



# 启发式联合 PCD 快速降噪算法

何选森<sup>1,2\*</sup>, 徐丽<sup>1</sup>, 许莹<sup>1</sup>

(1. 广州商学院信息技术与工程学院 广州 511363; 2. 湖南大学信息科学与工程学院 长沙 410082)

**【摘要】**基于稀疏冗余表示, 并行坐标下降 (PCD) 是最优秀的降噪算法之一。然而在音频信号处理中, 当分割的帧数很大时, PCD 的计算负担加重, 造成运行时间剧增。该文基于联合稀疏表示 (JSR) 和同时稀疏近似 (SSA), 建立一种新的时域框架, 并提出了一种启发式联合 PCD 算法 (Joint-PCD)。在该框架中, 每个音频帧作为一个列向量生成信号矩阵; 利用超完备字典, Joint-PCD 同步 (同时) 对信号矩阵降噪, 提高了算法的运行效率, 减小了运行时间。仿真结果表明: Joint-PCD 算法不仅与 PCD 具有几乎相同的降噪性能, 且把 PCD 算法的降噪速度提高了约 5 倍, 极大地改进了 PCD 算法的收敛性能。

**关键词** 迭代收缩降噪; 联合稀疏表示; 并行坐标下降; 同时稀疏近似; 超完备字典

**中图分类号** TN911.4, TN911.7 **文献标志码** A **doi**:10.12178/1001-0548.2020053

## Novel Heuristic Joint PCD Fast Denoising Algorithm

HE Xuan-sen<sup>1,2\*</sup>, XU Li<sup>1</sup>, and XU Ying<sup>1</sup>

(1. School of Information Technology and Engineering, Guangzhou College of Commerce Guangzhou 511363;

2. College of Information Science and Engineering, Hunan University Changsha 410082)

**Abstract** Based on sparse and redundant representation, parallel coordinate descent (PCD) is one of the best denoising algorithms. In audio signal processing, however, when the number of segmented frames is large, the computational burden is heavy. Processing each frame separately with the PCD algorithm causes a dramatic increase in time cost. Therefore, this paper proposes a new time-domain framework and a heuristic joint PCD (called joint-PCD) algorithm based on joint sparse representation (JSR) and simultaneous sparse approximation (SSA). In this framework, each audio frame is used as a column vector to generate a signal matrix. Utilizing an over-complete dictionary, joint-PCD is used to synchronously (simultaneously) denoise a signal matrix (instead of an audio frame), which greatly improves the efficiency of the algorithm and reduces the burden of running time. The simulation results show that the joint-PCD algorithm not only has almost the same and excellent denoising performance with PCD, but also increases the denoising speed of PCD by about five times, which greatly improves the convergence performance of the PCD algorithm.

**Key words** iterative shrinkage denoising; joint sparse representation; parallel coordinate descent; simultaneous sparse approximation; over-complete dictionary

在对信号的描述中, 基于冗余字典的稀疏表示提供了一种强有力的信号模型, 它是将获得的观测信号近似为字典中少量原子的线性组合<sup>[1]</sup>。由于噪声无处不在, 纯粹的干净信号是无法得到的, 对信号降噪已经成为信号处理最重要的任务之一。稀疏表示本身具有潜在的降噪能力<sup>[2]</sup>, 特别是联合稀疏表示 (joint sparse representation, JSR)<sup>[3]</sup> 和同时稀疏近似 (simultaneous sparse approximation, SSA)<sup>[4]</sup> 在信号降噪领域都获得了广泛的应用。另外, 收缩是建立在稀疏表示基础上的具有吸引力的降噪技术<sup>[5]</sup>,

著名的小波收缩使用正交变换实现降噪<sup>[6]</sup>。然而, 新的趋势是采用超完备变换, 它对信号的分解不是唯一的。例如: 帧方法、匹配追踪法、最佳正交基法等, 而基本追踪 (basis pursuit, BP)<sup>[7]</sup> 法是通过优化技术将信号分解为字典元素的最佳 (叠加) 表示, 且使表示系数的  $l_1$  范数 (或  $l_0$  范数) 最小化<sup>[8]</sup>。将 BP 应用于含噪声的信号, 即基本追踪降噪 (BP denoising, BPDN)<sup>[9]</sup> 方法对于消除高斯白噪声非常有效。同样地, 冗余字典的发展也催生了高效的数值技术, 即迭代收缩 (iterative shrinkage, IS) 算法<sup>[10]</sup>。

收稿日期: 2020-02-15; 修回日期: 2020-12-10

基金项目: 国家自然科学基金 (60572183)

作者简介: 何选森 (1958-), 男, 教授, 主要从事统计信号处理、盲源分离及通信系统方面的研究。E-mail: xshe2010@163.com

为解决 BPDN 问题, 文献 [11] 提出了 4 种 IS 算法, 分别是启发式收缩、基于迭代重加权最小二乘、阶段式正交匹配追踪、并行坐标下降 (parallel coordinate descent, PCD) 算法。其中 PCD 的搜索速度最快。因此, 文献 [12] 又对 PCD 进行了算法的收敛性分析, 并引入了一种正则化函数形式, 使得 PCD 能对坐标优化提供解析的解。

利用 PCD 对每个含噪声的音频 (语音) 帧进行降噪能获得整个信号很好的降噪效果。然而, 若信号很长, 则分割的音频帧数量就很大, 用 PCD 对每帧分别进行降噪将导致繁重的计算负担。为此, 本文建立了一种以信号矩阵为操作对象的时域处理框架, 并提出了基于 SSA 和 JSR 的启发式联合 PCD 算法 (Joint-PCD)。在此框架中, 每个音频帧作为一个列向量生成信号矩阵, Joint-PCD 算法每次是对一个信号矩阵 (而不仅仅是对一个音频帧) 进行降噪, 从而极大地降低了算法的运行时间开销, 提高了算法的收敛速度。

## 1 稀疏信号的降噪

在对信号进行表示时, 最常用、最简单的方法就是对信号进行线性变换。用  $\mathbf{s} \in \mathbf{R}^N$  表示  $N$  维信源矢量,  $\mathbf{x} \in \mathbf{R}^N$  表示对应的观测矢量,  $\mathbf{T}$  表示  $N \times N$  变换矩阵, 则  $\mathbf{x} = \mathbf{T}\mathbf{s}$ 。由于变换后的信号数量等于信源的数量, 则  $\mathbf{T}$  称为正交变换。信号的另一种重要表示是线性稀疏表示。假设在同一时刻只有少量的分量  $x_i \in \mathbf{x}$  是非零的, 其他绝大多数分量都为零值, 则称  $\mathbf{x}$  服从稀疏分布。若信号  $\mathbf{x}$  被高斯噪声  $\mathbf{v}$  污染, 通过对  $\mathbf{x}$  的稀疏分量进行软阈值收缩可实现  $\mathbf{x}$  的降噪<sup>[6]</sup>。

信号  $\mathbf{x} \in \mathbf{R}^N$  一般由几个单位能量原子  $d_k$  的线性组合近似表示, 所有原子组成的集合  $\{d_k, k=1, 2, \dots, K\}$  构成一个字典  $\mathbf{D} \in \mathbf{R}^{N \times K}$ 。观测向量  $\mathbf{x}$  可以表示为:

$$\mathbf{x} = \mathbf{D}\mathbf{z} \quad (1)$$

式中,  $\mathbf{z} \in \mathbf{R}^K$  是稀疏的, 是  $K$  个系数的列向量。当  $K=N$  时,  $\mathbf{D}$  为正交, 其表示也是唯一的; 当  $K>N$  时,  $\mathbf{D}$  为冗余, 式 (1) 存在无限多个可能的系数集。冗余字典的优势体现在可以从众多的可能性中选择最适合的一种表示, 其中最吸引人的表示法是求具有最少数量的非零的系数解:

$$\min_z \|\mathbf{z}\|_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \quad (2)$$

式 (2) 是寻找欠定线性方程组  $\mathbf{x} = \mathbf{D}\mathbf{z}$  的最稀疏

解<sup>[13]</sup>。优化式 (2) 还可以等价地表示为<sup>[13]</sup>:

$$\min_z \|\mathbf{z}\|_0 \quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 \leq \varepsilon \quad (3)$$

式中,  $\varepsilon$  是误差容限;  $\|\cdot\|_p$  表示  $l_p$  范数;  $l_0$  范数是对一个向量非零元素的计数;  $l_2$  范数采用欧几里得范数。式 (2) 和式 (3) 可用正交匹配追踪法求解<sup>[11]</sup>。式 (2) 还可表示为<sup>[7]</sup>:

$$\min_z \|\mathbf{z}\|_1 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \quad (4)$$

式中,  $l_1$  范数是求一个向量元素绝对值的总和。由于  $l_1$  范数是  $l_0$  范数的凸化<sup>[13]</sup>, 因此, 式 (4) 可以看作是式 (2) 的凸化。

对于含噪声的观测信号矢量:

$$\mathbf{y} = \mathbf{x} + \mathbf{v} \quad (5)$$

式中,  $\mathbf{x} \in \mathbf{R}^N$  是干净的信号向量;  $\mathbf{v} \in \mathbf{R}^N$  是能量为  $\delta$  的高斯零均值白噪声向量。对信号  $\mathbf{y}$  降噪的优化问题表示为<sup>[8]</sup>:

$$\min_z \|\mathbf{z}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{z}\|_2^2 \leq \delta \quad (6)$$

优化式 (6), 可由拉格朗日形式代替<sup>[8]</sup>:

$$\min_z \frac{1}{2} \|\mathbf{D}\mathbf{z} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{z}\|_1 \quad (7)$$

式中,  $\lambda$  是用于调节近似误差与系数向量稀疏性之间关系的一个参数。实际上, 式 (7) 就是 PCD 算法的目标函数<sup>[11]</sup>。

一般地, 从含噪声信号矢量  $\mathbf{y}$  中去除高斯白噪声的过程可归结为对下式函数的最小化<sup>[11]</sup>:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \rho(\mathbf{x}) \quad (8)$$

式中, 第一项为对数似然, 它描述了干净信号  $\mathbf{x}$  与含噪声向量  $\mathbf{y}$  之间的关系; 第二项的  $\rho(\mathbf{x})$  表示  $\mathbf{x}$  的先验值, 不同的  $\rho(\mathbf{x})$  对应于不同的  $f(\mathbf{x})$  展开式。将式 (1) 代入式 (8) 中, 得到信号降噪的目标函数为<sup>[11]</sup>:

$$\min_z \frac{1}{2} \|\mathbf{D}\mathbf{z} - \mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho(\mathbf{z}) \quad (9)$$

式中,  $\mathbf{1}_K$  表示元素为 1 的  $K \times 1$  矩阵 (列向量)。

迭代收缩 (IS) 算法是为了解决目标函数为式 (9) 这样的混合  $l_2$ - $l_p$  ( $p \leq 1$ ) 优化问题而产生的新型高效数值技术。比较式 (9) 与式 (7) 可知: 在 PCD 算法中, 函数  $\rho(\mathbf{z})$  是  $l_1$  范数。

式 (7) 表示的 PCD 算法的操作对象是向量, 在音频信号降噪过程中, PCD 每次是对一个音频帧

(列向量)降噪。若音频帧数量巨大,则PCD算法的运行时间将急剧增加。特别对于大规模数据和实时信号的降噪问题,分别用PCD对每帧降噪是不现实的,必须寻找新的方法同时(并行)对多个列向量降噪,以减少算法的计算负担。

## 2 Joint-PCD 启发式算法

用冗余字典原子的线性组合近似表示一个音频帧,称之为简单稀疏近似(simple sparse approximation);对于被相同信道噪声污染的多个帧(矢量),用相同字典原子的不同线性组合同时近似表示,称之为JSR。通过对多个向量同时稀疏表示的理论分析和研究<sup>[14]</sup>,从而就产生了求解SSA问题的策略<sup>[15]</sup>。对于音频(语音)信号在加性高斯白噪声(additive white Gaussian noise, AWGN)信道的传输与降噪,SSA具有重要意义。

为了对一个音频信号的所有帧同时进行降噪,需建立新的时域框架以区别传统的帧处理模式。这种框架的基本思想为:把信号中每个被分割的帧按顺序作为一个矩阵的列向量,信号与矩阵唯一对应。由于矩阵中的各个列向量是同一个音频信号的不同帧,因此具有相同的稀疏性。基于新的时域框架,就可将简单稀疏近似推广为SSA,并将向量形式的PCD扩展为矩阵形式的Joint-PCD,从而利用冗余(超完备)字典原子的不同线性组合,实现对一个音频信号所有帧同时进行降噪处理。

令  $y_1 = x_1 + v_1$ ,  $y_2 = x_2 + v_2$  分别为干净信号  $x_1$ ,  $x_2 \in \mathbf{R}^N$  的含噪声观测量,其中  $v_1, v_2 \in \mathbf{R}^N$  分别为加性噪声向量。 $x_1, x_2$  可以由同一个超完备字典  $D \in \mathbf{R}^{N \times K}$  近似表示为  $x_1 = Dz_1, x_2 = Dz_2$ , 其中  $z_1, z_2 \in \mathbf{R}^K$  分别为重建系数。由式(7)可得:

$$\begin{cases} \min_{z_1} \frac{1}{2} \|Dz_1 - y_1\|_2^2 + \lambda \|z_1\|_1 \\ \min_{z_2} \frac{1}{2} \|Dz_2 - y_2\|_2^2 + \lambda \|z_2\|_1 \end{cases} \quad (10)$$

联合考虑这两个优化问题,对信号向量  $y_1, y_2$  进行同时降噪的目标函数可表示为:

$$\min_{z_1, z_2} \frac{1}{2} \{ \|Dz_1 - y_1\|_2^2 + \|Dz_2 - y_2\|_2^2 \} + \lambda \|z_1\|_1 + \lambda \|z_2\|_1 \quad (11)$$

再令  $X = [x_1, x_2]$ ,  $Y = [y_1, y_2]$ ,  $Z = [z_1, z_2]$ , 则将式(11)简化为标准的稀疏编码问题:

$$\min_Z \frac{1}{2} \|DZ - Y\|_F^2 + \lambda \|Z\|_1 \quad (12)$$

对于  $Z$  和  $D$  两个变量,式(12)并不是一个凸

优化问题,只有当其中一个变量给定时,它才是凸优化<sup>[3]</sup>。本文中超完备字典  $D$  是给定的,通过优化  $Z$  来构造 Joint-PCD 算法。式(12)还可以推广到多个(大于2个)信号矢量的情况。

对于多个观测信号的矩阵模型:

$$Y = X + V \quad (13)$$

式中,  $X \in \mathbf{R}^{N \times P}$  是包含了  $P$  个向量(音频帧)  $x_i \in \mathbf{R}^N$  的干净信号矩阵;  $V \in \mathbf{R}^{N \times P}$  是  $P$  个噪声向量  $v_i \in \mathbf{R}^N$  组成的矩阵;  $Y \in \mathbf{R}^{N \times P}$  是含噪声的观测信号矩阵。基于联合稀疏表示(JSR)<sup>[3]</sup>,干净信号矩阵  $X$  可以由超完备字典  $D$  表示为  $X = DZ$ , 则含噪声的观察信号矩阵为:

$$Y = DZ + V \quad (14)$$

式中,  $Z \in \mathbf{R}^{K \times P}$  是稀疏系数矩阵,其每个列向量  $z_i \in \mathbf{R}^K$  是每个信号向量(帧)  $x_i$  的稀疏表示。假设噪声的总能量有限,即  $\sum \|v_k\|_2^2 \leq \delta^2$ 。为了恢复理想的表示系数矢量  $z_i$ ,可以通过求解如下的数学规划问题:

$$\min_Z (\# \text{ of nonzero rows in } Z) \quad \text{s.t. } \|DZ - Y\|_F < \delta \quad (15)$$

式中,  $\|\cdot\|_F$  是矩阵的 Frobenius 范数。如果系数矩阵  $Z$  中的非零行向量能正确地辨识出  $z_i$  中的非零元素,就可利用线性的方法估计  $z_i$  值。因此,可以把向量形式的降噪优化式(9)扩展为如下的矩阵形式:

$$\min_Z \frac{1}{2} \|DZ - Y\|_F^2 + \lambda \cdot \mathbf{1}^T \cdot \rho(Z) \quad (16)$$

式中,  $\rho(\cdot)$  是任意标量函数,而  $\rho(Z)$  是衡量矩阵  $Z$  稀疏性的惩罚函数。式(16)是 Joint-PCD 算法的目标函数。

综合考虑式(15)和式(16),可以归纳出构造 Joint-PCD 算法的基本思想为:通过最小化  $Z$  中非零行的数量来限制参与对  $Y$  进行逼近的字典原子的总数。采用两方面的措施来实现。首先,希望参与对  $Y$  近似的原子数量越少越好;其次,希望每个参与近似的原子要尽可能多地对矩阵  $Y$  的各列向量做出贡献。对于系数矩阵  $Z$ ,希望它的大多数行向量都是全零元素,而它的每个非零行(向量)都应包含尽可能多的非零元素。基于这种考虑,将函数  $\rho(Z)$  设置为行- $l_0$  准范数(row- $l_0$  quasi-norm)的松弛版:

$$\rho(Z) = \|Z\|_{\text{rx}} = \sum_{i=1}^K \left\{ \max_{1 \leq j \leq P} |z_{ij}| \right\} \quad (17)$$

式中,  $\|\mathbf{Z}\|_{\text{rx}}$  范数是把矩阵  $\mathbf{Z}$  每一行中的最大绝对值项相加。本质上, 将  $l_{\infty}$  范数应用于  $\mathbf{Z}$  的行以促进它的非稀疏性, 而将  $l_1$  范数应用于所得向量  $\mathbf{z}_i$  以促进其稀疏性。若矩阵  $\mathbf{Z}$  退化成一个列向量, 则松弛范数降为  $l_1$  范数, 此时 SSA 的松弛条件与简单稀疏近似的松弛条件相同。

把式 (17) 代入式 (16) 中, 得到信号矩阵的降噪优化问题:

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{D}\mathbf{Z} - \mathbf{Y}\|_F^2 + \lambda \|\mathbf{Z}\|_{\text{rx}} \quad (18)$$

同样地, 从优化式 (12) 可知, 函数  $\rho(\mathbf{Z})$  是矩阵  $\mathbf{Z}$  的  $l_1$  范数, 本文对  $\rho(\mathbf{Z})$  的  $l_1$  范数使用偶对称的函数:

$$\rho(\mathbf{Z}) = \|\mathbf{Z}\|_1 = \sum_{i=1}^K \sum_{j=1}^P \{|z_{ij}|\} \quad (19)$$

式中,  $z_{ij}$  是矩阵  $\mathbf{Z}$  的第  $(i, j)$  个元素。另外, 式 (12) 是将两个列向量推广到多个列向量的情形, 即式 (12) 中的系数矩阵一般是  $P$  个列向量组成的矩阵  $\mathbf{Z} \in \mathbf{R}^{K \times P}$ , 而不仅仅是由 2 个列向量组成的矩阵  $\mathbf{Z} \in \mathbf{R}^{K \times 2}$ 。

为方便起见, 把  $\|\mathbf{Z}\|_{\text{rx}}$  称为矩阵  $\mathbf{Z}$  的 rx-范数。求解式 (18) 的算法称为 rx-范数的 Joint-PCD, 求解式 (12) 的算法称为  $l_1$  范数的 Joint-PCD。在给定系数矩阵的当前解  $\mathbf{Z}_k$  (其中  $k$  是迭代次数) 条件下, 定义矩阵  $\mathbf{U}(\mathbf{D}, \mathbf{Y}, \mathbf{Z}_k)$  为:

$$\mathbf{U}(\mathbf{D}, \mathbf{Y}, \mathbf{Z}_k) = \text{diag}^{-1}\{\mathbf{D}^T \mathbf{D}\} \mathbf{D}^T (\mathbf{Y} - \mathbf{D}\mathbf{Z}_k) + \mathbf{Z}_k \quad (20)$$

考虑到超完备字典  $\mathbf{D}$  的列是归一化的, 即  $\text{diag}^{-1}(\mathbf{D}^T \mathbf{D}) = \mathbf{I}$ , 其中  $\mathbf{I}$  是单位矩阵, 则可以将式 (20) 简化为:

$$\mathbf{U}(\mathbf{D}, \mathbf{Y}, \mathbf{Z}_k) = \mathbf{D}^T (\mathbf{Y} - \mathbf{D}\mathbf{Z}_k) + \mathbf{Z}_k \quad (21)$$

矩阵  $\mathbf{Z}$  的更新迭代值  $\mathbf{Z}_{k+1}$  为:

$$\mathbf{Z}_{k+1} = \mathbf{Z}_k + \mu \mathbf{Q}_k \quad (22)$$

且

$$\mathbf{Q}_k = S[\mathbf{U}(\mathbf{D}, \mathbf{Y}, \mathbf{Z}_k), \lambda] - \mathbf{Z}_k \quad (23)$$

式中,  $\mu$  是迭代步长;  $\lambda$  是收缩函数  $S(\cdot)$  的阈值; 从启发式收缩算法<sup>[11]</sup>的推导过程可知, 式 (23) 中的收缩函数  $S(\cdot)$  是一个软阈值函数<sup>[11]</sup>, 其定义为:

$$S(z, \theta) = \begin{cases} z + \theta & (z < -\theta) \\ 0 & (|z| \leq \theta) \\ z - \theta & (z > \theta) \end{cases} \quad (24)$$

式中,  $\theta$  为软收缩函数  $S(\cdot)$  的阈值。

式 (22) 中的步长参数  $\mu$  对算法的性能影响很大; 对足够小的  $\mu > 0$ , 步长必须导致惩罚函数  $\rho(\cdot)$  的可行下降。本文使用线搜索<sup>[10]</sup>寻找最佳  $\mu$  值。线搜索作为一维优化寻求迭代步长, 用于解决以下的优化问题:

$$\min_{\mu} \frac{1}{2} \|\mathbf{D}(\mathbf{Z}_k + \mu \mathbf{Q}_k) - \mathbf{Y}\|_F^2 + \lambda \rho(\mathbf{Z}_k + \mu \mathbf{Q}_k) \quad (25)$$

式中,  $\rho(\cdot)$  可以是 rx-范数, 也可以是  $l_1$ -范数。

如果信号矩阵  $\mathbf{Y}$  退化为一个列向量  $\mathbf{y}$ , 系数矩阵  $\mathbf{Z}$  也仅仅是一个列向量  $\mathbf{z}$ , 则 Joint-PCD 算法将退化为 PCD 算法。因此, 将 Joint-PCD 称为一种启发式算法。

在音频信号处理中, 经常使用两个超完备字典<sup>[16]</sup>, 即离散余弦变换 (DCT) 字典和 Gabor 字典。

一个加窗的 Gabor 字典为  $\mathbf{D}^G = [\mathbf{d}_{(k, \phi)}]_{(k, \phi) \in \Psi}$ , 其原子以连续的参数集  $\Psi = [1, K] \times [0, 2\pi]$  为索引, 其定义为:

$$\mathbf{d}_{k, \phi}(t) = w_d(t) \cos \left[ \frac{\pi}{K} \left( t + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) + \phi \right] \quad (26)$$

式中,  $1 \leq k \leq K$ ,  $0 \leq t \leq N-1$ ,  $K$  是 Gabor 字典的大小;  $\phi$  是相位;  $w_d$  是加权窗口。 $\mathbf{d}_{(k, \phi)}$  的分解可以用离散字典中的原子对表示:

$$\mathbf{d}_k^S(t) = w_d(t) \sin \left[ \frac{\pi}{K} \left( t + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \right] \quad (27)$$

$$\mathbf{d}_k^C(t) = w_d(t) \cos \left[ \frac{\pi}{K} \left( t + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \right] \quad (28)$$

以上两式中, 成对的原子是相同频率且相位为零的余弦和正弦对。为简化计算, 词典的列都做归一化处理, 即 Gabor 字典的正弦和余弦原子都使用其单位范数的版本。

综上所述, 给出 Joint-PCD 算法如下:

任务: 通过  $\hat{\mathbf{X}} = \arg \min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{D}\mathbf{Z} - \mathbf{Y}\|_F^2 + \lambda \mathbf{1}^T \rho(\mathbf{Z})$  对信号  $\mathbf{Y}$  降噪

输入: 观测矩阵  $\mathbf{Y}$ ; 字典  $\mathbf{D}$ ; 阈值  $\lambda$ ; 函数  $\rho(\cdot)$ ; 迭代次数  $M$

开始: 设置  $k=1$ , 迭代开始

If:  $k < M$

误差: 计算  $\mathbf{E} = \mathbf{Y} - \mathbf{D}\mathbf{Z}_k$

映射: 计算  $\mathbf{E}_T = \mathbf{D}^T \mathbf{E}$

收缩: 计算  $\mathbf{E}_S = S(\mathbf{E}_T + \mathbf{Z}_k, \lambda)$

线收缩: 寻找  $\mu_0$  以获得  $\mathbf{Z}_k + \mu(\mathbf{E}_S - \mathbf{Z}_k)$  的下降

放松: 更新迭代  $Z_{k+1} = Z_k + \mu_0(E_S - Z_k)$

Next:  $k \leftarrow k+1$

End 降噪后的输出:  $\hat{X} = DZ_M$

### 3 仿真实验及结果分析

#### 3.1 仿真环境和性能指标

通过仿真对矩阵形式的 Joint-PCD 与向量形式的 PCD 算法的性能进行全面比较, 各算法的仿真实验环境完全相同。

测试用的 5 个音频音乐信源取自著名的音频网站<sup>[17]</sup>。源信号类型为 wav, 采样率为 44 100 Hz。每个信源的样本数量为  $L=2^{16}=65\ 536$ 。5 个音频信源矩阵为  $X=[x_1, x_2, x_3, x_4, x_5]^T$ , 其时域的波形如图 1 所示。

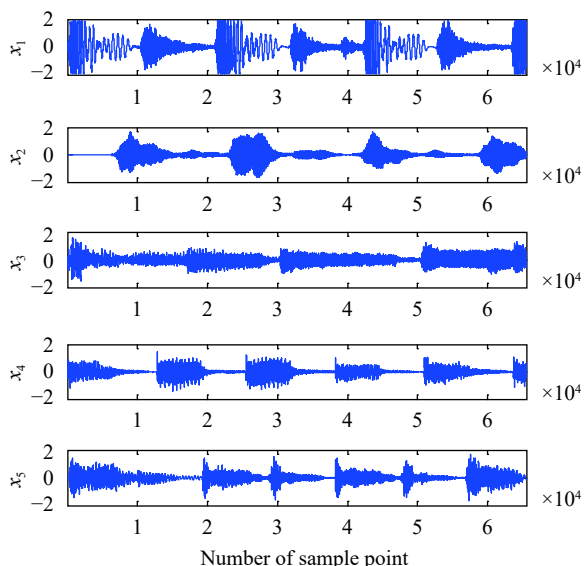


图 1 5 路音乐音频源信号的波形

仿真实验的硬件平台: 笔记本电脑, 其 CPU 为 Intel (R) Celeron (R) 1007U CPU-1.50 GHz, 内存为 4 GB, PC 机操作系统为 Windows10。软件平台为 MATLAB9 (R2016a)。

降噪性能通过信噪比 (signal to noise ratio, SNR) 和均方根误差 (root mean square error, RMSE) 来衡量:

$$\text{SNR} = 10 \lg \frac{\|x\|_2^2}{\|x - \hat{x}\|_2^2} \quad (29)$$

$$\text{RMSE} = \sqrt{\frac{1}{L} \|x - \hat{x}\|_2^2} \quad (30)$$

式中,  $x$  为干净信号;  $\hat{x}$  为降噪后信号;  $L$  为  $x$  的样本数。

#### 3.2 参数设置及仿真过程

对信源  $X$  施加一定强度的高斯白噪声后形成含噪声的观测信号  $Y$ , 噪声强度一般采用高斯随机变量的标准差  $\sigma$  来度量。显然, 信号  $Y$  中噪声强度的大小对降噪算法的性能带来很大影响。另外, 在 PCD 和 Joint-PCD 算法中, 软收缩函数  $S(\cdot)$  的阈值  $\lambda$  以及所采用的超完备字典 (DCT 或者 Gabor) 也都会对降噪性能造成一定影响。这几个参数的最佳值分别为: 噪声强度  $\sigma=0.1$ ; 软阈值  $\lambda=0.1$ ; 选用 Gabor 字典, 其尺寸为  $N \times K=64 \times 1\ 024$ , 加权窗为  $N=64$  的对称正弦窗。

在 PCD 和 Join-PCD 算法中, 惩罚函数  $\rho(\cdot)$  的形式、算法迭代步长  $\mu$  也都需确定。不同形式的  $\rho(\cdot)$  使用不同的矩阵 (向量) 范数。对于 PCD, 采用向量的  $l_1$  范数来求解优化式 (7); 对于 Joint-PCD, 使用矩阵的  $l_1$  范数来求解优化式 (12), 使用矩阵的  $rx$ -范数来求解优化式 (18)。对于 PCD 和 Joint-PCD 算法的迭代步长  $\mu (0 < \mu < 1)$ , 在算法的运行过程中使用线搜索来获得最佳值, 即在每次迭代中, 用 MATLAB 函数 fminbnd 在线搜索  $\mu$  值作为更新的步长。

音频信源与高斯噪声相加生成含噪声信号, 其降噪仿真过程如下。首先, 每个含噪声 (时域) 信号乘以长度为 64 的汉明窗生成语音帧, 并使相邻两帧之间的重叠样本数为 32, 即每帧的前后各有 16 个样本与相邻帧的样本重叠, 分帧使用 Voicebox<sup>[18]</sup> 中的 MATLAB 函数 enframe 实现。然后, 用 PCD 和 Joint-PCD 算法分别对分割后的含噪信号实施降噪。执行 PCD 是对音频信号的一帧进行降噪; Joint-PCD 执行一次是对一个信号 (而不仅仅是一帧) 进行降噪。最后, 重建降噪后的信号, 经过降噪的音频信号乘以逆汉明窗, 在每一帧与相邻两帧重叠的样本中, 分别在两边各去除 16 个样本, 仅保留每一帧中间部分的样本; 将经过处理的各个音频帧简单串联起来即可得到重新合成的音频信号。

#### 3.3 性能分析

对 PCD 和 Joint-PCD 算法的性能分析从两个方面进行: 1) 算法在相同条件下所消耗的运行时间成本; 2) 算法降噪性能指标 (SNR 和 RMSE 值) 的比较。

为了全面分析 PCD 和 Joint-PCD 算法的性能, 在第一个仿真中, 设置算法的最大迭代次数  $M=50$ 。为便于表示, 把 Joint-PCD 两种范数对应

的算法分别表示为 Joint-PCD with rx norm 和 Joint-PCD with  $l_1$  norm。算法 Joint-PCD 和 PCD 的运行时间指标如图 2 所示。

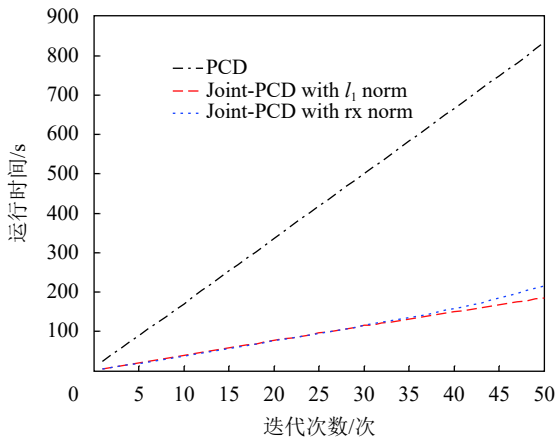


图 2 算法的运行时间比较

从图 2 可以看到, 随着迭代次数的增加, 算法的运行时间几乎都是呈线性增长, 但不同算法对应的直线斜率 (增长率) 是不同的。对于 Joint-PCD with rx norm 和 Joint-PCD with  $l_1$  norm 算法, 随着迭代次数的增加, 它们的运行时间增长率几乎是相同的; PCD 算法运行时间增长率却远大于 Joint-PCD 的增长率。显然, 与 PCD 算法相比较, Joint-PCD 算法能够极大地降低运行时间。

为了便于比较 PCD 和 Joint-PCD 的降噪性能, 在每次迭代后, 分别计算每种算法对每个信号的降噪指标值。图 3 给出了 PCD 算法的输出 SNR 和 RMSE 波形。

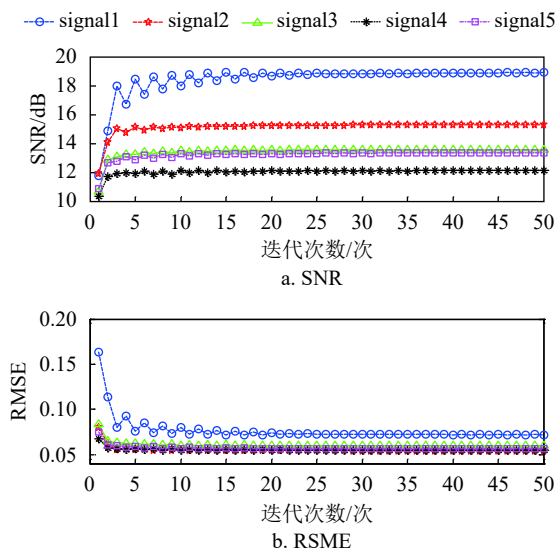


图 3 PCD 算法对每个信号的输出 SNR 和 RMSE 指标

同样地, 图 4 和图 5 分别给出了 Joint-PCD with

$l_1$  norm 和 Joint-PCD with rx norm 算法的输出 SNR 和 RMSE 波形。

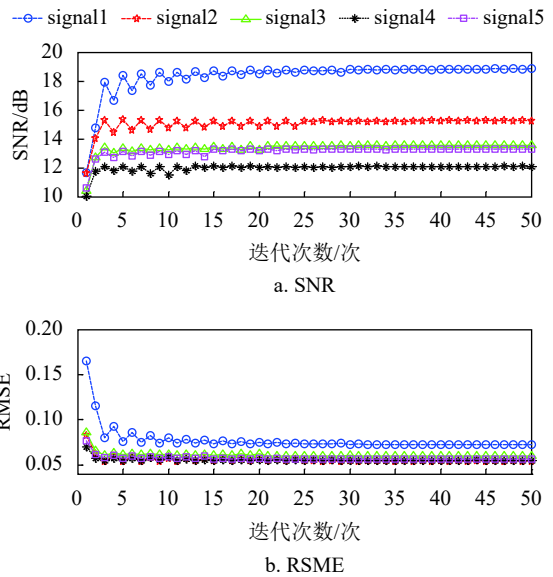


图 4 Joint-PCD with  $l_1$  norm 算法对每个信号的降噪指标

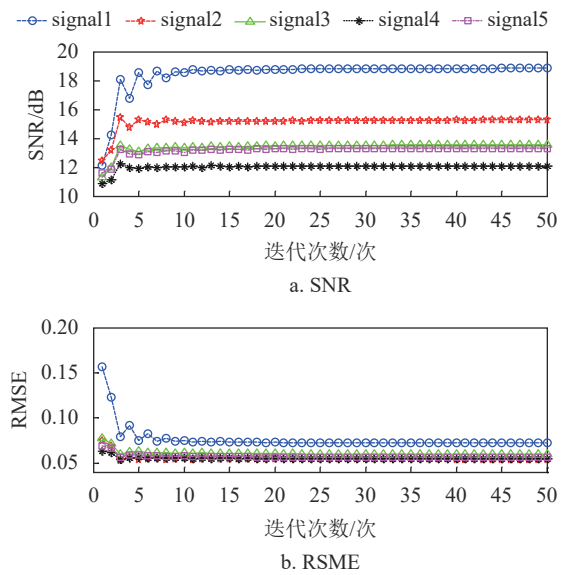


图 5 Joint-PCD with rx norm 算法对每个信号的降噪指标

比较图 3、图 4 和图 5 可知, 对于每个信号, PCD 算法的降噪性能几乎与 Joint-PCD with rx norm 和 Joint-PCD with  $l_1$  norm 算法的性能完全相同。经过 25 次迭代, PCD 和 Joint-PCD 算法对每个信号的平均输出 SNR 和 RMSE 值都逐渐收敛到其稳定值。这表明从降噪性能的角度来看, 用 Joint-PCD 算法代替 PCD 算法是完全可行的。

为了更直观地观察 PCD 和 Joint-PCD 算法的平均降噪性能, 在每次迭代后, 计算并记录了每种算法对于 5 个信号的平均 SNR 值。其结果如图 6 所示。

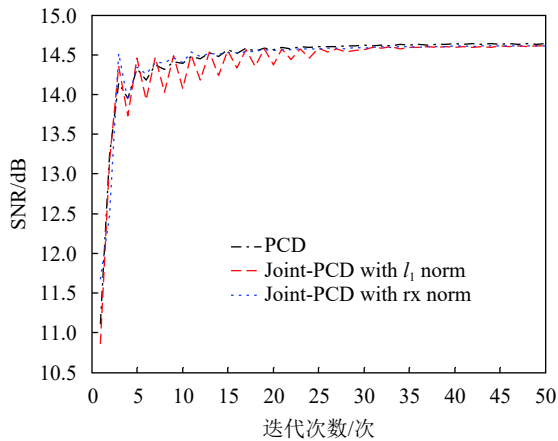


图 6 5 个信号的平均 SNR 指标

同样地, PCD 和 Joint-PCD 算法对 5 个信号的平均降噪性能指标 RMSE 的值如图 7 所示。

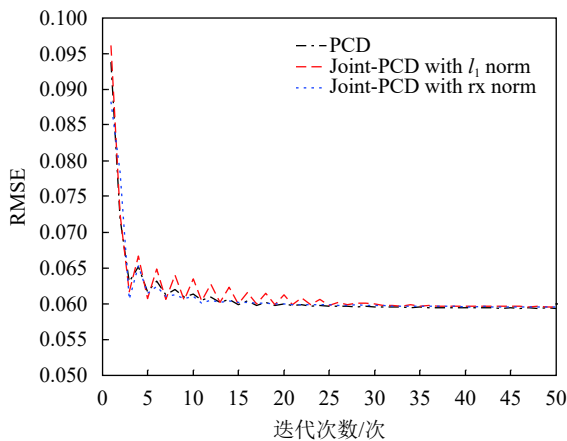


图 7 5 个信号的平均 RMSE 指标

从图 6 和图 7 可以看出, PCD 和 Joint-PCD 算法的平均性能指标随着迭代次数的增加具有一定的波动, 特别是在算法的初始迭代阶段, 其波动更为明显。这是因为每个信源的幅度值不同, 而施加到各个信源上的高斯白噪声的强度却一样, 这就造成每个信号受到噪声的干扰程度不同。另外, 在降噪过程中, 由于每个信号被噪声污染的严重性不一

样, 它们输出的 SNR 和 RMSE 值 (如图 3~图 5 所示) 就具有不同的波动范围。当把这 5 个信号的降噪性能平均之后, 其对应的平均 SNR 和 RMSE 值有波动就是一种很自然的现象, 不可能消除这种平均性能的波动。将 5 个信号的降噪性能进行平均是为了研究算法稳定收敛的条件。从图 6 和图 7 还可看出, 对于 PCD 和 Joint-PCD with rx norm 算法, 平均 SNR 和 RMSE 值的波动在大约 20 次迭代后结束; 对于 Joint-PCD with  $l_1$  norm 算法, 其波动在大约 25 次迭代后结束。这表明 Joint-PCD 和 PCD 算法在经过大约 25 次迭代后能稳定收敛。

为了更精细地分析不同算法的降噪性能, 通过第二个仿真分析 PCD 和 Joint-PCD 算法的差异性。从上述的结果可知, PCD 和 Joint-PCD 算法在经过 25 次迭代后能稳定收敛, 因此, 在这个仿真中, 将算法的最大迭代次数设置为  $M=25$ 。比较 PCD 和 Joint-PCD 算法达到稳定收敛所需要的运行时间, 其结果如表 1 所示。

表 1 PCD 和 Joint-PCD 算法的运行时间

算法	PCD	Joint-PCD with rx-norm	Joint-PCD with $l_1$ norm	s
运行时间	432.197 9	94.284 8	<b>93.905 6</b>	

从表 1 可以看出, Joint-PCD with  $l_1$  norm 算法的运行时间最少, 而 Joint-PCD with rx norm 算法的运行时间比前者仅多了 0.379 2 s, 因此, 两种 Joint-PCD 算法的运行时间几乎相同。与之相反, PCD 算法收敛所花费的运行时间大约是 Joint-PCD 算法的 4.6 倍。这充分说明 Joint-PCD 算法比 PCD 算法具有更高的计算效率。

对于 PCD 和 Joint-PCD 算法的降噪性能, 采用每种算法对每个信号在整个迭代过程中输出 SNR 和 RMSE 的平均值来度量, 其结果如表 2 所示。表中粗体数字为相应的最佳降噪性能指标。

表 2 每个信号在整个迭代过程的平均 SNR 和 RMSE 值

信号目录	PCD		Joint-PCD with rx-norm		Joint-PCD with $l_1$ norm	
	SNR/dB	RMSE	SNR/dB	RMSE	SNR/dB	RMSE
signal1	<b>18.8492</b>	<b>0.0722</b>	18.8164	0.0725	18.7463	0.0731
signal2	15.2442	<b>0.0544</b>	15.2131	0.0546	<b>15.2482</b>	<b>0.0544</b>
signal3	<b>13.5420</b>	<b>0.0598</b>	13.5103	0.0601	13.5222	0.0600
signal4	12.0844	0.0551	12.0771	0.0551	<b>12.0948</b>	<b>0.0550</b>
signal5	13.2911	0.0565	13.2611	0.0567	<b>13.3069</b>	<b>0.0564</b>

从表 2 中的数据可以看出, 对于第 2、第 4 和第 5 个信号, Joint-PCD with  $l_1$  norm 算法的平均

SNR 和 RMSE 值是最优的; 对于第 1 和第 3 个信号, PCD 算法的平均指标是最优的。另外, 从表 2

还可以看出, PCD 和 Joint-PCD 算法平均 SNR 值的差异仅发生在小数点后面第二位; 而平均 RMSE 值的差异仅发生在小数点后面第四位。这表明, 用 Joint-PCD 代替 PCD, 对音频信号降噪性能几乎不产生任何影响。

综合表 1 和表 2 的结果可知, 在降噪性能基本相同的情况下, Joint-PCD 算法的计算复杂度大幅度降低。

## 4 结束语

本文定义了一种音频信号的时域处理框架, 即将每个被分割的音频帧都作为一个列向量从而生成信号矩阵。在联合稀疏表示 (JSR) 和同时稀疏近似 (SSA) 的基础上, 利用新的处理框架, 提出了以信号矩阵为操作对象的 Joint-PCD 算法。仿真实验的结果表明: Joint-PCD 算法不仅能够提供与向量形式的 PCD 算法几乎完全相同的降噪性能, 更重要的是, Joint-PCD 算法提供了更高的计算效率。与传统的 PCD 算法相比, Joint-PCD 算法在大规模数据处理和实时信号处理应用方面具有巨大的潜在优势。

## 参 考 文 献

- [1] MURATA N, KOYAMA S, TAKAMUNE N, et al. Sparse representation using multidimensional mixed-norm penalty with application to sound field decomposition[J]. *IEEE Transactions on Signal Processing*, 2018, 66(12): 3327-3338.
- [2] RAPHAN M, SIMONCELLI E. Optimal denoising in redundant representations[J]. *IEEE Transactions on Image Processing*, 2008, 17(8): 1342-1352.
- [3] LI W, ZHOU Y, POH N, et al. Feature denoising using joint sparse representation for in-car speech recognition[J]. *IEEE Signal Processing Letters*, 2013, 20(7): 681-684.
- [4] CHEN W, WIPF D, WANG Y, et al. Simultaneous Bayesian sparse approximation with structured sparse models[J]. *IEEE Transactions on Signal Processing*, 2016, 64(23): 6145-6159.
- [5] ISOGAWA K, IDA T, SHIODERA T, et al. Deep shrinkage convolutional neural network for adaptive noise reduction[J]. *IEEE Signal Processing Letters*, 2018, 25(2): 224-228.
- [6] DONOHO D L, JOHNSTONE I M. Ideal spatial adaptation via wavelet shrinkage[J]. *Biometrika*, 1994, 81(3): 425-455.
- [7] GOLDSTEIN T, STUDER C. PhaseMax: Convex phase retrieval via basis pursuit[J]. *IEEE Transactions on Information Theory*, 2018, 64(4): 2675-2689.
- [8] ZIBULEVSKY M, ELAD M.  $L_1$ - $L_2$  optimization in signal and image processing[J]. *IEEE Signal Processing Magazine*, 2010, 27(3): 76-88.
- [9] SUN J, SUN J, XU Z. Color image denoising via discriminatively learned iterative shrinkage[J]. *IEEE Transactions on Image Processing*, 2015, 24(11): 4148-4159.
- [10] ZIBETTI M V W, HELOU E S, PIPA D R. Accelerating overrelaxed and monotone fast iterative shrinkage-thresholding algorithms with line search for sparse reconstructions[J]. *IEEE Transactions on Image Processing*, 2017, 26(7): 3569-3578.
- [11] ELAD M. Why simple shrinkage is still relevant for redundant representations?[J]. *IEEE Transactions on Information Theory*, 2006, 52(12): 5559-5569.
- [12] ELAD M, MATALON B, ZIBULEVSKY M. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization[J]. *Applied and Computational Harmonic Analysis*, 2007, 23: 346-367.
- [13] BABAIE-ZADEH M, JUTTEN C. On the stable recovery of the sparsest overcomplete representations in presence of noise[J]. *IEEE Transactions on Signal Processing*, 2010, 58(10): 5396-5400.
- [14] CHEN J, HUO X. Theoretical results on sparse representations of multiple-measurement vectors[J]. *IEEE Transactions on Signal Processing*, 2006, 54(12): 4634-4643.
- [15] WIPF D P, RAO B D. An empirical Bayesian strategy for solving the simultaneous sparse approximation problem[J]. *IEEE Transactions Signal Processing*, 2007, 55(7): 3704-3716.
- [16] ADLER A, EMIYA V, JAFARI M G, et al. Audio inpainting[J]. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012, 20(3): 922-932.
- [17] Collaborative database of creative-commons licensed sound for musicians and sound lovers[EB/OL]. [2019-11-12]. <http://www.freesound.org/>.
- [18] HYPERLINK. VOICEBOX is a speech (and audio) processing toolbox consists of MATLAB routines that are maintained[EB/OL]. [2019-09-11]. <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>.

编辑 税 红