lemma 1, the convex problem with two norms in (2.1) has been successfully converted to an optimization only with ℓ_1 norm, and therefore, it is called ℓ_1-heuristic. ℓ_1 norm is the convex envelope of the concave ℓ_0 norm, but a number of previous research works have indicated the limitation of approximating ℓ_0 sparsity with ℓ_1 norm, e.g., [17, 18]. It is natural to ask, for example, whether might a different alternative not only find a correct solution, but also outperform the performance of ℓ_1 norm? One natural inspiration is to use some non-convex terms lying much closer to the ℓ_0 norm than the convex ℓ_1 norm. However, by using the non-convex heuristic terms, two problems come out inevitably: (1) Which non-convex functionality is ideal and (2) how to efficiently solve the non-convex optimization. In the following two subsections, we will, respectively, address these two problems by introducing the log-sum heuristic recovery and its reweighted solution.

In this section, we will introduce two non-convex terms to enhance the sparsity of model in (2.4). The first one is the widely used ℓ_p norm with $0 < p < 1$. Intuitively, it lies in the scope between the ℓ_0 norm and the ℓ_1 norm. Therefore, it is believed to have a better sparse representation ability than the convex ℓ_1 norm. We define the general concave ℓ_p norm by $f_p(X) = \sum_{ij} |X_{ij}|^p, 0 < p < 1$. Therefore, by taking it into (2.4), the following ℓ_p-norm heuristic recovery (pHR) optimization is obtained.

$$(p\text{HR})\ H_p(\hat{X}) = \min_{\hat{X}\in\hat{D}} \tfrac{1}{2}\left[f_p(\text{diag}(\mathbf{Y})) + f_p(\text{diag}(\mathbf{Z}))\right] + \lambda f_p(\mathbf{E}). \tag{2.5}$$

In the formulation of pHR, obviously, it differs from (2.4) only on the selection of the sparse norm, where the later uses concave ℓ_p norm instead of the typical ℓ_1 norm. Starting from pHR, another non-convex heuristic model with much sparser penalization can be derived. Obviously, $\forall p > 0$, minimizing the above pHR is equivalent to

$$\begin{aligned}\min_{\hat{X}\in\hat{D}} F(\hat{X}) &= \frac{1}{p}\left[H_p(\hat{X}) - \left(\frac{1}{2}m + \frac{1}{2}n + \lambda mn\right)\right] \\ &= \frac{1}{2}\sum_{i=1}^{m}\frac{|Y_{ii}|^p - 1}{p} + \frac{1}{2}\sum_{i=1}^{n}\frac{|Z_{ii}|^p - 1}{p} \\ &\quad + \lambda\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{|E_{ij}|^p - 1}{p}.\end{aligned} \tag{2.6}$$

The optimization in (2.5) is the same as the problem in (2.6) because the multiplied scaler $\frac{1}{p}$ is a positive constant and $\frac{1}{2}m + \frac{1}{2}n + \lambda mn$ is a constant. According to L'Hôspital's rule, we know that $\lim_{p\to 0}\frac{x^p-1}{p} = \frac{\partial_p(x^p-1)}{\partial_p(p)} = \log x$, where $\partial_p(f(p))$ stands for the derivative of $f(p)$ with respect to p. Accordingly, by taking the limit

$\lim_{p\to 0^+} F(X)$ in (2.6), we get the log-sum heuristic recovery (LHR) model $H_L(\hat{X})$:

$$\text{(LHR) } H_L(\hat{X}) = \min_{\hat{X}\in\hat{D}} \tfrac{1}{2}[f_L(\text{diag}(\mathbf{Y})) + f_L(\text{diag}(\mathbf{Z}))] + \lambda f_L(\mathbf{E}). \tag{2.7}$$

For any matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the log-sum term is defined as $f_L(\mathbf{X}) = \sum_{ij} \log(|X_{ij}| + \delta)$, where $\delta > 0$ is a small regularization constant. From (2.5) and (2.7), we know that LHR is a particular case of pHR by taking the limit of p at 0^+. It is known that when $0 < p < 1$, the closer p approaches to zero, the stronger sparse enhancement that ℓ_p-based optimization exhibits. We also comment here that when p equals to zero, the pHR exactly corresponds to the intractable discrete problem in P0. When $p = 0$ and $p \to 0^+$, pHR gives two different objectives. This finding does not deny our basic derivation since when $p = 0$ or $p < 0$, the equivalence from (2.5) to (2.6) does not hold any longer. This is meanwhile the very reason why we denote a "plus" on the superscript of zero in limit $p \to 0^+$. Due to much more powerful sparseness of LHR, we will discuss the formulations of LHR for low-rank optimization in detail in the remainder of this paper.

2.2.3 Solving LHR via Reweighed Approaches

Although we have placed a powerful term to enhance the sparsity, unfortunately, it also causes non-convexity into the objective function. For example, the LHR model is not convex since the log function over $\mathbb{R}_{++} = (\delta, \infty)$ is concave. In most cases, non-convex problem can be extremely hard to solve. Fortunately, the convex upper bound of $f_L(\cdot)$ can be easily found and defined by its first-order Taylor expansion. Therefore, we will introduce the MM algorithm to solve the LHR optimization.

The MM algorithm replaces the hard problem by a sequence of easier ones. It proceeds in an expectation–maximization (EM)-like fashion by repeating two steps of **m**ajorization and **m**inimization in an iterative way. During the *majorization* step, it constructs the convex upper bound of the non-convex objective. In the *minimization* step, it minimizes the upper bound. The first-order Taylor expansion of each component in (2.7) is well defined. Therefore, we can construct the upper bound of LHR and instead to optimize the following problem

$$\begin{aligned}\min_{\hat{X}\in\hat{D}} T(\hat{X}|\hat{\Gamma}) = {} & \frac{1}{2}\text{tr}[(\Gamma_Y + \delta\mathbf{I}_m)^{-1}\mathbf{Y}] + \frac{1}{2}\text{tr}[(\Gamma_z + \delta\mathbf{I}_n)^{-1}\mathbf{Z}] \\ & + \lambda \sum_{ij} (\Gamma_{E_{ij}} + \delta)^{-1} E_{ij} + \text{const.}\end{aligned} \tag{2.8}$$

In (2.8), set $\hat{X} = \{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$ contains all the variables to be optimized and set $\hat{\Gamma} = \{\Gamma_Y, \Gamma_Z, \Gamma_E\}$ contains all the parameter matrices. The parameter matrices define the points at which the concave function is linearized via Taylor expansion. At the end of (2.8), const stands for the constants that are irrelative to $\{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$.

In some previous works of MM algorithms [17, 19, 20], they denote the parameter $\hat{\Gamma}$ in tth iteration with the optimal value of $\hat{X}$ of the last iteration, i.e., $\hat{\Gamma} = \hat{X}^{t^*}$.

To numerically solve the LHR optimization, we remove the constants that are irrelative to $\mathbf{Y}$, $\mathbf{Z}$, and $\mathbf{E}$ in $T(\hat{X}|\hat{\Gamma})$ and get the new convex objective

$$\min \frac{1}{2}[\mathrm{tr}(\mathbf{W}_Y^2\mathbf{Y}) + \mathrm{tr}(\mathbf{W}_Z^2\mathbf{Z})] + \lambda \sum_{ij} (W_E)_{ij} E_{ij}$$

where $\mathbf{W}_{Y(Z)} = (\Gamma_{Y(Z)} + \delta \mathbf{I}_{\mathbf{m(n)}})^{-\mathbf{1/2}}$ and $(W_E)_{ij} = (E_{ij} + \delta)^{-1}, \forall ij$. It is worth noting that $\mathrm{tr}(\mathbf{W}_Y^2\mathbf{Y}) = \mathrm{tr}(\mathbf{W}_Y\mathbf{Y}\mathbf{W}_\mathbf{Y})$. Besides, since both $\mathbf{W}_Y$ and $\mathbf{W}_Z$ are positive definite, the first constraint in (2.7) is equivalent to

$$\begin{bmatrix} \mathbf{W}_Y & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_Z \end{bmatrix} \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{W}_Y & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_Z \end{bmatrix} \succeq \mathbf{0}$$

Therefore, after convex relaxation, the optimization in (2.7) is now subjected to

$$\begin{aligned} \min \ & \tfrac{1}{2}[\mathrm{tr}(\mathbf{W}_Y\mathbf{Y}\mathbf{W}_Y) + \mathrm{tr}(\mathbf{W}_Z\mathbf{Z}\mathbf{W}_Z)] + \lambda\|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} \ & \begin{bmatrix} \mathbf{W}_Y\mathbf{Y}\mathbf{W}_Y & \mathbf{W}_Y\mathbf{A}\mathbf{W}_Z \\ (\mathbf{W}_Y\mathbf{A}\mathbf{W}_Z)^T & \mathbf{W}_Z\mathbf{Z}\mathbf{W}_Z \end{bmatrix} \succeq \mathbf{0} \\ & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}) \end{aligned} \tag{2.9}$$

Here, we apply Lemma 1 to (2.9) once again and rewrite the optimization in (2.9) in the form of the summation of the nuclear norm and ℓ_1 norm,

$$\begin{aligned} \min_{(\mathbf{A},\mathbf{E})} \ & .\ \|\mathbf{W}_Y\mathbf{A}\mathbf{W}_Z\|_* + \lambda\|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} \ & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}) \end{aligned} \tag{2.10}$$

In (2.10), the operator $\odot$ in the error term denotes the component-wise product of two variables, i.e., for $\mathbf{W}_E$ and $\mathbf{E}$: $(\mathbf{W}_E \odot \mathbf{E})_{ij} = (W_E)_{ij}E_{ij}$. According to [16], we know that $\mathbf{Y}^* = \mathbf{U}\Sigma\mathbf{U}^T$ and $\mathbf{Z}^* = \mathbf{V}\Sigma\mathbf{V}^T$, if we do singular value decomposition for $\mathbf{A}^* = \mathbf{U}\Sigma\mathbf{V}^T$. Accordingly, the weight matrix $\mathbf{W}_Y = (\mathbf{U}\Sigma\mathbf{U}^T + \delta\mathbf{I}_m)^{-1/2}$ and matrix $\mathbf{W}_Z = (\mathbf{V}\Sigma\mathbf{V}^T + \delta\mathbf{I}_n)^{-1/2}$.[1] We should comment here that Eq. 2.10 is also applied to solve pHR problem. It just uses different weighting matrices $\mathbf{W}_Y = \mathrm{diag}((\mathbf{U}\Sigma\mathbf{U}^T + \delta\mathbf{I}_m))^{(p-1)/2}$, $\mathbf{W}_Z = \mathrm{diag}((\mathbf{V}\Sigma\mathbf{V}^T + \delta\mathbf{I}_n))^{(p-1)/2}$, and $\mathbf{W}_E = [(E_{ij} + \delta)^{(p-1)}]$.

Here, based on MM algorithm, we have converted the non-convex LHR optimization to be a sequence of convex reweighted problems. Besides, the objective in (2.10) is convex with a summation of a nuclear norm and a ℓ_1 norm and can be solved by convex optimization. In the following section, we first extend some theoretic discussions of the LHR model.

[1] In cases, the weighting matrices may cause complex numbers due to the inverse operation. In such condition, we use the approximating matrices $\mathbf{W}_Y = \mathbf{U}(\Sigma + \delta\mathbf{I}_m)^{-1/2}\mathbf{U}^T$ and $\mathbf{W}_Z = \mathbf{V}(\Sigma + \delta\mathbf{I}_m)^{-1/2}\mathbf{V}^T$ in LHR.

2.3 Theoretical Justifications

In this part, for the sake of simplicity, we define the objective in (2.7) as $H(\hat{X})$ and the surrogate function in (2.8) is defined as $T(\hat{X}|\hat{\Gamma})$. $\hat{X}$ is a set containing all the variables, and set $\hat{\Gamma}$ records the parameter matrices. The convergence property of general MM algorithm was separately distributed on some early mathematical journals [21, 22] which are bit obscure and were not generally read by researchers in the community of computer science. Besides, previous works on MM convergence are almost on the variable selection models. In this paper, we specify it to our LHR model and try to explain it in a plain way. Before discussing the convergence property of LHR, we will first provide two lemmas.

Lemma 2 *If set* $\hat{\Gamma}^t := \hat{X}^t$*, MM algorithm could monotonically decrease the non-convex objective function* $H(\hat{X})$*, i.e.,* $H(\hat{X}^{t+1}) \leq H(\hat{X}^t)$.

Proof In order to prove the monotonically decrease property, we can instead to prove

$$H(\hat{X}^{t+1}) \leq T(\hat{X}^{t+1}|\hat{\Gamma}^t) \leq T(\hat{X}^t|\hat{\Gamma}^t) = H(\hat{X}^t). \tag{2.11}$$

We prove (2.11) by the following three steps:

i. The first inequality follows from the argument that $T(\hat{X}|\hat{\Gamma})$ is the upper bound of $H(\hat{X})$.
ii. The second inequality holds since the MM algorithm computes $\hat{X}^{t+1} = \arg\min_{\hat{X}} T(\hat{X}|\hat{\Gamma}^t)$. The function $T(\cdot)$ is convex; therefore, $\hat{X}^{t+1}$ is the unique global minimum. This property guarantees that $T(\hat{X}^{t+1}|\hat{\Gamma}^{t+1}) < T(\cdot|\hat{\Gamma}^t)$ with any $\hat{X} \neq \hat{X}^{t+1}$ and $T(\hat{X}^{t+1}|\hat{\Gamma}^{t+1}) = T(\cdot|\hat{\Gamma}^t)$ if and only if with $\hat{X} = \hat{X}^{t+1}$.
iii. The last equality can be easily verified by expanding $T(\hat{X}^t|\hat{\Gamma}^t)$ and making some simple algebra. The transformation is straightforward, and we omit it here.

Lemma 3 *Let* $\hat{X} = \{\hat{X}^0, \hat{X}^1, \ldots, \hat{X}^t, \ldots\}$ *be a sequence generated by MM framework, after successive iterations, such a sequence converge to the same limit point.*

Proof We give a proof by contradiction. We assume that sequence $\hat{X}$ diverge which means that $\lim_{t\to\infty} \|\hat{X}^{t+1} - \hat{X}^t\|_F \neq 0$. According to the discussions previously, we know that there exists a convergent subsequence $\hat{X}^{t_k}$ converge to ϕ, i.e., $\lim_{k\to\infty} \hat{X}^{t_k} = \phi$, and meanwhile, we can construct another convergent subsequence $\hat{X}^{t_k+1}$ that $\lim_{k\to\infty} \hat{X}^{t_k+1} = \varphi$. We assume that $\phi \neq \varphi$. Since the convex upper bound $T(\cdot|\hat{\Gamma})$ is continuous, we get $\lim_{k\to\infty} T(\hat{X}^{t_k+1}|\hat{\Gamma}^{t_k}) = T(\underbrace{\lim_{k\to\infty} \hat{X}^{t_k+1}}_{\varphi}|\hat{\Gamma}^{t_k}) <$ $T(\underbrace{\lim_{k\to\infty} \hat{X}^{t_k}}_{\phi}|\hat{\Gamma}^{t_k}) = \lim_{k\to\infty} T(\hat{X}^{t_k}|\hat{\Gamma}^{t_k})$. The *strict less than operator* "$<$" holds

because $\varphi \neq \phi$. See (ii) in the proof of Lemma 2 for details. Therefore, it is straightforward to get the following inequalities that $\lim_{k\to\infty} H(\hat{X}^{t_k+1}) \leq \lim_{k\to\infty} T(\hat{X}^{t_k+1}|\hat{\Gamma}^{t_k}) < \lim_{k\to\infty} T(\hat{X}^{t_k}|\hat{\Gamma}^{t_k}) = \lim_{k\to\infty} H(\hat{X}^{t_k})$. Accordingly,

$$\lim_{k\to\infty} H(\hat{X}^{t_k+1}) < \lim_{k\to\infty} H(\hat{X}^{t_k}) \tag{2.12}$$

Besides, it is obvious that the function of $H(\cdot)$ in (2.7) is bounded below, i.e., $H(\hat{X}) > (mn + m + n)\log\delta$. Moreover, as proved in Lemma 2, $H(\hat{X})$ is monotonically decreasing, which guarantees that $\lim_{t\to\infty} H(\hat{X}^t)$ exists, i.e.,

$$\lim_{k\to\infty} H(\hat{X}^{t_k}) = \lim_{t\to\infty} H(\hat{X}^t) = \lim_{t\to\infty} H(\hat{X}^{t+1}) = \lim_{k\to\infty} H(\hat{X}^{t_k+1}) \tag{2.13}$$

Obviously, (2.13) contradicts to (2.12). Therefore, $\phi = \varphi$, and we get the conclusion that $\lim_{t\to\infty} \|\hat{X}^{t+1} - \hat{X}^t\|_F = 0$.

Based on the two lemmas proved previously, we can give the convergence theorem of the proposed LHR model.

Theorem 1 *With the MM framework, LHR model finally converges to a stationary point.*

Proof As stated in Lemma 3, the sequences generated by MM algorithm converge to a limitation, and here, we will first prove that the convergence is a *fixed point*. We define the mapping from $\hat{X}^k$ to $\hat{X}^{k+1}$ as $M(\cdot)$, and it is straightforward to get, $\lim_{t\to\infty} \hat{X}^t = \lim_{t\to\infty} \hat{X}^{t+1} = \lim_{t\to\infty} M(\hat{X}^t)$, which implies that $\lim_{t\to\infty} \hat{X}^t = \phi$ is a fixed point. In the MM algorithm, when constructing the upper bound, we use the first-order Taylor expansion. It is well known that the convex surrogate $T(\hat{X}|\hat{\Gamma})$ is tangent to $H(\hat{X})$ at $\hat{X}$ by the property of Taylor expansion. Accordingly, the gradient vector of $T(\hat{X}|\hat{\Gamma})$ and $H(\hat{X})$ is equal when evaluating at $\hat{X}$. Besides, we know that at the fixed point, $\mathbf{0} \in \nabla_{\hat{X}=\phi} T(\hat{X}|\hat{\Gamma})$, and because it is tangent to $H(\hat{X})$, we can directly get that $\mathbf{0} \in \nabla_{\hat{X}=\phi} H(\hat{X})$, which proves that the convergent fixed point ϕ is also a *stationary point* of $H(\cdot)$.

In this part, we have shown that with the MM algorithm, LHR model could converge to a stationary point. However, it is not possible to claim that the converged point is the global minimum since the objective function of LHR is not convex. Fortunately, with a good starting point, we can always find a desirable solution by iterative approaches. In this chapter, the solution of ℓ_1 heuristic model was used as a starting point and it could always lead to a satisfactory result.

2.4 LRSL for Visual Information Processing

2.4.1 LHR for Low-Rank Matrix Recovery

In this part, we first apply the LHR model to recover a low-rank matrix from corruption and its performance is compared with the widely used principal component pursuit (PCP).

Based on the LHR derivations, the corrupted low-rank matrix recovery problem can be formulated as a reweighted problem:

$$\begin{aligned} &\min_{(\mathbf{A},\mathbf{E})} . \|\mathbf{W}_Y\mathbf{A}\mathbf{W}_Z\|_* + \lambda\|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ &\text{s.t. } \mathbf{P} = \mathbf{A} + \mathbf{E} \end{aligned} \tag{2.14}$$

Due to the reweighted weights that are placed in the nuclear norm, it is impossible to directly get the closed-form solution of the nuclear norm minimization. Therefore, inspired by the work [3], we introduce another variable $\mathbf{J}$ to (2.14) by adding another equality constraint and to solve

$$\begin{aligned} &\min . \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ &\text{s.t. } \mathbf{h}_1 = \mathbf{P} - \mathbf{A} - \mathbf{E} = \mathbf{0} \\ &\quad\ \mathbf{h}_2 = \mathbf{J} - \mathbf{W}_Y\mathbf{A}\mathbf{W}_Z = \mathbf{0} \end{aligned} \tag{2.15}$$

Using the augmented Lagrangian multiplier (ALM) method [23], it is computationally expedient to relax the equality in (2.15) and instead solve

$$L = \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} + \langle \mathbf{C}_1, \mathbf{h}_1 \rangle + \langle \mathbf{C}_2, \mathbf{h}_2 \rangle + \frac{\mu}{2}\left(\|\mathbf{h}_1\|_F^2 + \|\mathbf{h}_2\|_F^2\right) \tag{2.16}$$

where $\langle , \rangle$ is an inner product and $\mathbf{C}_1$ and $\mathbf{C}_2$ are the Lagrange multipliers, which can be updated via dual ascending method. Equation (2.16) contains three variables, i.e., $\mathbf{J}$, $\mathbf{E}$, and $\mathbf{A}$. The joint optimization above can be minimized by four steps as $\mathbf{E}$-minimization, $\mathbf{J}$-minimization, $\mathbf{A}$-minimization, and dual ascending. The detailed derivations are similar to the previous works in [23], and we omit them here. The whole framework to solve the LHR model for LRMR via reweighted schemes is given in Algorithm 1. [2] In lines 6 and 7 of the algorithm, $s_\alpha(\cdot)$ and $d_\alpha(\cdot)$ are defined as signal shrinkage operator and matrix shrinkage operator, respectively [23].

We have explained how to recover a low-rank matrix via LHR in preceding sections. In this section, we will conduct some experiments to test its performances with the comparisons to robust PCP from numerical simulations. For an equivalent comparison, we adopted the same data generating method in [12] that all the algorithms are performed on the squared matrices and the ground truth low-rank matrix (rank r) with $m \times n$ entries, denoted as $\mathbf{A}^*$, is generated by independent random

[2] The optimization for pHR is very similar by changing the weight matrices.

Algorithm 1: Optimization strategy of LHR for corrupted matrix recovery

Input : Corrupted matrix P and parameter λ

Initialization : $t := 1$, $E_{ij}^0 := 1, \forall i, j$. $\mathbf{W}_{Y(Z)}^{(1)} = \mathbf{I}_{m(n)}$.

repeat

  Update the weighting matrices $\mathbf{W}_E^{(t)}$, $\mathbf{W}_Y^{(t)}$ and $\mathbf{W}_Z^{(t)}$ according to current estimation of $\mathbf{A}^{(t)}$ and $\mathbf{E}^{(t)}$;

  Reset $C_0 > 0$; $\mu_0 > 0$; $\rho > 1$; $k = 1$; $\mathbf{A}^0 = \mathbf{E}^0 = \mathbf{0}$;

  while *not converged* **do**

    `Variables updating.` $E_{ij}^k = s_{\lambda\mu^{-1}\left|(W_E^{(t)})_{ij}\right|}(P - A^{k-1} - \mu^{-1}C_1^k)_{ij}, \forall ij$;

    $\mathbf{J}^k = d_{\mu^{-1}}(\mathbf{W}_Y^{(t)}\mathbf{A}^{\mathbf{k-1}}\mathbf{W}_Z^{(t)} + \mu^{-1}\mathbf{C}_2^k)$;

    $\mathbf{A}^{\mathbf{k}} = \mathbf{A}^{k-1} + \gamma[-\mathbf{W}_Y^{(t)}(\mathbf{h}_1^k + \mu^{-1}\mathbf{C}_2^k)\mathbf{W}_Z^{(t)} + (\mathbf{h}_2^k + \mu^{-1}\mathbf{C}_1^k)]$;

    `Dual ascending.` $\mathbf{C}_1^k = \mathbf{C}_1^{k-1} + \mu_k\mathbf{h}_1^k$;

    $\mathbf{C}_2^k = \mathbf{C}_2^{k-1} + \mu_k\mathbf{h}_2^k$;

    $k := k + 1$, $\mu_{k+1} = \rho\mu_k$;

  end

  $(\mathbf{A}^{(t)}, \mathbf{E}^{(t)}) = (\mathbf{A}_k, \mathbf{E}_k)$;

  $t := t + 1$;

until *convergence*;

Output : $(\mathbf{A}^{(t)}, \mathbf{E}^{(t)})$.

orthogonal model [12]; the sparse error $\mathbf{E}^*$ is generated via uniformly sampling the matrix, and the error values are randomly generated in the range $[-100,100]$. The corrupted matrix is generated by $\mathbf{P} = \mathbf{A}^* + \mathbf{E}^*$, where $\mathbf{A}^*$ and $\mathbf{E}^*$ are the ground truth. For simplicity, we denote the rank rate as $\eta = \frac{\text{rank}(\mathbf{A}^*)}{\max\{m,n\}}$ and the error rate as $\xi = \frac{\|\mathbf{E}\|_{\ell_0}}{m \times n}$.

Previous work [12] indicated that PCP method could exactly recover a low-rank matrix from corruptions within the region of $\eta + \xi < 0.35$. Here, in order to highlight the effectiveness of our LHR model, we directly consider much difficult tasks that we set $\eta + \xi = 0.5$. We compare the PCP ($p = 1$) model with the proposed pHR (with $p = 1/3$ and $p = 2/3$) and the LHR (can be regarded as $p \to 0^+$). Each experiment is repeated for ten times, and the mean values and their standard deviations (std) are tabulated in Table 2.1. In the table, $\frac{\|\mathbf{A}-\mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F}$ denotes the recovery accuracy, $rank$ denotes the rank of the recovered matrix $\mathbf{A}$, $\|\mathbf{E}\|_{\ell_0}$ is the card of the recovered errors, and $time$ records the computational costs (in seconds).

From the results, obviously, compared with PCP, LHR model could exactly recover the matrix from higher ranks and denser errors. pHR model could correctly recover the matrix in most cases, but the recover accuracy is a bit lower than LHR. We also report the processing time in Table 2.1. The computer to implement these experiments is equipped with a 2.3 GHZ CPU processor and a 4-GB RAM.

The basic optimization involves two terms, i.e., low-rank matrix and sparse error. In this part, we will vary these two variables to test the feasible boundary of PCP, pHR, and LHR, respectively. The experiments are conducted on the 400×400 matrices with sparse errors uniformly distributed in $[-100, 100]$. In the feasible region verification,

Table 2.1 Evaluations of low-rank matrix recovery of robust PCA and non-convex heuristic recovery (mean)

		rank $(\mathbf{A}^*) = 0.4\,\mathrm{m}$ $\|\|\mathbf{E}^*\|\|_{\ell_0} = 0.1\,\mathrm{m}^2$				rank $(\mathbf{A}^*) = 0.1\,\mathrm{m}$ $\|\|\mathbf{E}^*\|\|_{\ell_0} = 0.4\,\mathrm{m}^2$			
m	Methods	$\frac{\|\mathbf{A}-\mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F}$	rank $(\mathbf{A})$	$\|\|\mathbf{E}\|\|_{\ell_0}$	Time (s)	$\frac{\|\mathbf{A}-\mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F}$	rank $(\mathbf{A})$	$\|\|\mathbf{E}\|\|_{\ell_0}$	Time (s)
200	PCP	$4.6e^{-1}$	102	21,132	5.9	$1.2e^{-1}$	107	23,098	7.4
	pHR$^{2/3}$	$3.7e^{-2}$	88	4,378	16.4	$9.3e^{-3}$	20	16,011	16.3
	pHR$^{1/3}$	$1.8e^{-2}$	83	4,113	13.1	$3.6e^{-3}$	20	16,000	13.4
	LHR	$8.1e^{-4}$	80	4,000	12.7	$1.3e^{-3}$	20	16,031	14.1
400	PCP	$4.5e^{-1}$	205	82,149	27.4	$6.4e^{-1}$	217	89,370	33.2
	pHR$^{2/3}$	$2.3e^{-2}$	193	15,782	73.8	$5.0e^{-3}$	71	64,000	63.2
	pHR$^{1/3}$	$1.2e^{-2}$	160	15,873	64.2	$4.0e^{-4}$	41	64,000	63.2
	LHR	$2.3e^{-3}$	160	16,038	53.4	$1.7e^{-4}$	40	64,000	54.3
800	PCP	$4.7e^{-1}$	435	336,188	36.2	$9.1e^{-2}$	348	355,878	50.1
	pHR$^{2/3}$	$2.3e^{-2}$	361	63,901	103.6	$6.2e^{-3}$	80	257,762	129.2
	pHR$^{1/3}$	$8.7e^{-3}$	320	63,962	96.2	$5.3e^{-3}$	80	256,097	119.2
	LHR	$1.7e^{-3}$	320	63,999	89.3	$4.1e^{-3}$	80	255,746	107.6

when the recovery accuracy is larger than 1 % (i.e., $\frac{\|A-A^*\|_F}{\|A^*\|_F} > 0.01$), it is believed that the algorithm diverges. The two rates η and ξ are varied from zero to one with the step of 0.025. On each test point, all the algorithms are repeated for 10 times. If the median recovery accuracy is less than 1 %, the point is regarded as the feasible point. The feasible regions of these two algorithms are shown in Fig. 2.1a. From Fig. 2.1a, the feasible region of LHR is much larger than the region of PCP.

We get the same conclusion as made in [12] that the feasible boundary of PCP roughly fits the curve that $\eta^{PCP} + \xi^{PCP} = 0.35$. The boundary of LHR is around the curve that $\eta^{LHR} + \xi^{LHR} = 0.575$. Moreover, on the two sides of the red curve in Fig. 2.1a, the boundary equation can be even extended to $\eta^{LHR} + \rho^{LHR} = 0.6$. Although the performance of pHR is not as good as LHR, it still greatly outperforms the performance of PCP. When $p = 1/3$ and $p = 2/3$, the boundary equations are subjected to $\eta^{pHR} + \rho^{pHR} = 0.52$ and $\eta^{pHR} + \rho^{pHR} = 0.48$, respectively. These

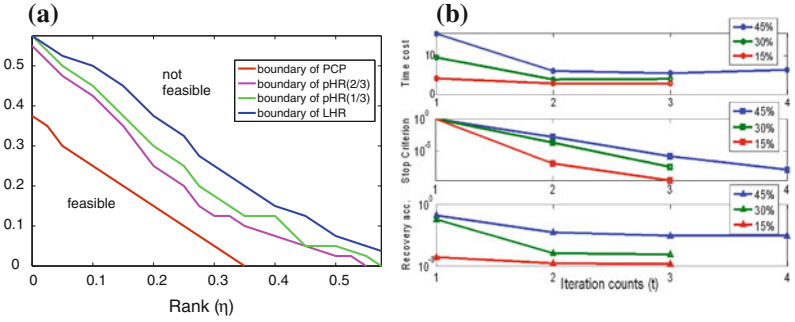

Fig. 2.1 **a** Feasible region and **b** the convergence verifications

improvements are reasonable since pHR and LHR use the functionalities that are much closer to the ℓ_0 norm. Accordingly, the proposed non-convex heuristic method covers a larger feasible region. From this test, it is apparent that the proposed LHR algorithm covers the largest area of the feasible region, which implies that LHR could handle more difficult tasks that robust PCA fails to do. Here, we will conduct two practical applications to verify the effectiveness of PCP and LHR on real-world data.

In the first application, we consider the shadow and specularities removal from faces. Following the framework suggested in [12], we stack the faces of the same subject under different lighting conditions as the columns in a matrix $\mathbf{P}$. The experiments are conducted on extended Yale-B dataset where each face is with the resolutions of 192×168. Then, the corrupted matrix $\mathbf{P}$ is recovered by PCP and LHR, respectively. After recovery, the shadows, specularities, and other reflectance are removed in the error matrix ($\mathbf{E}$), and the clean faces are accumulated in the low-rank matrix ($\mathbf{A}$).

The experimental results are provided in Fig. 2.2, where in each subfigure, from left to right are original faces in Yale-B, faces recovered by PCP, faces recovered by pHR ($p = 1/3$),[3] and faces recovered by LHR, respectively. It is greatly recommended to enlarge the faces in Fig. 2.2 to view the details. In Fig. 2.2a, when there

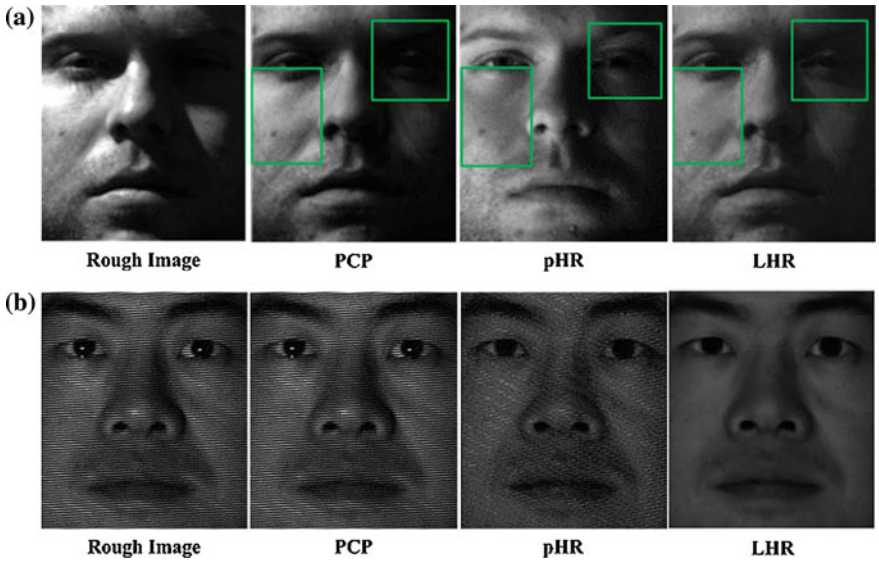

Fig. 2.2 Shadow and specularities removal from faces (best viewed on screen). **a** Dense shadow, **b** shadow texture

[3] We only report the result of $p = 1/3$ here since in the previous numerical simulation, pHR ($p = 1/3$) achieves higher recovery accuracy than pHR with $P = 2/3$.

exist dense shadows on the face image, the effectiveness of LHR becomes apparent to remove the dense shadows distributed on the left face. However, in Fig. 2.2a, there is no significant differences between the two non-convex models. Both of them achieve sound result. However, the dense texture removal ability is especially highlighted in Fig. 2.2b, where there are significant visual contrasts between the faces recovered by PCP, pHR, and LHR. The face recovered by LHR is much clean.

The background modeling can also be categorized as a low-rank matrix recovery problem, where the backgrounds correspond to the low-rank matrix $\mathbf{A}$ and the foregrounds are removed in the error matrix $\mathbf{E}$. We use the videos and ground truth in [24] for quantitative evaluations. Three videos used in this experiment are listed in Fig. 2.3. In each subfigure, from left to right are original video frame, ground truth of foregrounds and solutions of LHR, pHR, and PCP.

For the sake of computational efficiency, we normalize each image to the resolutions of 120×160 and all the frames are converted to gray-scaler. The benchmark videos used here contain too many frames which lead to a large matrix. It is theoretically feasible to use the two methods for any large matrix recovery. Unfortunately, for practical implementation, large matrices are always beyond the memory limitation of MATLAB. Therefore, for each video, we uniformly divide the large matrix to be submatrices, which has less than 200 columns.

The segmented foregrounds and the ground truth are shown in Fig. 2.3. From the results, we know that LHR could remove much denser errors from the corrupted matrix rather than PCP. Such claim is verified from three sequences in Fig. 2.3 that LHR makes much complete object recovery from the video. Besides, in Fig. 2.3c, it is also apparent that LHR only keeps dense errors in the sparse error term. In the seam sequences, there are obvious illumination changes in different frames. PCP is sensitive to these small variations and thus makes much more small isolated

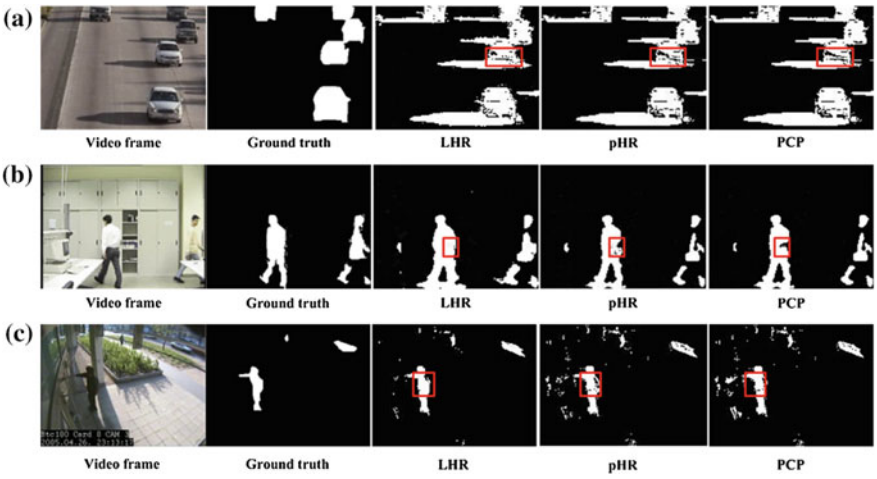

Fig. 2.3 Benchmark videos for background modeling. In each subfigure, from left to right are original video frames, foreground ground truth, LHR result, and PCP result, respectively. **a** HW (439 frames). **b** Lab (886 frames). **c** Seam (459 frames)

Table 2.2 Quantitative evaluation of PCP and non-convex heuristic recovery for video surveillance

Data	False-negative rate %				False-positive rate %				Time (m)		
	MoG	PCP	pHR	LHR	MoG	PCP	pHR	LHR	PCP	pHR	LHR
HW	22.2	18.7	16.2	14.3	8.8	7.8	8.2	8.4	13.2	24.7	23.5
Lab.	15.1	10.1	9.4	8.3	6.7	6.4	6.4	6.1	25.4	45.3	43.7
Seam	23.5	11.3	10.1	9.2	9.7	6.1	6.5	6.3	11.4	23.2	19.9

noise parts in the foreground. On the other hand, LHR is much robust to these local variations and only keeps dense corruptions in the sparse term.

Although there are many advanced techniques for video background modeling, it is not the main concern of this work. Therefore, without the loss of generality, we use the mixture of Gaussian (MoG) as the comparison baseline. In MoG, five Gaussian components are used to model each pixel in the image. For evaluation, both the false-negative rate (FNR) and false-positive rate (FPR) are calculated in the sense of foreground detection. These two scores exactly correspond to the type I and type II errors in machine learning. FNR indicates the ability of the method to correctly recover the foreground, and FPR represents the potential of a method on distinguishing the background. these two rates are judged by the criterion that the less the better. The experimental results are tabulated in Table 2.2.

From the results, PCP and LHR greatly outperform the performance of MoG. Moreover, LHR has lower FNRs than PCP and pHR, which implies that LHR could better detect the foreground than them. However, on the video highway and seam, the FPR score of LHR is a little worse than that of PCP and pHR. One possible reason may ascribe to that there are too many moving shadows in these two videos, where both the objects and shadows are regarded as errors. In the ground truth frames, the shadows are regarded as background. LHR could recover much denser errors from a low-rank matrix and thus causes a relative low FNR score.

2.4.2 LHR for Low-Rank Representation

In this part, LHR will be applied to the task of LRR [3] by formulating the constraint as $\mathbf{P} = \mathbf{PA} + \mathbf{E}$, where the correlation affine matrix $\mathbf{A}$ is low rank and the noises in $\mathbf{E}$ are sparse. In the remaining parts of this section, we will first show how to use the joint optimization strategy to solve the LRR problem by LHR model. Then, two practical applications on motion segmentation and stock clustering will be presented and discussed.

When applying LHR to LRR, we should solve a sequence of convex optimizations in the form

$$\begin{aligned} &\min . \ \|\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ &\text{s.t. } \mathbf{P} = \mathbf{PA} + \mathbf{E} \end{aligned} \tag{2.17}$$

To make the nuclear norm trackable, we add equality and try to solve

$$\begin{aligned} &\min . \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ &\text{s.t. } \mathbf{b}_1 = \mathbf{P} - \mathbf{PA} - \mathbf{E} = \mathbf{0} \\ &\quad \mathbf{b}_2 = \mathbf{J} - \mathbf{W}_Y \mathbf{A} \mathbf{W}_Z = \mathbf{0} \end{aligned} \tag{2.18}$$

Using the ADM strategy and following the similar derivations introduced in Sect. 2.4.1, we can solve the optimization in (2.18) and we directly provide the update rules for each variable in Algorithm 2.

Algorithm 2: Update rule for the variables in (2.18)

1 $E_{ij}^k = s_{\lambda\mu^{-1}\left|(W_E^{(t)})_{ij}\right|}(P - PA^{k-1} - \mu^{-1}C_1^k)_{ij}, \forall ij;$

2 $\mathbf{J}^k = d_{\mu^{-1}}(\mathbf{W}_Y^{(t)}\mathbf{A}^{\mathbf{k-1}}\mathbf{W}_Z^{(t)} + \mu^{-1}\mathbf{C}_2^k);$

3 $\mathbf{A}^{\mathbf{k}} = \mathbf{A}^{k-1} + \gamma[\mathbf{W}_Y^{(t)}(\mathbf{b}_1^k + \mu^{-1}\mathbf{C}_2^k)\mathbf{W}_Z^{(t)} + \mathbf{P}^T(\mathbf{b}_2^k + \mu^{-1}\mathbf{C}_1^k)];$

4 `Dual ascending.` $\mathbf{C}_1^k = \mathbf{C}_1^{k-1} + \mu_k \mathbf{b}_1^k;$

5 $\mathbf{C}_2^k = \mathbf{C}_2^{k-1} + \mu_k \mathbf{b}_2^k;$

To show LHR ideally represents low-rank structures from data, experiments on subspace clustering are conducted on two datasets.

In this part, we apply LHR to the task of motion segmentation in Hopkins155 dataset [15]. Hopkins155 database is a benchmark platform to evaluate general subspace clustering algorithms, which contains 156 video sequences, and each of them has been summarized to be a matrix recoding 39–50 data vectors. The primary task of subspace clustering is to categorize each motion to its corresponding subspace, where each video corresponds to a sole clustering task and it leads to 156 clustering tasks in total.

For comparisons, we will compare LHR with LRR as well as other benchmark algorithms for subspace clustering. The comparisons include random sample consensus (RANSAC) [25], generalized principal component analysis (GPCA) [26], local subspace affinity (LSA), locally linear manifold clustering (LLMC), and sparse subspace clustering (SSC). RANSAC is a statistic method which clusters data by iteratively distinguishing the data by inliers and outliers. GPCA presents an algebraic method to cluster the mixed data by the normal vectors of the data points. Manifold-based algorithms, e.g., LSA and LLMC, assume that one point and its neighbors span as a linear subspace and they are clustered via spectral embedding. SSC assumes that the affine matrix between data is sparse and it segments the data via normalized cut [27].

In order to provide a thorough comparison with LRR, we strictly follow the steps and the default parameter settings suggested in [3]. For LHR model, we choose parameter $\lambda = 0.4$. In the experiments of LRR for motion segmentation, some post-processing is performed on the learned low-rank structure to seek for the best clustering accuracy. For example, in LRR, after getting the representation matrix $\mathbf{A}$,

Table 2.3 Motion segmentation errors (mean) of several algorithms on the Hopkins155 motion segmentation database

Category	Method	TWO	THREE	ALL
Algebraic	GPCA	11.2	27.7	14.2
Statistic	RANSAC	8.9	24.1	12.5
Manifold	LSA	8.7	21.4	11.6
	LLMC	8.1	20.8	10.9
	SSC	5.4	15.3	7.6
Sparse	LRR	4.7	15.1	6.9
	pHR	4.2	14.4	6.1
	LHR	3.1	13.9	5.6

an extra PCP processing is implemented on **A** to enhance the low rankness and such post-processing definitely increases SC accuracy. However, the main contribution of this work only focuses LHR model on low-rank structure learning while not on the single task of subspace clustering. Therefore, we exclude all the post-processing steps to emphasize the effectiveness of the LRSL model itself.

Hopkins155 contains two subspace conditions in a video sequence, i.e., with two motions or three motions, and thus, we report the segmenting errors for two subspaces (TWO), for three subspaces (THREE), and for both conditions (ALL) in Table 2.3. From the results, we know that sparse-based methods generally outperform other algorithms for motion segmentation. Among three sparse methods, LHR gains the best clustering accuracy. However, the accuracy only has slight improvements on LRR. As indicated in [3], motion data only contain small corruptions and LRR could already achieve promising performance with the accuracy higher than 90 %. With some post-processing implementations, the accuracy can even be further improved. Therefore, in order to highlight the effectiveness of LHR on LRR with corrupted data, some more complicated problems will be considered.

In practical world, one of the most difficult data structures to be analyzed is the stock price which can be greatly affected by company news, rumors, and global economic atmosphere. Therefore, data mining approaches of financial signals have been proven to be very difficult. Here, we will discuss how to use the LRR and LHR models to the interesting, albeit not very lucrative, task of stock clustering based on their industrial categories. In many stock exchange centers around the world, stocks are always divided into different industrial categories. For example, on the New York Stock Exchange Center, IBM and J.P.Morgan are, respectively, categorized into the *computer-based system* category and *money center banks* category. It is generally assumed that stocks in the same category always have similar market performance. This basic assumption is widely used by many hedge funds for statistic arbitrage. In this paper, we consider that stocks in the same industrial category span as a subspace, and therefore, the goal of stock clustering, a.k.a. stock categorization, is to identify a stock's industrial label by its historical prices.

Table 2.4 Clustering errors of the stocks in ten categories from New York and Hong Kong markets

Markets	GPCA	RANSAC	LSA	LLMC
New York	60.1	59.3	51.7	54.3
Hong Kong	57.3	55.8	54.7	53.7
Markets	SSC	LRR	pHR	LHR
New York	48.6	44.3	39.3	36.2
Hong Kong	49.1	47.2	42.7	38.3

The experiments are conducted on stocks from two global stock exchange markets in New York and Hong Kong. In each market, we choose 10 industrial categories which have the largest market capitalizations. The categories divided by the exchange centers are used as the ground truth label. In each category, we only choose the stocks whose market capitalizations are within the top 10 ranks in one category. The stock prices on New York market are obtained from [28], and the stock prices in Hong Kong market are obtained from [29]. Unfortunately, some historical prices for stocks in [28] are not provided.[4] Therefore, for the US market, we accumulated 76 stocks divided into 10 classes and each class contains 7–9 stocks; for Hong Kong market, we obtain 96 stocks spanning 10 classes. For classification, the weekly closed prices, from January 07, 2008, to October 31, 2011, including 200 weeks, are used because financial experts always look at weekly close prices to judge the long-term trend of a stock.

As stated previously, the stock prices may have sharp drop and up, which cause outliers in the raw data. Besides, the prices of different stocks are various that cannot be evaluated with the same quantity scaler. For the ease of pattern mining, we use the time-based normalization to preprocess the stock prices:

$$\tilde{p}(t) = \frac{p(t) - \mu_\alpha(t)}{\sigma_\alpha(t)},$$

where $p(t)$ is the price of a certain stock at time t and $\mu_\alpha(t)$ and $\sigma_\alpha(t)$ are, respectively, the average value and standard deviation of the stock prices in the interval $[t - \alpha, t]$. We plot the normalized stock prices of three categories in Fig. 2.4. After normalization, we further adopt PCA method to reduce the dimensions of stocks from $\mathbb{R}^{200}$ to $\mathbb{R}^5$. Theoretically, the rank of subspaces after PCA should be $10 - 1 = 9$ because it contains 10 subspaces and the rank is degraded by 1 during PCA implementation. But, in the simulation, we find that the maximal clustering accuracies for both markets are achieved with the PCA dimensions of 5.

The clustering errors of different SC methods on the stocks from these two markets are summarized in Table 2.4. From the results, it is obvious that LHR significantly outperforms other methods. It improves statistic and graph-based methods for about

[4] For example, in the industrial category of drug manufactures, it is not possible to get the historical data of CIPILA.LTD from [28] which is the only the interface for us to get the stock prices in USA.

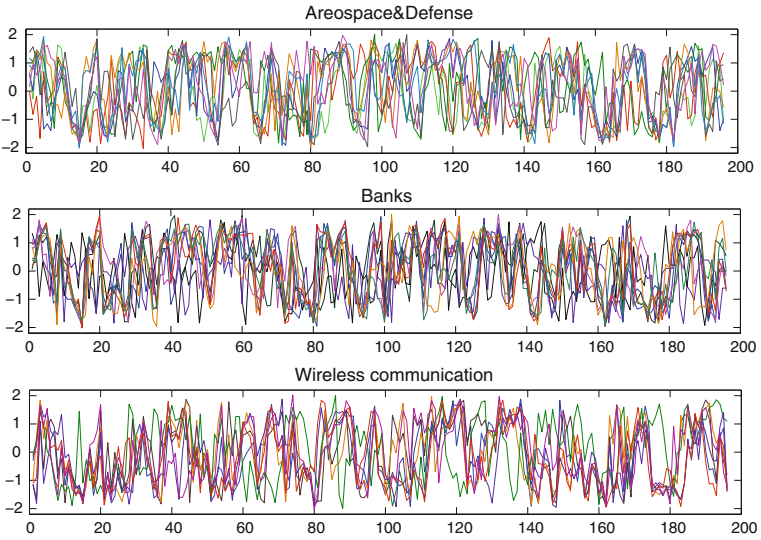

Fig. 2.4 Normalized stock prices in NY of the categories: aerospace and defense, banks, and wireless communication. In each category, lines in different colors represent different stocks (best viewed on screen)

20 %. Among all the sparse methods, LHR makes improvements on LRR for about 8 %. Although LHR performs the best among all the methods, the clustering accuracy is only about 63 and 61 % on US and Hong Kong markets, respectively. The clustering accuracy is not as high as those on the motion data. This may be ascribed to that the raw data and ground truth label themselves contain many uncertainties. See the bottom subfigure in Fig. 2.4 for the stocks in the *wireless communication* category, and the normalized stock marked with the green color performs quite different from other stocks in the same category. But the experimental results reported here are sufficient to verify the effectiveness of subspace clustering for 10 classes categorization. If no intelligent learning approaches were imposed, the expected accuracy may be only 10 %. Although with such "bad" raw data, the proposed LHR could achieve the accuracy as high as 62 % in a definitely unsupervised way.

2.5 Conclusion

This paper presents a log-sum heuristic recovery algorithm to learn the essential low-rank structures from corrupted matrices. We introduced a MM algorithm to convert the non-convex objective function a series of convex optimizations via reweighed approaches and proved that the solution may converge to a stationary point. Then, the general model was applied to two practical tasks of LRMR and SC. However, a limitation of the proposed LHR model is for the reweighted phenomenon that requires

to solve convex optimizations for multiple times. The implementation of LHR is a bit more time-consuming than PCP and LRR. Therefore, LHR model is especially recommended to learn the low-rank structure from data with denser corruptions and higher ranks.

References

1. Hu S, Wang J (2003) Absolute exponential stability of a class of continuous-time recurrent neural networks. IEEE Trans Neural Netw 14(1):35–45
2. Goldberg AB, Zhu X, Recht B, Xu J-M, Nowak RD (2010) Transduction with matrix completion: three birds with one stone. In: Advances in Neural Information Processing Systems, pp 757–765
3. Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2013) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35(1):171–184
4. Hu S, Wang J (2001) Quadratic stabilizability of a new class of linear systems with structural independent time-varying uncertainty. Automatica 37(1):51–59. Available http://www.sciencedirect.com/science/article/pii/S0005109800001229
5. Deng Y, Li Y, Qian Y, Ji X, Dai Q (2014) Visual words assignment via information-theoretic manifold embedding. IEEE Trans Cybern
6. Deng Y, Liu Y, Dai Q, Zhang Z, Wang Y (2012) Noisy depth maps fusion for multiview stereo via matrix completion. IEEE J Sel Top Sig Process 6(5):566–582
7. Deng Y, Dai Q, Zhang Z (2011) Graph laplace for occluded face completion and recognition. IEEE Trans Image Process 20(8):2329–2338
8. Deng Y, Dai Q, Wang R, Zhang Z (2012) Commute time guided transformation for feature extraction. Comput Vis Image Underst 116(4):473–483
9. Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theor 52(4):1289–1306
10. Recht B, Fazel M, Parrilo PA (2010) Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev 52(3):471–501
11. Deng Y, Dai Q, Liu R, Zhang Z, Hu S (2013) Low-rank structure learning via nonconvex heuristic recovery. IEEE Trans Neural Netw Learn Syst 24(3):383–396
12. Candès EJ, Li X, Ma Y, Wright J (2011) Robust principal component analysis? J ACM 58(3):11
13. Chandrasekaran V, Sanghavi S, Parrilo PA, Willsky AS (2011) Rank-sparsity incoherence for matrix decomposition. SIAM J Optim 21(2):572–596
14. Hsu D, Kakade SM, Zhang T (2010) Robust matrix decomposition with outliers. arXiv:1011.1518
15. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: Computer vision and pattern recognition. IEEE Conference on CVPR 2009. IEEE 2009, pp 1794–1801
16. Fazel M (2002) Matrix rank minimization with applications. Ph.D. dissertation, PhD thesis, Stanford University
17. Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ? 1 minimization. J Fourier Anal Appl 14(5–6):877–905
18. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J Roy Stat Soc: Series B (Stat Methodol) 67(2):301–320
19. Foo C-S, Do CB, Ng AY (2009) A majorization-minimization algorithm for (multiple) hyperparameter learning. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 321–328
20. Mohan K, Fazel M (2010) Reweighted nuclear norm minimization with application to system identification. In: American control conference (ACC). IEEE, pp 2953–2959

21. Hunter D, Li R (2005) Variable selection using mm algorithms. Ann Stat 33(4):1617
22. Lange K (1995) A gradient algorithm locally equivalent to the em algorithm. J Roy Stat Soc Series B (Methodol) 425–437
23. Lin Z, Chen M, Ma Y (2011) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report arXiv:1009.5055v2
24. Benedek C, Sziranyi T (2008) Bayesian foreground and shadow detection in uncertain frame rate surveillance videos. IEEE Trans Image Process 17(4):608–621
25. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24:381–395. Available http://doi.acm.org/10.1145/358669.358692
26. Vidal R, Ma Y, Sastry S (2003) Generalized principal component analysis (gpca). In: Proceedings of 2003 IEEE computer society conference on computer vision and pattern recognition, vol 1, pp I-621– I-628
27. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
28. Yahoo!finacial, http://finance.yahoo.com
29. Google finacial. Available http://www.google.com.hk/finance?q=

http://www.springer.com/978-3-662-44525-9

High-Dimensional and Low-Quality Visual Information Processing
From Structured Sensing and Understanding
Deng, Y.
2015, XV, 99 p. 23 illus., 18 illus. in color., Hardcover
ISBN: 978-3-662-44525-9