Signal Processing 164 (2019) 284–294

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Cauchy greedy algorithm for robust sparse recovery and multiclass classification

Yulong Wang[a], Cuiming Zou[a,*], Yuan Yan Tang[b], Luoqing Li[c], Zhaowei Shang[d]

[a] School of Information Science and Engineering, Chengdu University, Chengdu 610106, China
[b] Faculty of Science and Technology, University of Macau, Macau 999078, China
[c] Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China
[d] College of Computer Science, Chongqing University, Chongqing 400030, China

ARTICLE INFO

Article history:
Received 1 December 2018
Revised 28 May 2019
Accepted 5 June 2019
Available online 6 June 2019

Keywords:
Greedy algorithm
Robust signal recovery
Multiclass classification

ABSTRACT

Greedy algorithms have attracted considerable interest for sparse signal recovery (SSR) due to their appealing efficiency and performance recently. However, conventional greedy algorithms utilize the ℓ_2 norm based loss function and suffer from severe performance degradation in the presence of gross corruption and outliers. Furthermore, they cannot be directly applied to the recovery of quaternion sparse signals due to the noncommutativity of quaternion multiplication. To alleviate these problems, we propose a robust greedy algorithm referred as Cauchy matching pursuit (CauchyMP) for SSR and extend it for quaternion SSR. By leveraging the Cauchy estimator and generalizing it to the quaternion space to measure the residual error, our method can robustly recover the sparse signal *in both real and quaternion space* from noisy data corrupted by various severe noises and outliers. To tackle the resulting quaternion optimization problem, we develop an efficient half-quadratic optimization algorithm by introducing two quaternion operators. In addition, we have also devised a CauchyMP based classifier termed CauchyMPC for robust multiclass classification. The experiments on both synthetic and real-world datasets validate the efficacy and robustness of the proposed methods for SSR, block SSR, quaternion SSR and multiclass classification.

1. Introduction

Sparse representation (SR) has shown great potential in a variety of problems in signal processing and computer vision in the past decade [1]. For instance, SR methods have achieved great success in signal recovery [6], face recognition [1], and image restoration [8]. To be specific, given a dictionary matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$ $(m < n)$, SR aims to recover the target sparse signal $\mathbf{x}_0 \in \mathbb{R}^n$ from its compressed measurement vector

$$\mathbf{y} = \mathbf{D}\mathbf{x}_0 + \mathbf{n}, \tag{1}$$

where $\mathbf{n} \in \mathbb{R}^m$ denotes the noise vector.

According to the mechanisms of inducing sparsity, existing SR approaches can be roughly divided into two categories: ℓ_1 minimization based approaches [5] and greedy algorithms [2]. A natural SR approach is to optimize the following problem related to the ℓ_0 constraint

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \le K. \tag{2}$$

where K denotes the sparsity parameter. However, the problem above is generally NP-hard due to the discontinuity and discrete nature of the ℓ_0. This makes it intractable to solve the problem (2) directly. To reduce such problem, ℓ_1 minimization based approaches consider the following surrogate optimization problem

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{3}$$

which is termed as Lasso [5]. Here the nonnegative scalar λ is the regularization parameter. While Lasso enjoys striking theoretical properties under appropriate conditions, most ℓ_1 minimization based approaches require heavy computation burden [6].

Unlike Lasso, greedy algorithms (GA) iteratively identify the indexes of nonzero entries of $\mathbf{x}_0$ and estimate the sparse vector. Due to the low complexity and competitive performance, greedy algorithms have attracted increasing interest in recent years. Orthogonal matching pursuit (OMP) [2] is perhaps the most popular GA method because of its simplicity. Specifically, OMP identifies a column of the dictionary $\mathbf{D}$ in each iteration and estimates

* Corresponding author.
E-mail addresses: zoucuiming@cdu.edu.cn (C. Zou), yytang@umac.mo (Y.Y. Tang), lilq@hubu.edu.cn (L. Li), szw@cqu.edu.cn (Z. Shang).

https://doi.org/10.1016/j.sigpro.2019.06.006
0165-1684/

the sparse vector using the selected atoms in previous iterations. To improve the efficiency of OMP, the generalized OMP (GOMP) [2] selects multiple informative atoms in each iteration. The regularized OMP (ROMP) [4] algorithm tries to keep the advantages of both Lasso and OMP, i.e., the strong theoretical guarantees of Lasso and the high efficiency of OMP using a regularization rule. Analogously, the compressive sampling matching pursuit (CoSaMP) [3] also enhances OMP with an additional pruning step to provide strong theoretical guarantees that OMP cannot.

Despite their empirical success, most existing greedy algorithms explore the squared ℓ_2 norm as the loss function, which depends on the Gaussianity assumption of the noise distribution and sensitive to outliers. A violation of this assumption, e.g., missing entries, impulsive noise or random occlusions in face image data, may lead to severe performance degradation. To reduce the limitation, various robust SR approaches have been developed recently. The first category of robust SR methods aim to improve the robustness of lasso against outliers and gross corruptions. For instance, Carrillo and Barner [7] exploit the Lorentzian-norm to measure the residual error and introduce a geometric optimization problem for robust SR. Recent studies [8] have shown that leveraging the ℓ_1 norm based loss function in Lasso can lead to much better robustness compared with the conventional ℓ_2 norm and Lorentzian-norm. The resulting objective function is

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_1+\lambda\|\mathbf{x}\|_1.$$

To solve the $\ell_1-\ell_1$ norm optimization problem above, many effective algorithms have been devised such as YALL1 [9]. In [10], the generalized ℓ_p-norm, $0\le p<2$ is also adopted as the loss function for the residual error and the authors developed an alternating direction method based algorithm termed Lp-ADM for the corresponding optimization problem. Since these robust SR methods are still based on ℓ_1 norm regularization, these methods have heavy computational burden.

The second category of robust SR methods attempt to improve the robustness of greedy algorithms while keeping high efficiency. In [11], Razavi et al. attempt to robustify conventional greedy algorithms such as OMP by drawing on robust statistics and replacing least squares regression in OMP with robust regression. The robust OMP (RobOMP) [11] first calculates the so-called residual pseudo-values and selects a new atom which has the largest correlation with the residual pseudo-values in each iteration. In [12], Zeng et al. generalize conventional GA algorithms such as MP and OMP from inner product space to ℓ_p space ($p>0$) for robust SR. The robust version of MP and OMP are called ℓ_p-MP and ℓ_p-OMP, respectively.

Another weakness of most existing SR methods is that they are designed for real or complex sparse recovery and cannot be directly applied to quaternion sparse signal recovery (QSSR). Because the product of two quaternions are *noncommunicative* in general, i.e., $\dot{q}_1\dot{q}_2\neq\dot{q}_2\dot{q}_1$. In addition, the definition and computation of the derivative (or gradient) of quaternion matrix function are much more complicated than those in $\mathbb{R}^n$ or $\mathbb{C}^n$ [13,14]. This greatly increases the difficulty to tackle the quaternion optimization problems for QSSR.

In fact, quaternion has been widely used in various applications, including but not confined to, vector-sensor array signal processing [15], color face recognition [13], color image denoising, superresolution and inpainting [14]. Recent advances on quaternion image analysis [13,14] show that quaternions are well adapted to color images by encoding the color channels into the three imaginary parts. In [14], Xu et al. extend OMP to the quaternion space and devise the quaternion OMP (QOMP) algorithm with application to color image restoration. In our previous work [13], we propose the quaternion Lasso (QLasso) model with quaternion ℓ_1 minimization for QSSR and color face recognition. However, both QOMP and QLasso rely on the quaternion ℓ_2 norm based loss function and may be sensitive to gross corruption and outliers.

1.1. Paper contributions

In this paper, we develop a robust greedy algorithm referred as Cauchy Matching Pursuit (CauchyMP) by exploiting and generalizing the Cauchy estimator for robust sparse signal recovery (SSR) and quaternion SSR. By devising the half-quadratic theory [16] based optimization algorithm, CauchyMP can be viewed as an adaptive weighted OMP approach. The intuition behind CauchyMP is that it adaptively assign large weights on clean entries of $\mathbf{y}$ and small weights on noisy or outlying entries of $\mathbf{y}$. Accordingly, the impact of corrupted entries and outliers can be well alleviated. The contributions of this work are summarized as below.

1. We present a CauchyMP algorithm for robust sparse signal recovery. In the presence of gross corruption and outliers, CauchyMP can improve many prior greedy algorithms with notable performance gains.
2. We generalize CauchyMP and devise the quaternion CauchyMP (QCauchyMP) algorithm for the recovery of quaternion sparse signals. Since the product of quaternions is *noncommunicative* in general, previous robust SR approaches cannot be directly applied to quaternion sparse signal recovery.
3. We devise a CauchyMP based classifier termed CauchyMPC for robust multiclass classification and establish the theoretical analysis. Compared to the original sparse representation-based classification (SRC) [1], the proposed approach has better robust property and is more efficient.

The key differences between prior robust greedy algorithms (e.g., RobOMP and ℓ_p-OMP) and the proposed method lie in the following two aspects. *First*, the steps of identifying a new atom by these methods are different. Concretely, RobOMP identifies a new atom which has the largest correlation with the residual pseudo-values $\mathbf{e}_\psi$ and ℓ_p-OMP selects the atom that has the largest ℓ_p correlation with the residual. While CauchyMP selects a new atom most correlated with the residual in a reweighted version

$$j_k=\arg\max_{j=1,\cdots,n}\left|\left\langle\mathbf{d}_j,\mathbf{w}_{k-1}\otimes\mathbf{r}\right\rangle\right|,$$

where $\mathbf{w}_{k-1}$ is the weight vector indicating the importance of each entry of $\mathbf{r}$. For real vectors $\mathbf{u},\mathbf{v}\in\mathbb{R}^m$, the inner product is defined as $\langle\mathbf{u},\mathbf{v}\rangle=\mathbf{u}^T\mathbf{v}$, while for quaternion vectors $\dot{\mathbf{u}},\dot{\mathbf{v}}\in\mathbb{H}^m$, the inner product is defined by $\langle\dot{\mathbf{u}},\dot{\mathbf{v}}\rangle=\dot{\mathbf{u}}^H\dot{\mathbf{v}}$. Here $\dot{\mathbf{u}}^H=[\bar{\dot{u}}_1,\ldots,\bar{\dot{u}}_m]^T$ denotes the conjugate transpose of $\dot{\mathbf{u}}$ and $\bar{\dot{u}}_i$ is the conjugate of the quaternion $\dot{u}_i$. Specifically, if the ith entry r_i of $\mathbf{r}$ is severely corrupted, it will receive small weight (the ith entry of $\mathbf{w}_{k-1}$, which is estimated adaptively) and the impact of noisy entries can be effectively suppressed. Thus the identification step of CauchyMP has clear and intuitive explanation. *Second*, the steps of estimating the sparse signal by these methods are also distinct. Specifically, both RobOMP and ℓ_p-OMP use the Iteratively Re-weighted Least Squares (IRLS) algorithm while CauchyMP estimates the sparse signal by the half-quadratic theory based optimization algorithm with guaranteed convergence.

1.2. Paper outlines

The rest of this paper is organized as follows. In Section 2.2, we present the proposed methods for sparse signal recovery (SSR), block SSR, quaternion SSR and multiclass classification, respectively. Section 3 presents the experiments. Finally, Section 4 concludes the paper.

Table 1
Key notations and acronyms used in this paper.

Notation	Description
$\mathbf{x}_0 \in \mathbb{R}^n$	target sparse vector
$\mathbf{y} \in \mathbb{R}^m$	measurement vector
$\mathbf{D} \in \mathbb{R}^{m\times n}$	dictionary matrix
$\mathbf{n} \in \mathbb{R}^m$	noise vector
K	sparsity parameter
$\Lambda_k \subset \{1,\ldots,n\}$	the set of selected indexes in the kth iteration
$\lvert\Lambda_k\rvert$	cardinality of the set Λ_k
$\Lambda_k^c \subset \{1,\ldots,n\}$	the complement set of Λ_k
$\mathbf{x}_k \in \mathbb{R}^n$	estimation of the target sparse signal in the kth iteration
$\mathbf{x}_{k,\Lambda_k} \in \mathbb{R}^{\lvert\Lambda_k\rvert}\mathbf{r}_k \in \mathbb{R}^n$	reconstruction residual in the kth iteration
supp($\mathbf{x}$)	support set of the vector $\mathbf{x} \in \mathbb{R}^n$
$\mathcal{L}_C(\cdot)$	Cauchy estimator based loss function
$\mathcal{L}_{QC}(\cdot)$	Quaternion Cauchy estimator based loss function
SSR	Sparse Signal Recovery
CauchyMP	Cauchy Matching Pursuit
BCauchyMP	Block Cauchy Matching Pursuit
QCauchyMP	Quaternion Cauchy Matching Pursuit
CauchyMPC	Cauchy Matching Pursuit based Classifier

1.3. Notations

In this work, scalars, vectors and matrices are represented using italic letters (e.g., x), boldface lowercase letters (e.g., $\mathbf{x}$), and boldface capital letters (e.g., $\mathbf{X}$), respectively. For each vector $\mathbf{x} \in \mathbb{R}^n$ and an index set $\mathcal{J} \subset \{1,2,\ldots,n\}$, x_j denotes its jth entry and $\mathbf{x}_\mathcal{J}$ denotes a subvector of $\mathbf{x}$ containing entries indexed by the set $\mathcal{J}$. Analogously, for a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathbf{X}_\mathcal{J}$ denotes a submatrix of $\mathbf{X}$ containing columns of $\mathbf{X}$ indexed by the set $\mathcal{J}$. Table 1 summarizes the key notations and acronyms used in this paper.

2. The proposed approach

This section is arranged as follows. Firstly, we introduce the Cauchy estimator and propose a novel robust greedy algorithm referred to as Cauchy Matching Pursuit (CauchyMP) for robust SSR. Secondly, we generalize CauchyMP to the recovery of block sparse signal by exploiting the block sparsity. Thirdly, we extend CauchyMP and develop the quaternion CauchyMP (QCauchyMP) for the recovery of quaternion sparse signals. Finally, we devise a CauchyMP based classifier for robust multiclass classification as well as the theoretical analysis.

2.1. Cauchy estimator

We first introduce the concept of Cauchy estimator in robust statistics [17]. Let e_i be the ith entry of the residual $\mathbf{e} \in \mathbb{R}^m$, i.e., the difference between the observation and the reconstruction. The conventional least squares (LS) method minimizes the mean square error (MSE) $\min \sum_{i=1}^m e_i^2$, which is sensitive to outliers and large impulsive noise [17]. To reduce the drawback of the LS method, the Cauchy estimator [18] solve the following problem

$$\min_{\mathbf{e}\in\mathbb{R}^m} \mathcal{L}_C(\mathbf{e}) = \sum_{i=1}^m \rho_C(e_i) = \sum_{i=1}^m \log\left(1 + e_i^2/\tau^2\right), \tag{4}$$

where $\rho_C(e) = \log(1+e^2/\tau^2)$, $e \in \mathbb{R}$ and τ is a positive constant. The Cauchy estimator has been shown to own better robust property than the conventional MSE criterion [18]. In other words, the Cauchy estimator can tolerate nearly half incorrect observations among all observations in linear regression. However, in sparse recovery we need to simultaneously estimate the support set of the sparse signal and the magnitude of its nonzero entries.

2.2. Cauchy matching pursuit (CauchyMP)

To improve the robustness of conventional MP methods, we consider the following ℓ_0 constrained Cauchy loss minimization problem

$$\min_{\mathbf{x}} \mathcal{L}_C(\mathbf{y} - \mathbf{D}\mathbf{x}) \quad \text{subject to} \quad \|\mathbf{x}\|_0 \le K. \tag{5}$$

To implement (5), we propose a Cauchy matching pursuit (CauchyMP) algorithm. Firstly, we initialize the index set as $\Lambda_0 = \emptyset$, the residual as $\mathbf{r}_0 = \mathbf{y}$, the iteration counter $k = 0$, and a weight vector $\mathbf{w}_0 = \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^m$ denotes a vector of entries as ones. We first let $k = k+1$. In the k-th iteration, we identify a column $\mathbf{d}_j$ of $\mathbf{D}$ most correlated with the residual magnitude in a reweighted version

$$j_k = \arg\max_{j=1,\ldots,n} \left|\left\langle \mathbf{d}_j, \mathbf{w}_{k-1} \otimes \mathbf{r}\right\rangle\right|. \tag{6}$$

where $\otimes$ denotes the entrywise product. The identified index is incorporated into the index set by $\Lambda_k = \Lambda_{k-1} \cup \{j_k\}$. Then we obtain a new estimate $\mathbf{x}_k$ of the ground-truth sparse vector $\mathbf{x}_0$ by solving the optimization problem

$$\mathbf{x}_k = \arg\min_{\mathbf{x}\in\mathbb{R}^n, \text{supp}(\mathbf{x})\subset\Lambda_k} \mathcal{L}_C(\mathbf{y} - \mathbf{D}\mathbf{x}), \tag{7}$$

where supp($\mathbf{x}$) denotes the support set of $\mathbf{x}$, i.e., the index set of nonzero entries of $\mathbf{x}$. To tackle the problem in Eq. (7), we devise a half-quadratic (HQ) theory [16] based optimization algorithm. For ease of optimization, we first rewrite the objective function (7) in the following equivalent form

$$\mathbf{z}_k = \arg\min_{\mathbf{z}\in\mathbb{R}^{|\Lambda_k|}} f(\mathbf{z}) = \mathcal{L}_C\left(\mathbf{y} - \mathbf{D}_{\Lambda_k}\mathbf{z}\right), \tag{8}$$

where $|\Lambda_k|$ denotes the cardinality of the set Λ_k and $\mathbf{D}_{\Lambda_k}$ is the sub-dictionary of $\mathbf{D}$, which contains columns of $\mathbf{D}$ indexed by the set Λ_k. After getting $\mathbf{z}_k$, we can obtain the solution of the primal problem (7) by setting $\mathbf{x}_{k,\Lambda_k} = \mathbf{z}_k$ and $\mathbf{x}_{k,\Lambda_k^c} = \mathbf{0}$. Here $\mathbf{x}_{k,\Lambda_k}$ is the subvector of $\mathbf{x}_k$ indexed by Λ_k, and Λ_k^c denotes the complement set of Λ_k. Now we proceed to tackle the complicated optimization problem (8) by exploiting the half-quadratic theory [16].

Let $\mathbf{d}^i$ be the i-th row of the dictionary $\mathbf{D}_{\Lambda_k}$ for $i = 1,\ldots,m$. The loss function $f(\mathbf{z})$ can be formulated as

$$f(\mathbf{z}) = \frac{1}{m}\sum_{i=1}^m \log\left(1 + \frac{(y_i - \mathbf{d}^i\mathbf{z})^2}{\tau^2}\right). \tag{9}$$

In light of the half-quadratic theory [16,19], for the function $\rho_C(e) = \log\left(1 + e^2/\tau^2\right)$, there exists a dual potential function $\varphi(w)$, $w \in \mathbb{R}$ such that

$$\rho_C(e) = \inf\left\{\frac{1}{2}we^2 + \varphi(w), w \in \mathbb{R}\right\}, \tag{10}$$

where the infimum is reached at $w = \sigma(e)$, which is defined as [16,19]

$$\sigma(e) = \begin{cases} \rho_C'(e)/e, & \text{if } e \neq 0, \\ \rho_C''(0^+), & \text{if } e = 0. \end{cases} = \frac{2}{e^2 + \tau^2}. \tag{11}$$

Based on Eq. (10), the problem (9) is reformulated as

$$\min_{\mathbf{z},\mathbf{w}} J(\mathbf{z},\mathbf{w}) = \frac{1}{2}\left\|\mathbf{w}^{\frac{1}{2}} \otimes (\mathbf{y} - \mathbf{D}_{\Lambda_k}\mathbf{z})\right\|_2^2 + \sum_{i=1}^m \varphi(w_i), \tag{12}$$

where $\mathbf{w}^{\frac{1}{2}} = [w_1^{\frac{1}{2}},\ldots,w_m^{\frac{1}{2}}]^T \in \mathbb{R}^m$ and w_i is the ith entry of the weight vector $\mathbf{w}$. Let $\sigma(\mathbf{e}) = [\sigma(e_1),\ldots,\sigma(e_m)]^T$ for $\mathbf{e} \in \mathbb{R}^m$. Then we can alternatively minimize Eq. (12) as below

$$\mathbf{w}^{(t+1)} = \sigma\left(\mathbf{y} - \mathbf{D}_{\Lambda_k}\mathbf{z}^{(t)}\right), \tag{13}$$

$$\mathbf{z}^{(t+1)} = \arg\min_{\mathbf{z}\in\mathbb{R}^{|\Lambda_k|}} \frac{1}{2}\left\|\left(\mathbf{w}^{(t+1)}\right)^{\frac{1}{2}} \otimes (\mathbf{y} - \mathbf{D}_{\Lambda_k}\mathbf{z})\right\|_2^2, \tag{14}$$

where t denotes the iterations counter. The parameter τ can be estimated in an adaptive way as a function of median of e_is, i.e., $\tau = c \times \text{median}_i(|y_i - \mathbf{d}^i\mathbf{z}^{(t)}|) + \varepsilon_0$, where the scale constant c is experimentally set to be 0.5 [19] and ε_0 is a very small constant to avoid τ to be zero in Eq. (9). We have conducted experiments to analyze the sensitivity of CauchyMP to the parameter c and see the results in the supplementary material due to space limitation. Let $\mathbf{D}_{w,\Lambda_k} = \text{diag}((\mathbf{w}^{(t+1)})^{\frac{1}{2}})\mathbf{D}_{\Lambda_k}$ and $\mathbf{y}_w = \text{diag}((\mathbf{w}^{(t+1)})^{\frac{1}{2}})\mathbf{y}$. The solution to (14) is closed-form and can be written as

$$\mathbf{z}^{(t+1)} = \left(\mathbf{D}_{w,\Lambda_k}^T\mathbf{D}_{w,\Lambda_k}\right)^{-1}\mathbf{D}_{w,\Lambda_k}^T\mathbf{y}_w = \mathbf{D}_{w,\Lambda_k}^{\dagger}\mathbf{y}_w. \tag{15}$$

The convergence property of the optimization procedure above is guaranteed by the following proposition, which is the same as Remark 2 in [16].

Proposition 1. *The sequence* $\{J(\mathbf{z}^{(t)}, \mathbf{w}^{(t)})\}_{t=1}^{\infty}$ *by Eqs. (13) and (14) converges.*

Proof. Please see the proof in the supplementary material due to space limitation. □

Let $\mathbf{x}_k$ be the solution to (7) with the support set Λ_k. The next step is to update the residual $\mathbf{r}_k = \mathbf{y} - \mathbf{D}\mathbf{x}_k$ and the weight vector $\mathbf{w}_k = \sigma(\mathbf{r}_k)$. The iterations above will be repeated until convergence, i.e., when $\hat{\mathbf{x}}$ reaches the desired sparsity K or the reweighted residual $\|\mathbf{w}_k \otimes \mathbf{r}_k\|_2$ is not larger than the error tolerance ϵ. Algorithm 1 summarizes the CauchyMP algorithm.

Algorithm 1 Cauchy Matching Pursuit (CauchyMP).

Input: A signal $\mathbf{y} \in \mathbb{R}^m$, a measurement matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$, the sparsity K, and the error tolerance ϵ.
Initialization: $\Lambda_0 = \emptyset$, $\mathbf{r}_0 = \mathbf{y}$, $k = 0$, $\mathbf{w}_0 = \mathbf{1} \in \mathbb{R}^m$.
while $\|\mathbf{w}_k \otimes \mathbf{r}_k\|_2 > \epsilon$ and $k < K$ **do**

1: $k = k + 1$.
2: $j_k = \arg\max_{j=1,\dots,n} \left|\left\langle \mathbf{d}_j, \mathbf{w}_{k-1} \otimes \mathbf{r} \right\rangle\right|$.
3: $\Lambda_k = \Lambda_{k-1} \cup \{j_k\}$.
4: $\mathbf{x}_k = \arg\min_{\text{supp}(\mathbf{x})\subset\Lambda_k} \mathcal{L}_C(\mathbf{y} - \mathbf{D}\mathbf{x})$.
5: $\mathbf{r}_k = \mathbf{y} - \mathbf{D}\mathbf{x}_k$.
6: $\mathbf{w}_k = \sigma(\mathbf{r}_k)$.

while end
Output: sparse vector $\hat{\mathbf{x}} = \mathbf{x}_k$ and the weight vector $\hat{\mathbf{w}} = \mathbf{w}_k$.

Complexity analysis. For notational simplicity, let $h_k = |\Lambda_k|$. In the problem of sparse signal recovery, there holds $h_k \le K << n$ where K is the number of nonzero entries of the ground-truth sparse signal $\mathbf{x}_0 \in \mathbb{R}^n$. The update on the weight vector $\mathbf{w}$ in Eq. (13) has $O(mh_k)$ computational time in each iteration. As for Eq. (14), the computational cost is $O(mh_k^2)$ due to matrix inverse and matrix multiplication. Thus, the alternate minimization on $\mathbf{w}$ and $\mathbf{z}$ has $O(T_1(mh_k^2 + mh_k))$ computational time, where T_1 is the number of iterations in total.

2.3. Generalized CauchyMP for block sparse recovery

In this part, we aim to generalize the proposed CauchyMP algorithm for block or group sparse representation. Consider a set of block indexes $\mathcal{B} = \{\mathcal{B}_l\}_{l=1}^L$ which is a partition of the index set $\mathcal{I} = \{1, \dots, n\}$, i.e., $\bigcup_{l=1}^L \mathcal{B}_l = \mathcal{I}$ and $\mathcal{B}_h \bigcap \mathcal{B}_l = \emptyset$ for $1 \le h, l \le L$. Assume that the ground-truth block sparse vector $\mathbf{x}_0 = [\mathbf{x}_{0,\mathcal{B}_1}^T, \dots, \mathbf{x}_{0,\mathcal{B}_L}^T]^T$, whose nonzero entries appear in a few blocks. To take into account the block sparsity prior of $\mathbf{x}_0$, the objective function for block sparse representation is formulated as

$$\min_{\mathbf{x}} \mathcal{L}_C(\mathbf{y} - \mathbf{D}\mathbf{x}) \quad \text{subject to} \quad \|\mathbf{x}\|_{2,0} \le K, \tag{16}$$

where $\|\mathbf{x}\|_{2,0} = \sum_{l=1}^L I(\|\mathbf{x}_{\mathcal{B}_l}\|_2 > 0)$ and $I(\cdot)$ is the indicator function. To tackle the optimization problem (16), we generalize CauchyMP and propose the block CauchyMP (BCauchyMP) algorithm for block sparse representation. The main differences between CauchyMP and BCauchyMP lie in Step 2 and 3 in Algorithm 1. Specifically, in Step 2 the BCauchyMP algorithm selects a block of indexes corresponding to multiple columns of the dictionary $\mathbf{D}$ instead of a single column by

$$l_k = \arg\max_{l=1,\dots,L} \sum_{j\in\mathcal{B}_l} \left|\left\langle \mathbf{d}_j, \mathbf{w}_{k-1} \otimes \mathbf{r} \right\rangle\right|.$$

Then in Step 3 we augment the index set Λ_{k-1} by incorporating the selected block $\mathcal{B}_k$

$$\Lambda_k = \Lambda_{k-1} \cup \{\mathcal{B}_{l_k}\}.$$

The other steps of BCauchyMP and CauchyMP are the same and the description is omitted due to space limitation. It can be found that CauchyMP can be regarded as a special case of BCauchyMP with the block indexes $\mathcal{B} = \{\{1\}, \{2\}, \dots, \{n\}\}$, i.e., each block $\mathcal{B}_l$ has only one index.

2.4. Quaternion CauchyMP for robust sparse recovery in quaternion space

This section aims to generalize CauchyMP for the recovery of quaternion sparse signals. To begin with, we introduce some basic concepts of quaternion algebra. The quaternion space $\mathbb{H}$ proposed by Hamilton [20] is an extension of the complex $\mathbb{C}$ with three imagery units i, j and k. The imagery units are defined by

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1. \tag{17}$$

A quaternion $\dot{q} \in \mathbb{H}$ is formally written as

$$\dot{q} = q_0 + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}, \tag{18}$$

where $q_i \in \mathbb{R}$ $(i = 0, 1, 2, 3)$ is a real number and q_0 is referred as the *scalar part* of $\dot{q}$. Analogously, a quaternion vector $\dot{\mathbf{v}} \in \mathbb{H}^n$ is a vector of quaternions and is expressed as

$$\dot{\mathbf{v}} = \mathbf{v}_0 + \mathbf{v}_1\mathrm{i} + \mathbf{v}_2\mathrm{j} + \mathbf{v}_3\mathrm{k}, \tag{19}$$

where $\mathbf{v}_i \in \mathbb{R}^n$ $(i = 0, 1, 2, 3)$ is a real vector. The conjugate $\bar{\dot{q}}$ and modulus $|\dot{q}|$ of $\dot{q}$ are defined as

$$\bar{\dot{q}} = q_0 - q_1\mathrm{i} - q_2\mathrm{j} - q_3\mathrm{k},$$
$$|\dot{q}| = \sqrt{\dot{q}\bar{\dot{q}}} = \left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)^{\frac{1}{2}}.$$

The ℓ_2 norm of a quaternion vector $\dot{\mathbf{v}}$ is defined as $\|\dot{\mathbf{v}}\|_2 = (\sum_i |\dot{v}_i|^2)^{\frac{1}{2}}$, where $\dot{v}_i$ denotes the i-th entry of $\dot{\mathbf{v}}$. For a quaternion vector $\dot{\mathbf{v}} = [\dot{v}_1, \dots, \dot{v}_n]^T \in \mathbb{H}^n$, its conjugate transpose $\dot{\mathbf{v}}^H$ is defined as $\dot{\mathbf{v}}^H = [\bar{\dot{v}}_1, \dots, \bar{\dot{v}}_n]^T$. The *key* difference between quaternion space $\mathbb{H}$ and complex space $\mathbb{C}$ is that the product of two quaternions $\dot{q}_1$ and $\dot{q}_2$ is *noncommunicative*, i.e., $\dot{q}_1\dot{q}_2 \neq \dot{q}_2\dot{q}_1$ in general.

Given a quaternion dictionary $\dot{\mathbf{D}} \in \mathbb{H}^{m\times n}$, the goal of quaternion sparse signal recovery (QSSR) is to recover the quaternion sparse signal $\dot{\mathbf{x}} \in \mathbb{H}^n$ from its compressed quaternion measurement vector $\dot{\mathbf{y}} = \dot{\mathbf{D}}\dot{\mathbf{x}} + \dot{\mathbf{n}}$ where $\dot{\mathbf{n}}$ is the quaternion noise vector.

To generalize CauchyMP for QSSR, we first define the quaternion Cauchy estimator based loss function

$$\mathcal{L}_{QC}(\dot{\mathbf{e}}) = \sum_{i=1}^m \log\left(1 + |\dot{e}_i|^2/\tau^2\right), \tag{20}$$

where $|\dot{e}_i| := \sqrt{e_{i,0}^2 + e_{i,1}^2 + e_{i,2}^2 + e_{i,3}^2}$ is the module of $\dot{e}_i$. Then the objective function of QCauchyMP is formulated as

$$\min_{\dot{\mathbf{x}}} \mathcal{L}_{QC}(\dot{\mathbf{y}} - \dot{\mathbf{D}}\dot{\mathbf{x}}) \quad \text{subject to} \quad \|\dot{\mathbf{x}}\|_0 \le K. \tag{21}$$

To begin with, we initialize the index set $\Lambda_0 = \emptyset$, the residual $\dot{\mathbf{r}}_0 = \dot{\mathbf{y}}$, $k = 0$, and the weight vector $\mathbf{w}_0 = \mathbf{1} \in \mathbb{R}^m$, respectively. In the kth iteration, the first step is to identify a new atom most correlated with the residual in a weighted version by

$$j_k = \arg\max_{j=1,\dots,n} \left| \dot{\mathbf{d}}_j^H \left(\mathbf{w}_{k-1} \otimes \dot{\mathbf{r}}_{k-1} \right) \right|.$$

Here $\dot{\mathbf{d}}_j^H$ denotes the conjugate transpose of $\dot{\mathbf{d}}_j$ in the quaternion space $\mathbb{H}^m$. The second step is to add index of the selected atom into the index set $\Lambda_k = \Lambda_{k-1} \cup \{j_k\}$. Then we estimate the quaternion sparse signal with the support set Λ_k by

$$\dot{\mathbf{x}}_k = \arg\min_{\dot{\mathbf{x}} \in \mathbb{H}^n} \mathcal{L}_{\mathrm{QC}}(\dot{\mathbf{y}} - \dot{\mathbf{D}}\dot{\mathbf{x}}), \quad \text{s.t.} \quad \mathrm{supp}(\dot{\mathbf{x}}) \subset \Lambda_k. \tag{22}$$

which is equivalent to

$$\dot{\mathbf{z}}_k = \arg\min_{\dot{\mathbf{z}} \in \mathbb{H}^{|\Lambda_k|}} \mathcal{L}_{\mathrm{QC}}(\dot{\mathbf{y}} - \dot{\mathbf{D}}_{\Lambda_k}\dot{\mathbf{z}}), \tag{23}$$

where $\dot{\mathbf{x}}_{k,\Lambda_k} = \dot{\mathbf{z}}_k$ and $\dot{\mathbf{x}}_{k,\Lambda_k^c} = \dot{\mathbf{0}}$. Due to the noncommutativity of the quaternion multiplication, conventional real or complex optimization algorithms cannot be applied for the problem above. For this reason, we extend and revise the half-quadratic (HQ) based optimization algorithm in section (2.2) for QSSR. Analogously, we first reformulate the problem (23) according to the HQ theory as below

$$\min_{\dot{\mathbf{z}},\mathbf{w}} J(\dot{\mathbf{z}}, \mathbf{w}) = \frac{1}{2} \left\| \mathbf{w}^{\frac{1}{2}} \otimes (\dot{\mathbf{y}} - \dot{\mathbf{D}}_{\Lambda_k}\dot{\mathbf{z}}) \right\|_2^2 + \sum_{i=1}^{m} \varphi(w_i). \tag{24}$$

Then we can alternatively minimize (24) by the following iterations

$$\mathbf{w}^{(t+1)} = \sigma\left(\left| \dot{\mathbf{y}} - \dot{\mathbf{D}}_{\Lambda_k}\dot{\mathbf{z}}^{(t)} \right| \right), \tag{25}$$

$$\dot{\mathbf{z}}^{(t+1)} = \arg\min_{\dot{\mathbf{z}} \in \mathbb{H}^{|\Lambda_k|}} \frac{1}{2} \left\| \left(\mathbf{w}^{(t+1)} \right)^{\frac{1}{2}} \otimes (\dot{\mathbf{y}} - \dot{\mathbf{D}}_{\Lambda_k}\dot{\mathbf{z}}) \right\|_2^2, \tag{26}$$

where t denotes the number of iterations. Recall that in CauchyMP we obtain the solution of problem (14) by setting the gradient of the objective function with respect to $\mathbf{z}$ to zero vector. However, such strategy cannot be applied to the problem (26) since the gradient of quaternion function is are much more complicated due to the noncommutativity of quaternion multiplication [21]. In addition, the traditional QSVD (quaternion singular value decomposition) based method [14] has large computational burden to compute the quaternion pseudoinverse.

To tackle the problem (26), we define two useful quaternion operators $\mathcal{O}_1$ and $\mathcal{O}_2$ as below. Concretely, for any quaternion matrix $\dot{\mathbf{M}} = \mathbf{M}_0 + \mathbf{M}_1\mathrm{i} + \mathbf{M}_2\mathrm{j} + \mathbf{M}_3\mathrm{k} \in \mathbb{H}^{m\times n}$, we define the operator $\mathcal{O}_1 : \mathbb{H}^{m\times n} \to \mathbb{R}^{4m\times 4n}$ such that

$$\mathcal{O}_1(\dot{\mathbf{M}}) := \begin{bmatrix} \mathbf{M}_0 & -\mathbf{M}_1 & -\mathbf{M}_2 & -\mathbf{M}_3 \\ \mathbf{M}_1 & \mathbf{M}_0 & -\mathbf{M}_3 & \mathbf{M}_2 \\ \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_0 & -\mathbf{M}_1 \\ \mathbf{M}_3 & -\mathbf{M}_2 & \mathbf{M}_1 & \mathbf{M}_0 \end{bmatrix} \in \mathbb{R}^{4m\times 4n}. \tag{27}$$

For any quaternion vector $\dot{\mathbf{v}} = \mathbf{v}_0 + \mathbf{v}_1\mathrm{i} + \mathbf{v}_2\mathrm{j} + \mathbf{v}_3\mathrm{k} \in \mathbb{H}^n$, define the operator $\mathcal{O}_2 : \mathbb{H}^n \to \mathbb{R}^{4n}$ such that

$$\mathcal{O}_2(\dot{\mathbf{v}}) := \left[\mathbf{v}_0^T, \mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T \right]^T \in \mathbb{R}^{4n}. \tag{28}$$

Denote $\mathcal{O}_2^{-1}$ as the inverse of $\mathcal{O}_2$. Then we can derive the closed-form solution to the problem (26) by giving the following theorem.

Theorem 1. *The problem (26) has a closed-form solution*

$$\dot{\mathbf{z}}^{(t+1)} = \mathcal{O}_2^{-1}\left(\left(\mathcal{O}_1\left(\mathbf{W}\dot{\mathbf{D}}_{\Lambda_k} \right)^T \mathcal{O}_1\left(\mathbf{W}\dot{\mathbf{D}}_{\Lambda_k} \right) \right)^{-1} \mathcal{O}_1\left(\mathbf{W}\dot{\mathbf{D}}_{\Lambda_k} \right)^T \mathcal{O}_2\left(\mathbf{W}\dot{\mathbf{y}} \right) \right). \tag{29}$$

where $\mathbf{W} = \mathrm{diag}((\mathbf{w}^{(t+1)})^{\frac{1}{2}})$ *and* $\mathrm{diag}(\mathbf{v})$ *denotes a square matrix with the entries of the vector* $\mathbf{v} \in \mathbb{R}^n$ *on its main diagonal.*

Proof. Please see the proof in the supplementary material due to space limitation. □

Since both $\mathcal{O}_1(\mathbf{W}\dot{\mathbf{D}}_{\Lambda_k})$ and $\mathcal{O}_2(\mathbf{W}\dot{\mathbf{y}})$ are real matrices, the computation of Eq. (29) is straightforward. After obtaining the estimation $\dot{\mathbf{x}}_k$ of the quaternion sparse signal in the kth iteration, we update the residual and the weight vector by $\dot{\mathbf{r}}_k = \dot{\mathbf{y}} - \dot{\mathbf{D}}\dot{\mathbf{x}}_k$ and $\mathbf{w}_k = \sigma(|\dot{\mathbf{r}}_k|)$, respectively. Algorithm 2 summarizes the main procedure of QCauchyMP.

Algorithm 2 Quaternion Cauchy Matching Pursuit (QCauchyMP).

Input: A quaternion signal $\dot{\mathbf{y}} \in \mathbb{H}^m$, a quaternion dictionary matrix $\dot{\mathbf{D}} \in \mathbb{H}^{m\times n}$, the sparsity K, and the error tolerance ϵ.
Initialization: $\Lambda_0 = \emptyset$, $\dot{\mathbf{r}}_0 = \dot{\mathbf{y}}$, $k = 0$, $\mathbf{w}_0 = \mathbf{1} \in \mathbb{R}^m$.
while $\|\mathbf{w}_k \otimes \dot{\mathbf{r}}_k\|_2 > \epsilon$ and $k < K$ **do**

1: $k = k + 1$.
2: $j_k = \arg\max_{j=1,\dots,n} \left| \dot{\mathbf{d}}_j^H \left(\mathbf{w}_{k-1} \otimes \dot{\mathbf{r}}_{k-1} \right) \right|$.
3: $\Lambda_k = \Lambda_{k-1} \cup \{j_k\}$.
4: $\dot{\mathbf{x}}_k = \arg\min_{\mathrm{supp}(\dot{\mathbf{x}}) \subset \Lambda_k} \mathcal{L}_{\mathrm{QC}}(\dot{\mathbf{y}} - \dot{\mathbf{D}}\dot{\mathbf{x}})$.
5: $\dot{\mathbf{r}}_k = \dot{\mathbf{y}} - \dot{\mathbf{D}}\dot{\mathbf{x}}_k$.
6: $\mathbf{w}_k = \sigma(|\dot{\mathbf{r}}_k|)$.

while end
Output: quaternion sparse vector $\hat{\mathbf{x}} = \dot{\mathbf{x}}_k$ and the weight vector $\hat{\mathbf{w}} = \mathbf{w}_k$.

2.5. CauchyMP for multiclass classification

In fact, the proposed CauchyMP can also be extended for multiclass classification using the framework of sparse representation based classification (SRC) [1]. It is worthwhile to mention that the original SRC is based on the ℓ_1 minimization algorithm and has heavy computation burden for large scale problems as mentioned before.

Consider a classification problem with I classes. For each class $i (1 \le i \le I)$, let $\mathbf{A}_i = [\mathbf{a}_1^i, \dots, \mathbf{a}_{n_i}^i] \in \mathbb{R}^{m\times n_i}$ denote the matrix composed of n_i labeled training samples from class i. Let $n = \sum_{i=1}^{I} n_i$ be the number of total training samples. Let $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_I] \in \mathbb{R}^{m\times n}$ be the dictionary matrix of n labeled training samples from I classes. The goal is to classify any new possibly noisy test sample $\mathbf{y} \in \mathbb{R}^m$. In the first step, we normalize each training sample to have unit Euclidean norm to alleviate the impact of the data scale. The dictionary normalization is needed and important in many applications of sparse recovery, such as face recognition, image denoising, inpainting and super-resolution [1,14]. In the second step, CauchyMP is used to compute the sparse representation vector $\hat{\mathbf{x}}$ and the weight vector $\hat{\mathbf{w}}$ of $\mathbf{y}$. Then we calculate the class-dependent reconstruction residual of $\mathbf{y}$ for each class

$$R^i(\mathbf{y}) = \left\| \hat{\mathbf{w}}^{\frac{1}{2}} \otimes (\mathbf{y} - \mathbf{A}\delta_i(\hat{\mathbf{x}})) \right\|_2, \quad i = 1, 2, \dots, I,$$

where $\delta_i(\hat{\mathbf{x}})$ denotes the vector only keeping the nonzero entries of $\hat{\mathbf{x}}$ with respect to the ith class [1]. Finally, the identity of $\mathbf{y}$ is recognized as the class with the minimal residual.

Algorithm 3 summarizes the classification procedure of the CauchyMPC. As shown in Algorithm 3, the weight vector $\hat{\mathbf{w}}$ can be estimated adaptively from data by CauchyMPC without manually setting it. When the weight vector is estimated and fixed, CauchyMPC reduces to WOMPC (weighted orthogonal matching pursuit based classifier). The objective function of WOMP

Algorithm 3 CauchyMP based Classification (CauchyMPC).

Input: Training data matrix $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_I] \in \mathbb{R}^{m\times n}$, a test sample $\mathbf{y} \in \mathbb{R}^m$, the sparsity K, and the error tolerance ϵ.
Output: identity($\mathbf{y}$).

1: Normalize the columns of $\mathbf{A}$ to have unit Euclidean norm.
2: Compute the coefficient vector $\hat{\mathbf{x}}$ and the weight vector $\hat{\mathbf{w}}$ using CauchyMP.
3: Calculate the class-dependent residuals $R^i(\mathbf{y}) = \|\hat{\mathbf{w}}^{\frac{1}{2}} \otimes (\mathbf{y} - \mathbf{A}\delta_i(\hat{\mathbf{x}}))\|_2, i = 1, \ldots, I$.
4: Predict identity($\mathbf{y}$) $= \arg\min_{i=1,\ldots,I} R^i(\mathbf{y})$.

(weighted orthogonal matching pursuit) is formulated as

$$\min_{\mathbf{x}} \left\| \mathbf{w}^{\frac{1}{2}} \otimes (\mathbf{y} - \mathbf{A}\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \le K, \tag{30}$$

where $\mathbf{w} = [w_1, \ldots, w_m]^T \in \mathbb{R}^m$ is a weight vector and $\otimes$ denotes the Hadamard product. Due to the space limitation, we present the WOMP algorithm in the supplementary material. After obtaining the sparse vector $\hat{\mathbf{x}}$ using WOMP, WOMPC assigns the test sample to the class, which minimizes the class-dependent residual identity($\mathbf{y}$) $= \arg\min_{i=1,\ldots,I} \|\mathbf{w} \otimes (\mathbf{y} - \mathbf{A}\delta_i(\hat{\mathbf{x}}))\|_2$. In this part, we study theoretical conditions under which WOMPC succeeds in classifying a new test sample $\mathbf{y}$. To this end, we first introduce some useful notations.

The basic assumption of SRC [1] is that data samples from each class lie in a linear subspace. Let $\mathcal{S}_i = \text{span}(\mathbf{A}_i)$ denote the subspace of data from the i-th class for $i = 1, \ldots, I$. To take into account the nonnegative weighted vector $\mathbf{w} \in \mathbb{R}^m$, denote $\mathcal{S}_{i,\mathbf{w}} := \{\mathbf{w}^{\frac{1}{2}} \otimes \mathbf{u} | \mathbf{u} \in \mathcal{S}_i\}$. Let $\mathbf{A}_{-i}$ be the submatrix of the dictionary $\mathbf{A}$ by excluding the samples from the i-th class. To characterize the relationship between samples from different classes, we first introduce the weighted class coherence [22].

Definition 1. The weighted class coherence between the submatrix $\mathbf{A}_{-i}$ of $\mathbf{A}$ and the subspace $\mathcal{S}_i$ is defined as

$$\mu_{\mathbf{w}}(\mathbf{A}_{-i}, \mathcal{S}_i) = \max_{\mathbf{w}^{\frac{1}{2}} \otimes \mathbf{z} \in \mathcal{S}_{i,\mathbf{w}}} \frac{\left\| \mathbf{A}_{-i}^T \text{diag}(\mathbf{w})\mathbf{z} \right\|_\infty}{\left\| \mathbf{w}^{\frac{1}{2}} \otimes \mathbf{z} \right\|_2}, \tag{31}$$

where $\mathbf{w}$ denotes a weight vector with nonnegative entries.

Definition 2. The weighted class circumradius of $\mathbf{A}_i$ with respect to the subspace $\mathcal{S}_i$ and the nonnegative weight vector $\mathbf{w}$ is defined as

$$R_{\mathbf{w}}(\mathbf{A}_i) = \max_{\mathbf{u}} \left\| \mathbf{w}^{\frac{1}{2}} \otimes \mathbf{u} \right\|_2 \quad \text{s.t.} \quad \left\| \mathbf{A}_i^T \text{diag}(\mathbf{w})\mathbf{u} \right\|_\infty \le 1, \quad \text{and} \quad \mathbf{w}^{\frac{1}{2}} \otimes \mathbf{u} \in \mathcal{S}_{i,\mathbf{w}}. \tag{32}$$

The quantity $R_{\mathbf{w}}(\mathbf{A}_i)$ measures the distribution of weighted training samples in class i. It is essentially the radius of the smallest ball containing the polar set of $\text{conv}(\pm\sqrt{\text{diag}(\mathbf{w})}\mathbf{A}_i)$. In Fig. 1, the three training samples $\{\mathbf{a}_1^i, \mathbf{a}_2^i, \mathbf{a}_3^i\}$ in class i and their centrally symmetry points $\{-\mathbf{a}_1^i, -\mathbf{a}_2^i, -\mathbf{a}_3^i\}$ lie in unit circle and are drawn in red. The six blue points are the extreme points of the polar set $\mathcal{P}(\mathbf{A}_i) = \{\|\mathbf{A}_i^T\mathbf{u}\|_\infty \le 1, \mathbf{u} \in \mathcal{S}_i\}$ of the convex hull of the set $\{\pm\mathbf{a}_1^i, \pm\mathbf{a}_2^i, \pm\mathbf{a}_3^i\}$. Thus, the circumradius $R_{\mathbf{w}}(\mathbf{A}_i)$ is the largest magnitude of points in the polar set $\mathcal{P}(\mathbf{A}_i)$ above. As shown in Fig. 1, it has small value if the points in $\mathbf{A}_i$ are spread out. In fact, the class coherence and class circumradius have been defined in our previous work [23]. They can be regarded as special cases of Definitions 1 and 2 with equal weights, i.e., the weight vector $\mathbf{w} = [1, \ldots, 1]^T$.

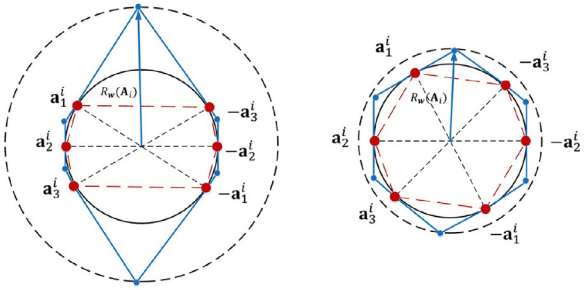


Fig. 1. Geometric illustration of the weighted class circumradius $R_{\mathbf{w}}(\mathbf{A}_i)$ with $\mathbf{w} = \mathbf{1}$ and $\mathbf{A}_i = [\mathbf{a}_1^i, \mathbf{a}_2^i, \mathbf{a}_3^i]$. Left: **large** $R_{\mathbf{w}}(\mathbf{A}_i)$; Right: **small** $R_{\mathbf{w}}(\mathbf{A}_i)$. It has small value if the points in $\mathbf{A}_i$ are well spread out. The three training samples $\{\mathbf{a}_1^i, \mathbf{a}_2^i, \mathbf{a}_3^i\}$ and their centrally symmetry points $\{-\mathbf{a}_1^i, -\mathbf{a}_2^i, -\mathbf{a}_3^i\}$ in class i lie in unit circle and are drawn in red. The six blue points are the extreme points of the polar set $\{\|\mathbf{A}_i^T\mathbf{u}\|_\infty \le 1, \mathbf{u} \in \mathcal{S}_i\}$ of the convex hull of the set $\{\pm\mathbf{a}_1^i, \pm\mathbf{a}_2^i, \pm\mathbf{a}_3^i\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Since the test sample $\mathbf{y}$ may be corrupted by various noises, it is not practical to assume $\mathbf{y} \in \bigcup_{i=1}^I \mathcal{S}_i \backslash \{\mathbf{0}\}$. Thus, we relax to assume $\mathbf{w}^{\frac{1}{2}} \otimes \mathbf{y} \in \bigcup_{i=1}^I \mathcal{S}_{i,\mathbf{w}} \backslash \{\mathbf{0}\}$ with an appropriate weight vector $\mathbf{w}$. With the notations above, we give the following theoretical results associated with the classification performance of WOMPC.

Theorem 2. *For any new test sample* $\mathbf{y}$ *satisfying* $\mathbf{w}^{\frac{1}{2}} \otimes \mathbf{y} \in \bigcup_{i=1}^I \mathcal{S}_{i,\mathbf{w}} \backslash \{\mathbf{0}\}$, *WOMPC succeeds in classifying* $\mathbf{y}$ *if*

$$\mu_{\mathbf{w}}(\mathbf{A}_{-i}, \mathcal{S}_i) R_{\mathbf{w}}(\mathbf{A}_i) < 1, \quad i = 1, 2, \ldots, I. \tag{33}$$

Proof. Please see the proof in the supplementary material due to space limitation. □

By recalling the definition of $\mu_{\mathbf{w}}(\mathbf{A}_{-i}, \mathcal{S}_i)$ and $R_{\mathbf{w}}(\mathbf{A}_i)$, the condition (33) can be satisfied if the samples in class i are away from the training samples in other classes and training samples in class i are well spread out. The weight vector serves to reduce the effect of noise and encourages the condition (33) to hold for severely noisy test sample. Consequently, Theorem 2 generalizes the results for clean test sample [22].

3. Experiments

In this section, we evaluate the performance of the proposed methods for SSR, block SSR, quaternion SSR and multiclass classification, respectively.

3.1. Sparse signal recovery

Settings. In this part, we assess the performance of CauchyMP for sparse signal recovery. Before showing the results, we illustrate how to generate the synthetic data for evaluation. Firstly, we generate a random dictionary matrix $\mathbf{D} \in \mathbb{R}^{100\times 400}$, and the entries of $\mathbf{D}$ are independent zero-mean unit-variance Gaussian random variables. Then we generate a sparse vector $\mathbf{x}_0 \in \mathbb{R}^{n=400}$. To this end, we randomly select five entries of $\mathbf{x}_0$ making their values as independent zero-mean unit-variance Gaussian random variables and set its rest entries as zeros. Accordingly, $\mathbf{x}_0$ has about five nonzero entries. Then we generate the noisy measurement vector $\mathbf{y} \in \mathbb{R}^{m=100}$ by

$$\mathbf{y} = \mathbf{D}\mathbf{x}_0 + \mathbf{n}, \tag{34}$$

where $\mathbf{n} \in \mathbb{R}^{m=100}$ denotes the random noise vector. In the experiments, we consider various noises, i.e., random missing entries, the bit error like corruption, three other noises following the chi-square χ^2 distribution, the Student's t-distribution, and Gaussian distribution, respectively. Specifically, we simulate missing entries

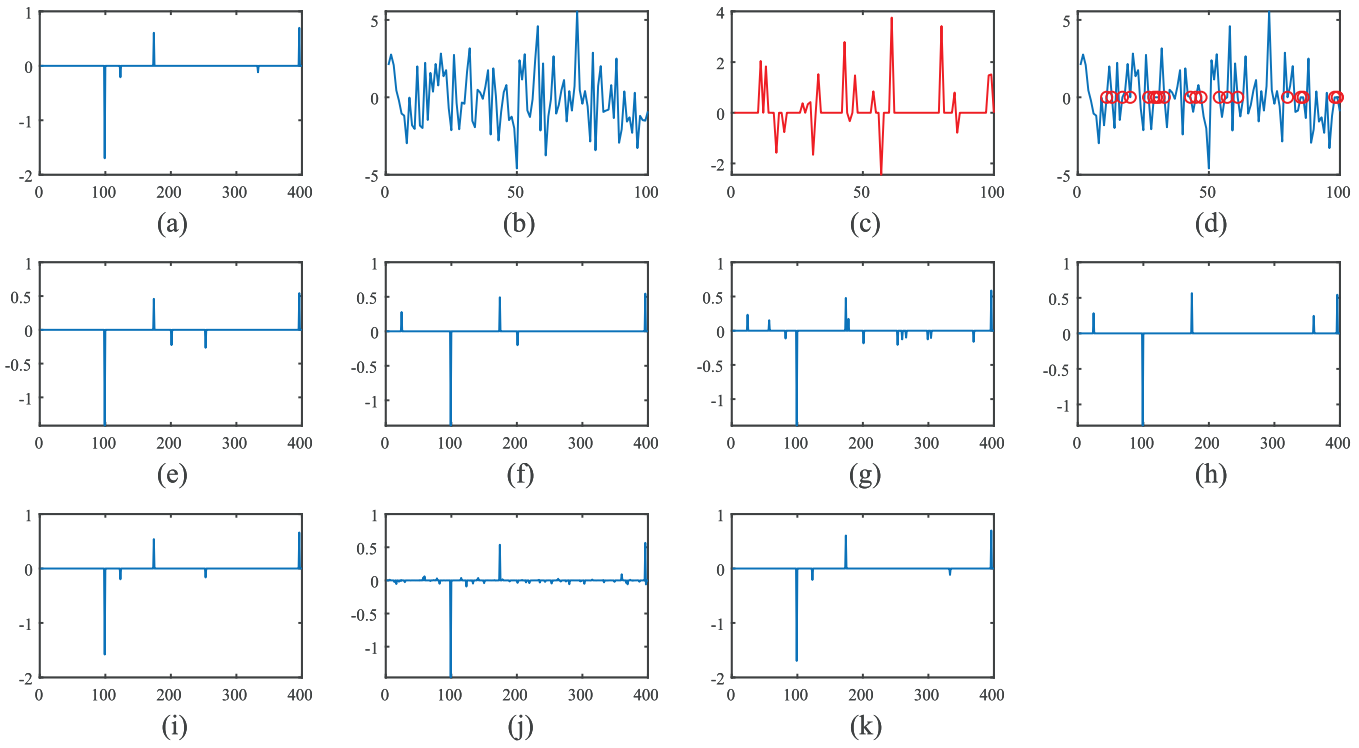


Fig. 2. Recovery performance of competing algorithms for sparse recovery against random missing measurements. (a) Test sparse signal; (b) Noiseless measurements; (c) noise; (d) Corrupted measurements (red circles mark the indices of missing entries); (e) OMP, RelErr=0.28; (f) GOMP, RelErr=0.29; (g) ROMP, RelErr=0.35; (h) CoSaMP, RelErr=0.32; (i) RobOMP, RelErr=0.13; (j) YALL1, RelErr=0.20; (k) CauchyMP, RelErr=1.8×10^{-5}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of $\mathbf{y}$ by randomly selecting a fraction of entries of $\mathbf{y}$ and setting them as zeros. Analogously, we simulate bit error like corruption by setting a fraction of randomly chosen measurements to be ±1000 to model the arbitrary unbounded errors. In the experiments, we use the codes provided by the corresponding authors and set the parameters of each method according to the corresponding paper. Specifically, for greedy algorithms we set the sparsity parameter $K=5$ for fair comparison while for the robust ℓ_1 method YALL1, we search the regularization parameter λ in a discrete set $\{10^{-4},10^{-3},10^{-2},10^{-1},1,10\}$.

For comparison, we consider several representative SR methods, including four popular greedy algorithms, i.e., OMP, GOMP [2], ROMP [4], and CoSaMP [3], and two robust SR methods, i.e., robust OMP (RobOMP) [11] and YALL1 [9]. For evaluation, we compute the relative recovery error (RelErr)

$$\text{RelErr}=\|\hat{\mathbf{x}}-\mathbf{x}_0\|_2/\|\mathbf{x}_0\|_2,$$

where $\hat{\mathbf{x}}$ denotes the recovered sparse vector by a certain algorithm.

Results: Random missing entries. Fig. 2 shows the sparse recovery performance of the competing algorithms against 20% random missing measurements. It is worth mentioning that it is the measurement vector $\mathbf{y}$ that has missing entries *rather than* the sparse signal $\mathbf{x}_0$, which is to be recovered from $\mathbf{y}$. Note that CauchyMP significantly outperforms other competing algorithms with much smaller relative recovery error. To make the results convincing, we vary the percentage $p(\%)$ of random missing entries from 0% to 60% and conduct sparse recovery experiments with 10 random runs. Fig. 3 shows the recovery error as a function of the percentage of missing entries. Note that CauchyMP is more robust to missing entries compared with other competing algorithms. For example, when 30% entries of the observation vector $\mathbf{y}$ are missing, CauchyMP has nearly vanishing recovery error while the recovery error of other competing methods exceed 0.25.

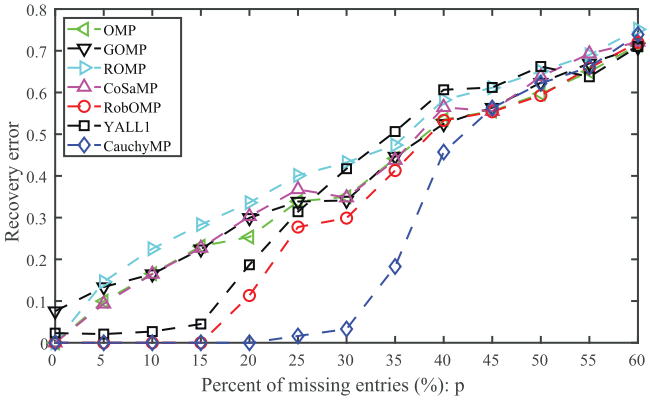


Fig. 3. Average recovery error of various SR algorithms as a function of the percentage p of random missing measurements over 10 random runs.

Results: Bit error like corruption. We also test the performance of CauchyMP against bit error like corruption. The bit errors are widespread in transmission, malfunctioning pixels, and faulty memory locations [10]. As mentioned before, we simulate bit error like corruption by randomly choosing a fraction of measurements and setting their values as ±1000 to model the arbitrary unbounded errors. Fig. 4 shows the recovery performance of the competing algorithms when 20% randomly selected noisy measurements are corrupted with bit error like corruption. Note that most prior GA methods break down with very large relative recovery error in this scenario. While CauchyMP significantly outperforms them with the relative recovery error 0.05.

For more convincing evaluation, we also vary the percentage $p(\%)$ of corrupted measurements from 0% to 60% and run the recovery experiments. Fig. 5 shows the relative recovery error of var-

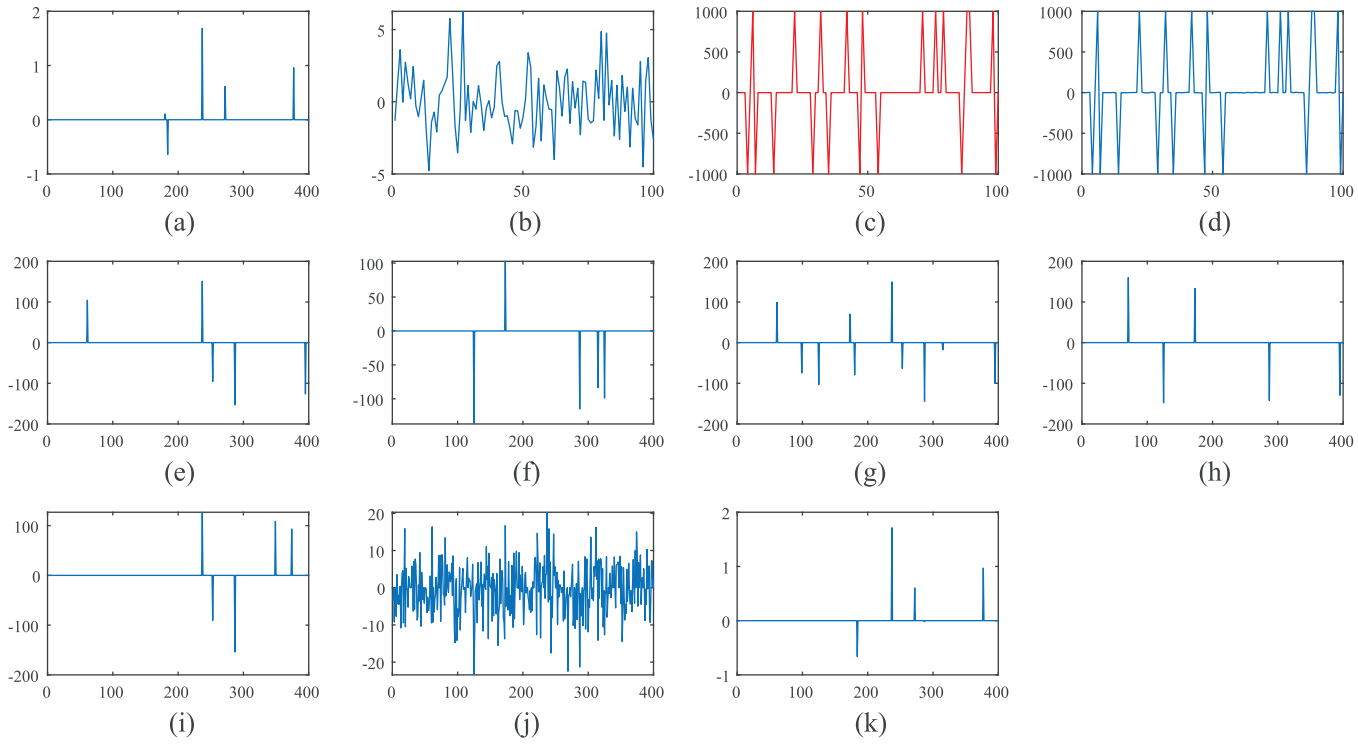


Fig. 4. Recovery performance of competing algorithms for block sparse recovery against bit error like corruption. (a) Test sparse signal; (b) Noiseless measurements; (c) Bit error noise; (d) Corrupted measurements; (e) OMP, RelErr=133.70; (f) GOMP, RelErr=113.91; (g) ROMP, RelErr=143.66; (h) CoSaMP, RelErr=149.64; (i) RobOMP, RelErr=122.10; (j) YALL1, RelErr=60.16; (k) CauchyMP, RelErr=0.05.

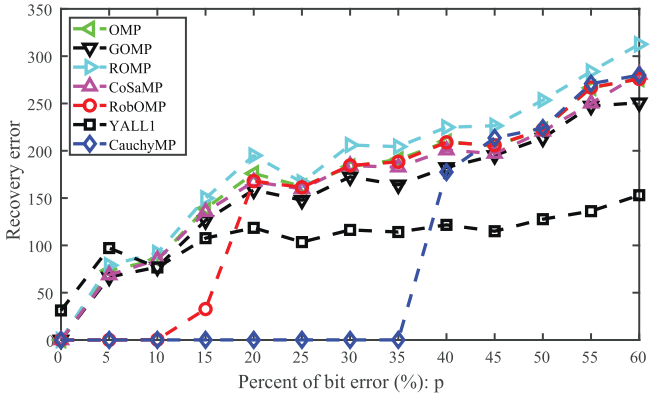


Fig. 5. Average recovery error of various SR algorithms as a function of the percentage p of measurements corrupted by bit error like corruption over 10 random runs.

ious SR algorithms as a function of the percentage of corrupted measurements, averaged over 10 random runs. With 35% bit error like corruption, the relative recovery error of CauchyMP is less than 0.2 while the errors of other methods exceed 50. The results demonstrate the strong robustness and superiority of CauchyMP over competing methods against bit error.

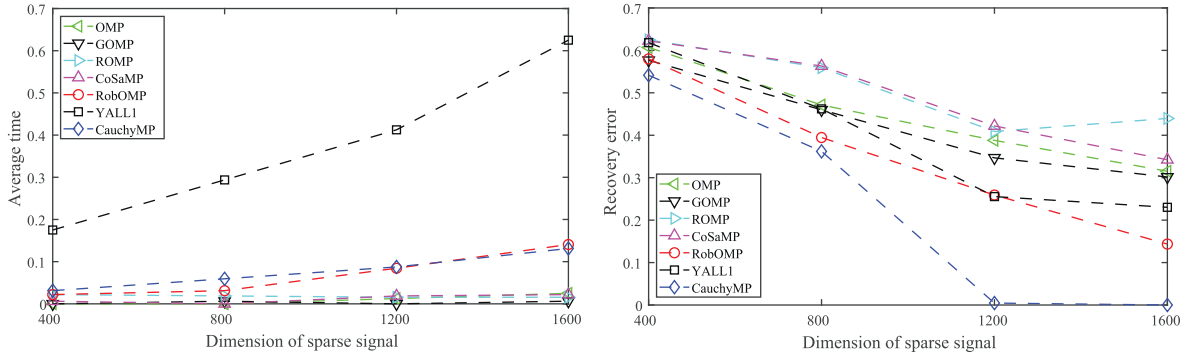
Results: Various noises. In this part, we evaluate the performance of CauchyMP for SSR under various noises, including both non-Gaussian and Gaussian noises. In the experiments, we consider 30% random missing entries, 30% bit error like corruption, chi-square χ^2 noise with 1 degree of freedom, the Student's t-distribution with 2 degrees of freedom, and the standard Gaussian distribution $\mathcal{N}(0,1)$, respectively.

To obtain reliable results, we repeat each experiments with ten random runs and compute the average evaluation indexes. Table 2 reports the average relative recovery error and the standard deviation of distinct SR algorithms over ten-run test with various noises. Note that CauchyMP outperforms other competing SR algorithms with smaller average relative recovery error in the presence of the four non-Gaussian noises aforementioned. In addition, the three robust SR algorithms, i.e., RobOMP, YALL1 and CauchyMP outperforms the other four greedy algorithms. However, with the Gaussian noise OMP, GOMP and CoSaMP obtain lower recovery error than the three robust SR methods.

Results: Comparison of the computing cost. In this part, we analyze and compare the computing cost of CauchyMP with other

Table 2
Average relative recovery error and standard deviation of various sparse representation methods with distinct noises over 10 random runs. Best results are highlighted.

Noise	Missing entries	Bit error	Chi-square	Student	Gaussian
OMP	0.35 ± 0.08	171.42 ± 79.23	0.49 ± 0.34	0.80 ± 0.41	0.13 ± 0.07
GOMP	0.36 ± 0.08	152.97 ± 81.14	0.49 ± 0.28	0.74 ± 0.39	0.14 ± 0.11
ROMP	0.43 ± 0.09	183.72 ± 93.22	0.59 ± 0.18	0.91 ± 0.44	0.21 ± 0.15
CoSaMP	0.38 ± 0.07	158.64 ± 80.26	0.42 ± 0.17	0.76 ± 0.47	$\mathbf{0.12} \pm 0.06$
RobOMP	0.33 ± 0.09	168.61 ± 84.99	0.35 ± 0.19	0.50 ± 0.25	0.15 ± 0.12
YALL1	0.44 ± 0.14	0.36 ± 0.19	0.42 ± 0.13	0.52 ± 0.13	0.21 ± 0.13
CauchyMP	$\mathbf{0.04} \pm 0.12$	$\mathbf{0.14} \pm 0.10$	$\mathbf{0.28} \pm 0.18$	$\mathbf{0.45} \pm 0.22$	0.15 ± 0.13



(a) Time (b) Recovery error

Fig. 6. Average running time and recovery error of various SSR algorithms as a function of the dimension n of the sparse signal over 10 random runs.

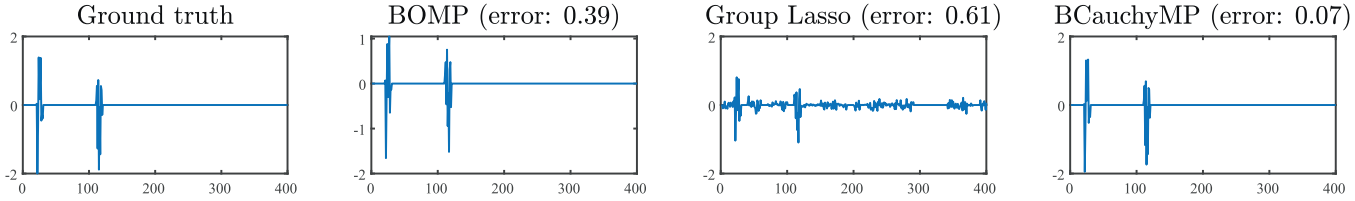


Fig. 7. Recovered performance of a block sparse signal using BOMP, Group Lasso and BCauchyMP with 30% random missing entries of $\mathbf{y}$.

competing algorithms for SSR. In the experiment, we vary the dimension n of the ground-truth sparse signal $\mathbf{x}_0$ from 400 to 1600. We set $m = n/4$ and the sparsity of $\mathbf{x}_0$ as $s = n/40$, respectively. As for the noise vector, we consider 40 random missing entries of the measurement vector $\mathbf{y}$. Fig. 6 shows the average computation time in seconds and the average recovery error of various SSR algorithms as a function of the dimension n of the ground-truth sparse signal. Note that CauchyMP achieves much better recovery performance in terms of recovery error with reasonable running time in comparison with other competing methods.

3.2. Block sparse signal recovery

Settings. In this part, we assess the performance of BCauchyMP for block sparse signal recovery (BSSR). Specifically, we generate the block sparse signal $\mathbf{x}_0 \in \mathbb{R}^{n=400}$ as follows. We first construct the block set $\mathcal{B} = \{\mathcal{B}_l\}_{l=1}^{L}$ with $L = 40$ and the size of each block $\mathcal{B}_l$ is 10. Concretely, $\mathcal{B}_l = \{10 \times (l-1) + 1, \ldots, 10 \times l\}$ for $l = 1, \ldots, L$. Then we randomly select $s = 2$ blocks of $\mathbf{x}_0$ and set their entries as zero-mean unit-variance Gaussian random variables. The rest entries of $\mathbf{x}_0$ are set as zeros. Then $\mathbf{x}_0$ has the block sparsity s. We compare BCauchyMP with block OMP (BOMP) and group lasso [24]. Here BOMP is the generalization of OMP for BSSR using the same strategy as BCauchyMP.

Now we clarify the parameter settings of each method. For BOMP and BCauchyMP, the block sparsity parameter K is set as 2 for fair comparison. As for group lasso, we use the algorithm implemented by the ADMM (Alternating Direction Method of Multipliers) framework [25] and the regularization parameter λ is searched from the candidate set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$.

Results. Fig. 7 shows the BSSR results against 30% missing entries using BOMP, Group Lasso and BCauchyMP. As shown in Fig. 7, BCauchyMP yields the recovered result with smaller recovery error than those by other competing methods. Table 3 shows the recovery performance of distinct BSSR methods against various noises.

3.3. Quaternion sparse signal recovery

Settings. In this part, we evaluate the performance of QCauchyMP for the recovery of quaternion sparse signal. We first generate a random quaternion dictionary matrix $\dot{\mathbf{D}} = \mathbf{D}_0 + \mathbf{D}_1\mathrm{i} + \mathbf{D}_2\mathrm{j} + \mathbf{D}_3\mathrm{k} \in \mathbb{H}^{100\times 400}$, where the entries of $\mathbf{D}_i$ are independent zero-mean unit-variance Gaussian random variables. We generate a quaternion sparse signal $\dot{\mathbf{x}} = \mathbf{x}_0 + \mathbf{x}_1\mathrm{i} + \mathbf{x}_2\mathrm{j} + \mathbf{x}_3\mathrm{k} \in \mathbb{H}^{n=400}$ by randomly selecting five entries of $\dot{\mathbf{x}}$ as nonzero entries and set its rest entries as zeros. Then we generate the noisy quaternion measurement vector by $\dot{\mathbf{y}} = \dot{\mathbf{D}}\dot{\mathbf{x}} + \dot{\mathbf{n}}$, where $\dot{\mathbf{n}} \in \mathbb{H}^{m=100}$ denotes the quaternion random noise vector.

Results. In the first experiment, we test the performance of QCauchyMP for quaternion sparse signal recovery (QSSR) against 20% random missing entries compared with QOMP [14] and QLasso [13]. Let $\dot{\mathbf{x}}$ and $\hat{\dot{\mathbf{x}}}$ be the ground-truth quaternion sparse signal and the estimation by some QSSR algorithm. Then the relative recovery error of the QSSR algorithm is $\mathrm{RelErr} = \|\hat{\dot{\mathbf{x}}} - \dot{\mathbf{x}}\|_2 / \|\dot{\mathbf{x}}\|_2$. For QOMP and QCauchyMP, we set the sparsity parameter $K = 5$ for fair comparison while for QLasso we search the regularization parameter from the candidate set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Fig. 8 shows the ground-truth quaternion sparse signal and the recovered results by different algorithms. Compared with QOMP and QLasso, QCauchyMP achieves much better recovery performance in terms of relative recovery error.

Table 3
Average recovery error and standard deviation of various block sparse representation methods with distinct noises over 10 random runs. Best results are highlighted.

Noise	Missing entries	Bit error	Chi-square	Student	Gaussian
BOMP	0.37 ± 0.09	72.85 ± 11.79	0.20 ± 0.05	0.26 ± 0.15	$\mathbf{0.13} \pm 0.02$
Group Lasso	0.64 ± 0.06	70.66 ± 9.67	0.41 ± 0.06	0.48 ± 0.12	0.25 ± 0.05
BCauchyMP	$\mathbf{0.12} \pm 0.13$	$\mathbf{8.49} \pm 23.13$	$\mathbf{0.12} \pm 0.04$	$\mathbf{0.15} \pm 0.03$	0.14 ± 0.03

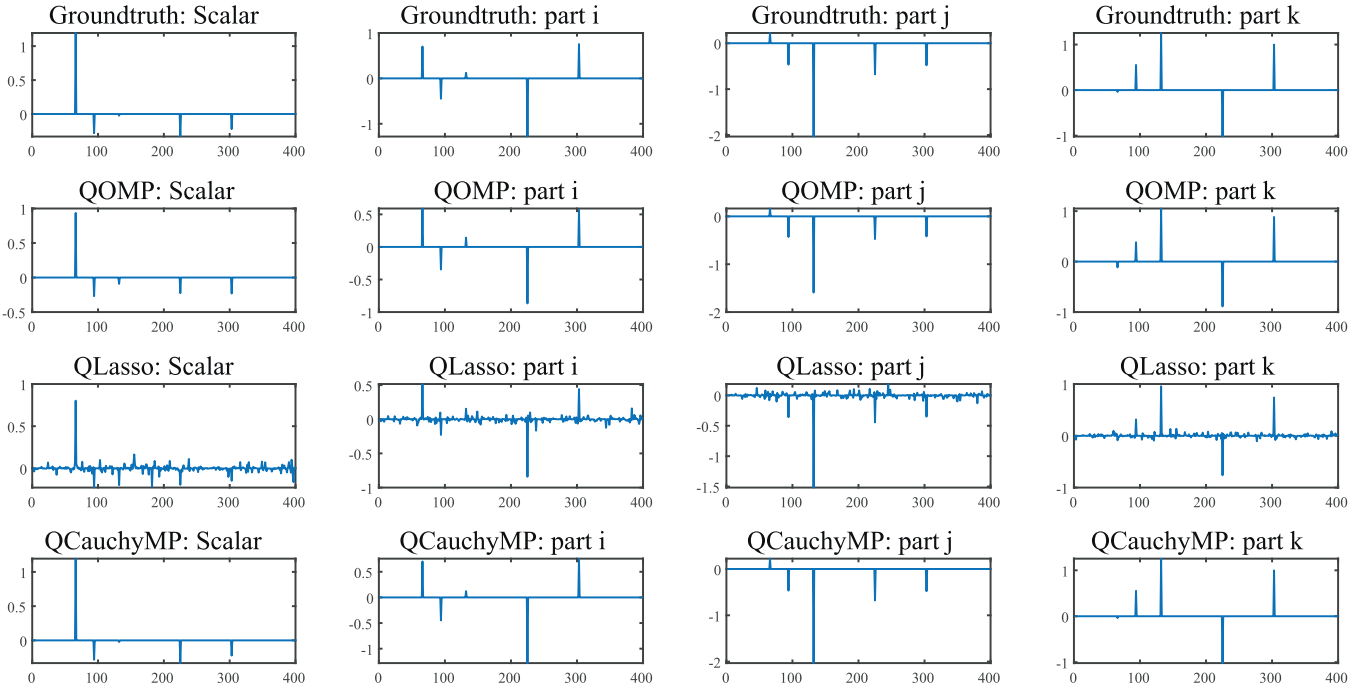


Fig. 8. Recovery performance of competing algorithms for quaternion sparse recovery against random missing measurements with sparsity level $s = 5$. From top row to bottom row: ground-truth quaternion sparse signal, recovered results of QOMP (RelErr=0.22), QLasso (RelErr=0.43) and QCauchyMP (RelErr=7×10^{-5}). In each row, from left to right we show the scalar part and three imagery parts of the ground-truth or recovered quaternion sparse signal.

Fig. 9. Sample images from (a) the Extended Yale B database; (b) the CMU PIE database.

In the second experiment, we assess the recovery performance of QCauchyMP for QSSR in the presence of various noises. The settings of the noises are analogous to those for SSR in Section 3.2. Table 4 lists the recovery results of the three QSSR algorithms over ten random runs.

3.4. Face recognition

Settings. In this part we assess the performance of the CauchyMP based classifier for robust face recognition against random pixel corruption. Two benchmark databases are used and they are the Extended Yale B database [26] and the CMU PIE database [27]. The former database has over 2000 facial images of 38 subjects with different illumination. Each subject has about 64 facial images and each image is resized to have 32×32 pixels. The latter database contains 41,368 facial images of 68 subjects with varying pose, illumination, and expression. In the experiment, we utilize a subset composed of the near frontal facial images of pose C05 and each image is resized to 32×32. Fig. 9 shows some sample images from these two databases. For comparison, we consider OMP [2], GOMP [2], ROMP [4], CoSaMP [3] and RobOMP [11]. These methods are also applied into classification and the resulting classifiers are referred as OMPC (OMP based classifier), GOMPC, ROMPC, CoSaMPC and RobOMPC, respectively. CauchyMPC is also compared with the SRC method [1] by ℓ_1 minimization. As for the parameter settings, we set the sparsity parameter K of each greedy algorithm as 10 for fair comparison and searching the regularization parameter λ of SRC in a discrete set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. For each database, we randomly select half images per subject for training and the rest for testing. For each test image, we corrupt a fraction of randomly selected pixels of it by replacing their values with random values following uniform distribution over [0, 255].

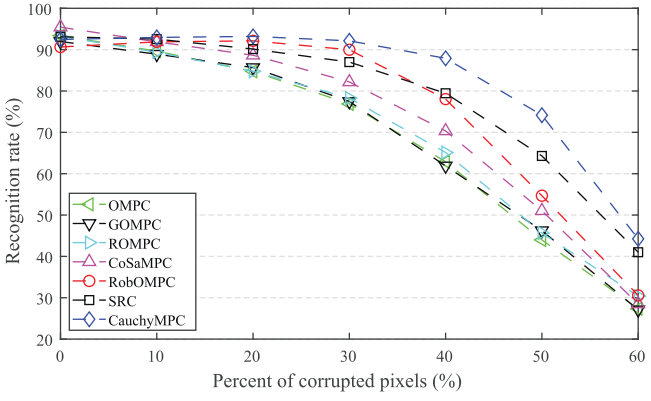


Fig. 10. Recognition rates as a function of percentage of corrupted pixels in each test image using the Extended Yale B database.

Results. Figs. 10 and 11 plot the recognition rates as a function of the percentage of corrupted pixels in each test image using the

Table 4
Average recovery error and standard deviation of various quaternion sparse recovery methods with distinct noises over 10 random runs. Best results are highlighted.

Noise	Missing entries	Bit error	Chi-square	Student	Gaussian
QOMP	0.37 ± 0.09	72.85 ± 11.79	0.20 ± 0.05	0.26 ± 0.15	$\mathbf{0.05} \pm 0.01$
QLasso	$0.64 \pm \mathbf{0.06}$	$70.66 \pm \mathbf{9.67}$	0.41 ± 0.06	0.48 ± 0.12	0.14 ± 0.03
QCauchyMP	$\mathbf{0.12} \pm 0.13$	$\mathbf{8.49} \pm 23.13$	$\mathbf{0.12} \pm \mathbf{0.04}$	$\mathbf{0.15} \pm \mathbf{0.03}$	0.06 ± 0.02

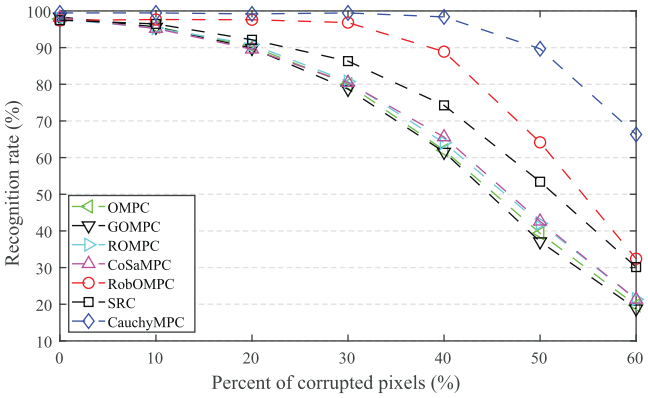


Fig. 11. Recognition rates as a function of percentage of corrupted pixels in each test image using the CMU PIE database.

Extended Yale B database and CMU PIE database, respectively. Note that CauchyMPC achieves higher recognition rates than other classifiers in most cases, especially with large percentage of corruption.

4. Conclusion

This paper presents a novel greedy algorithm referred as CauchyMP for robust sparse signal recovery in real and quaternion space and multiclass classification. Specifically, CauchyMP leverages the robust Cauchy estimator to recover the sparse signal, which can tolerate noisy data contaminated with various severe noise. The experiments demonstrate the efficacy of the proposed methods for sparse recovery and classification.

Disclosure of conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61806027, 61702057, 61672114 and 11771130.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sigpro.2019.06.006.

References

[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2) (2009) 210–227.
[2] J. Wang, S. Kwon, B. Shim, Generalized orthogonal matching pursuit, IEEE Trans. Signal Process. 60 (12) (2012) 6202–6216.
[3] D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, Appl. Comp. Harmonic Anal. 26 (3) (2009) 301–321.
[4] D. Needell, R. Vershynin, Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit, IEEE J. Sel. Topics Signal Process. 4 (2) (2010) 310–316.
[5] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. Ser. B 58 (1) (1996) 267–288.
[6] D. Donoho, Y. Tsaig, I. Drori, J.L. Starck, Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit, IEEE Trans. Inf. Theory 58 (2) (2012) 1094–1121.
[7] R. Carrillo, K. Barner, Lorentzian iterative hard thresholding: robust compressed sensing with prior information, IEEE Trans. Signal Process. 61 (19) (2013) 4822–4833.
[8] D.S. Pham, S. Venkatesh, Efficient algorithms for robust recovery of images from compressed data, IEEE Trans. Image Process. 22 (12) (2013) 4724–4737.
[9] J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (2011) 250–278.
[10] F. Wen, P. Liu, Y. Liu, R. Qiu, W. Yu, Robust sparse recovery in impulsive noise via ℓ_p-ℓ_1 optimization, IEEE Trans. Signal Process. 65 (1) (2017) 105–118.
[11] S. Razavi, E. Ollila, V. Koivunen, Robust greedy algorithms for compressed sensing, in: Proceedings of the 20th European Signal Process Conference (EUSIPCO), 2012, pp. 969–973.
[12] W. Zeng, H. So, X. Jiang, Outlier-robust greedy pursuit algorithms in ℓ_p-space for sparse approximation, IEEE Trans. Signal Process. 64 (1) (2016) 60–75.
[13] C. Zou, K.I. Kou, Y. Wang, Quaternion collaborative and sparse representation with application to color face recognition, IEEE Trans. Image Process. 25 (7) (2016) 3287–3302.
[14] Y. Xu, L. Yu, H. Xu, H. Zhang, T. Nguyen, Vector sparse representation of color image using quaternion matrix analysis, IEEE Trans. Image Process. 24 (4) (2015) 1315–1329.
[15] X. Zhang, W. Liu, Y. Xu, Z. Liu, Quaternion-valued robust adaptive beamformer for electromagnetic vector-sensor arrays with worst-case constraint, Signal Process. 104 (2014) 274–283.
[16] M. Nikolova, M.K. Ng, Analysis of half-quadratic minimization methods for signal and image recovery, SIAM J. Sci. Comput. 27 (3) (2005) 937–966.
[17] P.J. Huber, Robust Statistics, Wiley, New York, 1981.
[18] I. Mizera, C.H. Muller, Breakdown points of cauchy regression-scale estimators, Stat. Probab. Lett. 57 (1) (2002) 79–89.
[19] R. He, W. Zheng, T. Tan, Z. Sun, Half-quadratic-based iterative minimization for robust sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 261–275.
[20] W. Hamilton, On quaternions; or on a new system of imaginaries in algebra, Dublin Philosoph. Mag. J. Sci 25 (163) (1844) 10–13.
[21] D. Xu, D. Mandic, The theory of quaternion matrix derivatives, IEEE Trans. Signal Process. 63 (6) (2015) 1543–1556.
[22] C. You, D. Robinson, R. Vidal, Scalable sparse subspace clustering by orthogonal matching pursuit, Proceedings of the IEEE Conference Computer Vision and Pattern Recognition(2016) 3918–3927.
[23] Y. Wang, Y. Tang, L. Li, H. Chen, J. Pan, Atomic representation-based classification: theory, algorithm and applications, IEEE Trans. Pattern Anal. Mach. Intell. 41 (1) (2019) 6–19.
[24] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B 68 (1) (2006) 49–67.
[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011) 1–122.
[26] K. Lee, J. Ho, D. Driegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[27] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Trans. Pattern Anal. Mach. Intell. 25 (12) (2003) 1615–1618.