

矩阵广义逆硬阈值追踪算法与 稀疏恢复问题^{*1)}

施章磊 李维国

(中国石油大学(华东)理学院, 山东青岛 266580)

摘要

本文通过引入支撑集捕获基数及 MP 广义逆, 提出了一种用于稀疏恢复问题的矩阵广义逆硬阈值追踪算法, 并在观测误差存在的情况下给出了算法在约束等距条件 (RIP) 下的收敛性. 数值实验表明, 算法不仅极大地减少了收敛所需迭代次数, 且观测误差存在的情况下稀疏恢复是强健的.

关键词: 压缩感知; 稀疏恢复; 硬阈值追踪; 广义逆

MR (2000) 主题分类: 65F10, 15A29, 94A12

1. 引言

压缩感知是一种新兴低采样率信号获得理论, 该理论通过线性编码的方式将 N 维稀疏信号压缩为 m 维信号 ($m \ll N$). 令 $y \in R^m$ 是一个已知的向量, A 是 $R^{m \times N}$ 矩阵且 ($m \ll N$). A 即字典且其列向量称为 A 的原子. 这篇文章想要处理的问题是寻找一个 N 维向量 x 满足如下关系:

$$y = Ax + e, \quad (1.1)$$

其中 $e \in R^m$ 称为观测误差. 若 $e \neq 0$, 则 (1.1) 式的求解为对 x 的近似; 同时若 $e = 0$, 则为 x 的精确表示. 因为 $m \ll N$, 故存在无数满足上述方程 (1.1) 的解 x . 通常的做法是寻找满足一定稀疏度量的解 x , 例如令解 x 具有最少个数的非零元素, 也即求解极小化 l_0 问题:

$$\min_{x \in R^N} \|x\|_0 \quad s.t. \quad y = Ax + e, \quad (1.2)$$

其中伪范数 $\|x\|_0$ 表示 x 分量中非零元素的个数.

然而通常情况下问题 (1.2) 是 NP- 难问题^[1], 许多文献^[2-4] 建议将该问题 (1.2) 凸松弛后求解. 基追踪问题 (极小化 l_1 问题) 作为一个更可行的模型由问题 (1.2) 凸松弛后得到:

$$\min_{x \in R^N} \|x\|_1 \quad s.t. \quad y = Ax + e, \quad (1.3)$$

其中 $\|x\|_1 = \sum_i |x_i|$. 理论上已经证明, 在某些确定的条件下, 极小化 l_0 问题 (1.2) 等价于极小化 l_1 问题 (1.3).

实际中不同的算法用到不同的策略来求解极小化 l_0 问题 (1.2), 极小化 l_1 问题 (1.3). 贪婪迭代算法通过捕获 x 的支撑集, 然后寻找在所捕获支撑集下适合欠定线性系统且最稀疏的

* 2016年5月30日收到.

¹⁾ 基金项目: 中央高校基本科研业务费 (15CX05051A, 15CX02055A, 16CX02011A); 山东省自然科学基金 (ZR2014AQ004); 山东省优秀中青年科学家科研奖励基金 (2014BSE28027) 资助项目.

x 来直接求解极小化 l_0 问题, 包括正交匹配追踪 (Orthogonal Matching Pursuit, OMP) [5], 压缩采样匹配追踪 (Compressive Sampling Matching Pursuit, CoSaMP) [6], 广义正交匹配追踪 (Generalized Orthogonal Matching Pursuit, gOMP) [7] 等. 同时需要指出的是迭代硬阈值算法 (Iterative Hard Thresholding Algorithm, IHT) [8] 也可以接求解极小化 l_0 问题但不属于贪婪迭代算法. IHT 展现了它的优越性, 因为它充分的利用了和 l_0 范数相关的硬阈值算子. 此外求解极小化 l_1 范数的算法包含迭代软阈值算法 (Iterative Soft Thresholding Algorithm, IST) [9], Bregman 迭代算法 (Bregman Iteration Algorithm) [10] 及其相关算法 [11] 等.

最近, Simon Foucart 提出了一种贪婪迭代策略和迭代硬阈值策略的混合算法硬阈值追踪算法 (Hard Thresholding Pursuit Algorithm, HTP) [12] 及其变体分级硬阈值追踪算法 (Graded Hard Thresholding Pursuit Algorithm, GHTP) [13]. 这两种算法具备较强的理论保证, 展现出了强大的稀疏恢复能力. 特别指出的是在进行 k -稀疏的信号重建时, GHTP 几乎在第 k 次迭代时完成捕获信号正确支撑集的任务, 并且所需的平均迭代数也几乎与信号稀疏度相同. 这说明 GHTP 收敛所需的迭代数与信号稀疏度的成正比, 即 GHTP 收敛所需的迭代数会随信号稀疏度增加而增大. 其后 Li H 等人提出了广义硬阈值追踪算法 (Generalized Hard Thresholding Pursuit Algorithm, GHTP) [14] 以期加速硬阈值追踪, 使其收敛所需迭代数大大减少. 这篇文章中提出一种新的矩阵广义逆硬阈值追踪算法 (Matrix Pseudoinverse Hard Thresholding Pursuit Algorithm, MPHTP), 该算法不仅具备较强的稀疏恢复能力且通过加速减少了收敛所需迭代数.

本文中我们所用到的记号如下:

- $\|x\|$ 表示 $\|x\|_2$, 除非有其它标注; 对应的 $|x|$ 表示 x 的绝对值, 若 x 为向量则为逐元作用;
- $A^\dagger = A^*(AA^*)^{-1}$ 表示矩阵 A 的 MP 广义逆, A^* 表示矩阵 A 的转置;
- $\text{supp}(x)$ 为向量 x 的支撑集; 有时为简便也用 S 表示 $|x|$ 的 k 个最大值对应指标集组成的支撑集;
- x_T 为向量 x 限制在支撑集 T 上的子向量, 同样适用于 A_T ;
- \bar{T} 为集合 T 在 $\{1, 2, \dots, N\}$ 中的余集, 即 $\bar{T} = \{1, 2, \dots, N\} \setminus T$;
- $T \Delta T'$ 为集合 T 和 T' 的对称差集, 即 $T \Delta T' = (T \setminus T') \cup (T' \setminus T)$.

2. HTP、MPHTP 算法框架

所有的贪婪迭代算法都致力于从 $A^*(y - Ax^n)$ 出发捕获 x 的支撑集, 之后寻找具备此支撑集且适合测量值的向量 x . 但是从 IHT 中受到启发, 似乎在某些确定的条件下从 $x^n + A^*(y - Ax^n) \approx x$ 寻找 x 的支撑集更为合适. 这一启发结合贪婪搜索算法的思想造就了如下的硬阈值追踪算法. 给定初始 k -稀疏向量 $x^0 \in C^N$, 一般情况下 $x^0 = \mathbf{0}$, 按照如下格式迭代直到满足一定的停机准则, 最后输出向量 x 的估计向量.

HTP 良好的理论保证了其较好的表现. 当矩阵 A 的约束等距常数满足 $\delta_{3k} < 1/\sqrt{3}$ 时, 硬阈值追踪可以在有限次迭代后找出线性系统所有的 k -稀疏解 [12]. 此处我们给出矩阵 $A \in R^{m \times N}$ 的 k 阶约束等距常数定义, 即对于所有的 k -稀疏向量 $z \in R^N$, 存在满足下式

$$(1 - \delta_k)\|z\|_2^2 \leq \|Az\|_2^2 \leq (1 + \delta_k)\|z\|_2^2 \quad (2.1)$$

的最小常数 $0 \leq \delta_k \leq 1$, 同时称矩阵 A 满足 RIP 性质. 此时矩阵 A 在 k -稀疏的信号集合上

它的原子是近似正交的. 若矩阵 A 的 $2k$ 阶约束等距常数存在, 则 (1.3) 式可以恢复任意 k -稀疏信号.

算法 1 HTP

参数: $A, y, k, threshold > 0$

初始化: $x^0 = \mathbf{0}, \mathbf{r}_0 = \mathbf{y}, \mathbf{S}^0 = \emptyset, \mathbf{n} = \mathbf{0}$

while: $\|\mathbf{r}_n\|_2 > threshold$ **do**

(HTP₁) 1. 计算 $S^{n+1} = \{|x^n + A^*(y - Ax^n)|$ 中前 k 个最大值对应的指标}

(HTP₂) 2. 估计 $x^{n+1} = \operatorname{argmin}\{\|y - Az\|_2, \operatorname{supp}(z) \subseteq S^{n+1}\}$

(HTP₃) 3. 更新 $r_{n+1} = y - Ax^{n+1}, n = n + 1$

end while

return x^{n+1}

受硬阈值追踪算法基本思想的启发, 所构建的新算法从 $x^n + A^\dagger(y - Ax^n) \approx A^\dagger y$ 中选取支撑集; 引入 MP 广义逆以及支撑集捕获基数 N_0 , 构建与硬阈值追踪类似的矩阵广义逆硬阈值追踪 (Matrix Pseudoinverse Hard Thresholding Pursuit Algorithm, MPHTP). 同时, 该算法不需要稀疏度的先验估计且在该框架下可以减少算法收敛所需的迭代数. 值得注意的一点是矩阵广义逆硬阈值追踪与广义正交匹配追踪算法有一定的相似性 (从支撑集选取数目角度), 故支撑集捕获基数 $N_0 \leq \sqrt{m}$ 且不能超过稀疏度 k , 也即 $N_0 \leq \min\{\sqrt{m}, k\}$ [7]. 在给出我们的算法之前, 需要给出以下定理及讨论来说明该算法用于信号恢复时从 $x^n + A^\dagger(y - Ax^n) \approx A^\dagger y$ 中选取支撑集的合理性.

定理 1. [15] 向量 $u \in R^N$ 是方程 $Ax = y$ 的最小二乘解, 即 $u = \operatorname{arg} \min_{x \in R^N} \|Ax - y\|$, 当且仅当 u 满足

$$A^\dagger Au = A^\dagger y.$$

上述定理 1 说明了 (1.2) 式与 $\min_{x \in R^N} \|x\|_0$ s.t. $A^\dagger Ax = A^\dagger y$ 之间的等价性, 间接表明支撑集从 $x^n + A^\dagger(y - Ax^n)$ 中选取是合理的. 此外 $A^\dagger A$ 的特征值为 1 和 0, 比 A^*A 更接近于正交阵, 也即 $A^\dagger A$ 更符合 RIP 条件. 同时, $A^\dagger A$ 具有较小的条件数, 当 A 的条件数较大时算法也是有效的. 另外格式 $A^\dagger(y - Ax^n)$ 比 $A^*(y - Ax^n)$ 更快更容易收敛到 0, 即 $x^n + A^\dagger(y - Ax^n)$ 可以快速地收敛到 x . 若近似求解 A^\dagger , $A^\dagger A$ 的特征值仍然接近 1 和 0 ($A^\dagger A$ 仍旧接近正交阵), 且具有较小的条件数, 对整个算法产生的影响不会很大. 与上述 HTP 算法迭代格式相似, 矩阵广义逆硬阈值追踪算法迭代格式如下.

算法 2 MPHTP

参数: $A, y, N_0, threshold > 0$

初始化: $x^0 = \mathbf{0}, \mathbf{r}_0 = \mathbf{y}, \mathbf{S}^0 = \emptyset, \mathbf{n} = \mathbf{0}$

while: $\|\mathbf{r}_n\|_2 > threshold$ **do**

(MPHTP₁) 1. 计算 $S^{n+1} = \{|x^n + A^\dagger(y - Ax^n)|$ 中前 $N_0(n+1)$ 个最大值对应的指标}

(MPHTP₂) 2. 估计 $x^{n+1} = \operatorname{argmin}\{\|y - Az\|_2, \operatorname{supp}(z) \subseteq S^{n+1}\}$

(MPHTP₃) 3. 更新 $r_{n+1} = y - Ax^{n+1}, n = n + 1$

end while

return x^{n+1}

3. 收敛性

本节主要收集了矩阵广义逆硬阈值追踪算法收敛所需的关键结论. 在证明该算法收敛性之前, 我们给出一个在之后证明中需要用到不等式^[12]. N 维向量 x 的支撑集为 $\text{supp}(x)$, S 是满足 $|\text{supp}(x) \cup S| \leq t$ 支撑集, t 为常数, 则:

$$\|((I - A^*A)x)_S\| \leq \delta_t \|x\|, \quad (3.1)$$

其中 I 为单位矩阵.

定理 2. 假设矩阵 $A \in R^{m \times N}$ 满足 k 阶 RIP 条件, 则

$$\|I - A^\dagger A\| \leq \frac{2\delta_k}{1 + \delta_k} \quad (3.2)$$

对于所有 k - 稀疏向量 x 成立.

证明. 矩阵 A 满足 RIP 条件, i.e., 对于任意 k - 稀疏向量 $x \in R^N$,

$$(1 - \delta_k)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \delta_k)\|x\|^2,$$

因此可得

$$(1 - \delta_k)\|x\|^2 \leq \|Ax\|^2 = \|AA^\dagger Ax\|^2 = \|A(A^\dagger Ax)\|^2 \leq (1 + \delta_k)\|A^\dagger Ax\|^2.$$

上式说明

$$\frac{1 - \delta_k}{1 + \delta_k} \|x\|^2 \leq \|A^\dagger Ax\|^2,$$

继续改写为向量内积形式, 则

$$0 \leq \frac{1 - \delta_k}{1 + \delta_k} x^* x \leq (A^\dagger Ax)^*(A^\dagger Ax) \leq x^* x,$$

其中 $(A^\dagger A)^*(A^\dagger A) = A^\dagger A$.

上式说明

$$x^* x - x^*(A^\dagger A)x \leq x^* x - \frac{1 - \delta_k}{1 + \delta_k} x^* x = \frac{2\delta_k}{1 + \delta_k} x^* x.$$

故综上所述可得

$$\|I - A^\dagger A\| \leq \frac{2\delta_k}{1 + \delta_k}.$$

下面我们开始证明算法的收敛性. 由于算法与 HTP 算法的相似性, 证明过程与 HTP 算法过程相似.

3.1. MPHTP₁ 的结果

若 $x \in R^N$ 是 k - 稀疏的, $y = Ax + e$ ($e \in C^m$ 为误差), x' 是 k' - 稀疏的, 且若 T 是从 $x' + A^\dagger(f - Ax')$ 中按模量大小选出的基数为 $t \geq k$ 的指标集, 则

$$\|x_T\| \leq 2\sqrt{2} \frac{\delta_{k+t+k'}}{1 + \delta_{k+t+k'}} \|x - x'\| + \sqrt{2} \|(A^\dagger e)_{T \Delta S}\|, \quad (3.3)$$

其中 $S = \text{supp}(x)$.

实际上由 $S := \text{supp}(x)$ 及 T 是从 $x' + A^\dagger(y - Ax')$ 中按模量大小选出的基数为 $t \geq k$ 的指标集, 我们注意到

$$\|(x' + A^\dagger A(x - x') + A^\dagger e)_T\|^2 \geq \|(x' + A^\dagger A(x - x') + A^\dagger e)_S\|^2,$$

不等号两边消去 $S \cap T$ 对应元素对上述不等式的贡献, 则

$$\|(x' + A^\dagger A(x - x') + A^\dagger e)_{T \setminus S}\| \geq \|(x' + A^\dagger A(x - x') + A^\dagger e)_{S \setminus T}\|.$$

不等式左端满足

$$\|(x' + A^\dagger A(x - x') + A^\dagger e)_{T \setminus S}\| = \|((A^\dagger A - I)(x - x') + A^\dagger e)_{T \setminus S}\|,$$

其中 $x_{T \setminus S} = \mathbf{0}$.

同时右端满足

$$\|(x' + A^\dagger A(x - x') + A^\dagger e)_{S \setminus T}\| \geq \|x_{S \setminus T}\| - \|(A^\dagger A - I)(x - x') + A^\dagger e)_{S \setminus T}\|.$$

显然

$$\begin{aligned} & \|x_{S \setminus T}\| \\ & \leq \|((A^\dagger A - I)(x - x') + A^\dagger e)_{T \setminus S}\| + \|((A^\dagger A - I)(x - x') + A^\dagger e)_{S \setminus T}\| \\ & \leq \sqrt{2} \|((A^\dagger A - I)(x - x') + A^\dagger e)_{T \Delta S}\| \\ & \leq \sqrt{2} (\|(A^\dagger A - I)(x - x')_{T \Delta S}\| + \|(A^\dagger e)_{T \Delta S}\|) \\ & \leq 2\sqrt{2} \frac{\delta_{k+t+k'}}{1 + \delta_{k+t+k'}} \|x - x'\| + \sqrt{2} \|(A^\dagger e)_{T \Delta S}\|, \end{aligned}$$

其中由 (3.2) 得最后一个不等式.

3.2. MPHTP₂ 的结果

若 $x \in R^N$ 是 k -稀疏的, $y = Ax + e$ ($e \in C^m$ 为误差向量), x' 是 k' -稀疏的, T 是基数为 t 的指标集, 且若 x' 是 $\|f - Az\|$ 满足 $\text{supp}(z) \subseteq T$ 的极小值解, 则

$$\|x - x'\| \leq \sqrt{\frac{1}{1 - 2\delta_{k+t}^2}} \|x_{\bar{T}}\| + \sqrt{\frac{2}{1 - 2\delta_{k+t}^2}} \|(A^* e)_T\|. \quad (3.4)$$

由 x' 定义可知 $(A^*(y - Ax'))_T = 0$, 因此

$$\begin{aligned} & \|(x - x')_T\|^2 \\ & = \langle (x - x')_T, (x - x')_T \rangle \\ & = \langle (x - x' - A^*(f - Ax'))_T, (x - x')_T \rangle \\ & \leq | \langle ((I - A^*A)(x - x'))_T, (x - x')_T \rangle | + | \langle (A^*e)_T, (x - x')_T \rangle | \\ & \leq \|((I - A^*A)(x - x'))_T\| \| (x - x')_T \| + \|(A^*e)_T\| \| (x - x')_T \| \\ & \leq \delta_{k+t} \|x - x'\| \| (x - x')_T \| + \|(A^*e)_T\| \| (x - x')_T \|, \end{aligned}$$

其中由 (3.1) 得最后一个不等式.

两边同除 $\|(x - x')_T\|$ 后可得

$$\|(x - x')_T\| \leq \delta_{k+t} \|x - x'\| + \|(A^*e)_T\|,$$

由上述不等式可得

$$\|x - x'\|^2 = \|(x - x')_T\|^2 + \|(x - x')_{\bar{T}}\|^2 \leq (\delta_{k+t} \|x - x'\| + \|(A^*e)_T\|)^2 + \|x_{\bar{T}}\|^2, \quad (3.5)$$

其中 $x'_{\bar{T}} = \mathbf{0}$.

(3.5) 式暗示着

$$\|x - x'\|^2 \leq 2(\delta_{k+t}^2 \|x - x'\|^2 + \|(A^*e)_T\|^2) + \|x_{\bar{T}}\|^2,$$

也即

$$(1 - 2\delta_{k+t}^2) \|x - x'\|^2 \leq 2\|(A^*e)_T\|^2 + \|x_{\bar{T}}\|^2,$$

故可知 3.4 成立.

定理 3. 令 $x \in R^N$ 为 k -稀疏向量, $y = Ax + e$ ($e \in C^m$ 是误差向量), 若 x^n 是由 MPHTP 生成的序列, N_0 是支撑集捕获基数, 则

$$\|x - x^n\| \leq \rho_p \|x - x^{n-1}\| + \tau_q \|e\|, \quad \forall n \geq 1, \quad (3.6)$$

其中 $\rho_p = \frac{2\delta_p}{1+\delta_p} \sqrt{\frac{2}{1-2\delta_p^2}}$, $\tau_q = \sqrt{\frac{2}{1-2\delta_q^2}} (\sqrt{1+\delta_q} + \frac{1}{\sqrt{1-\delta_q}})$, $p = k + (2n-1)N_0$, $q = k + nN_0$.

证明. 在 (3.3) 中令 $x' = x^{n-1}$, $T = S^n$, 需要注意的是此时 x' 的稀疏度为 $(n-1)N_0$, T 中元素个数为 nN_0 , 则可知

$$\|x_{S^n}\| \leq 2\sqrt{2} \frac{\delta_{k+(2n-1)N_0}}{1 + \delta_{k+(2n-1)N_0}} \|x - x^{n-1}\| + \sqrt{2} \|(A^\dagger e)_{S^n \cup S}\|.$$

相似的, 在 (3.4) 中令 $T = S^n$, $x' = x^n$ (此时 x' 的稀疏度为 nN_0 , T 中元素个数为 nN_0), 则有

$$\|x - x^n\| \leq \sqrt{\frac{1}{1 - 2\delta_{k+nN_0}^2}} \|x_{S^n}\| + \sqrt{\frac{2}{1 - 2\delta_{k+nN_0}^2}} \|(A^*e)_{S^n}\|.$$

联合上述两式, 容易得

$$\begin{aligned} & \|x - x^n\| \\ & \leq \frac{2\sqrt{2}\delta_{k+(2n-1)N_0}}{1 + \delta_{k+(2n-1)N_0}} \sqrt{\frac{1}{1 - 2\delta_{k+nN_0}^2}} \|x - x^{n-1}\| \\ & \quad + \sqrt{\frac{2}{1 - 2\delta_{k+nN_0}^2}} (\|(A^\dagger e)_{S^n \cup S}\| + \|(A^*e)_{S^n}\|) \end{aligned}$$

由 $\|(A^*e)_{S^n}\| \leq \sqrt{1 + \delta_{k+nN_0}} \|e\|$ 及 $\|(A^\dagger e)_{S^n \cup S}\| \leq \frac{1}{\sqrt{1 - \delta_{k+nN_0}}} \|e\|$ [6]:

$$\begin{aligned} & \|x - x^n\| \\ & \leq \frac{2\sqrt{2}\delta_{k+(2n-1)N_0}}{1 + \delta_{k+(2n-1)N_0}} \sqrt{\frac{1}{1 - 2\delta_{k+nN_0}^2}} \|x - x^{n-1}\| \\ & \quad + \sqrt{\frac{2}{1 - 2\delta_{k+nN_0}^2}} \left(\sqrt{1 + \delta_{k+nN_0}} + \frac{1}{\sqrt{1 - \delta_{k+nN_0}}} \right) \|e\|. \end{aligned}$$

4. 数值模拟结果

这部分给出了 IHT²⁾, CoSaMP, MPHTP 三种方法在有噪声以及无噪声情况下的数值结果. 我们从恢复成功率以及收敛所需迭代数两方面对上述三种算法进行了比较. 无噪声实验中所有方法的停机准则为 $\|y - Ax^n\| \leq 10^{-8}$ 或迭代数达到最大迭代数 500, 并且判断一次恢复是否成功的条件为 $\|x - x^n\| \leq 10^{-4}\|x\|$. 有噪声实验中, 所有方法的停机准则为 $\|y - Ax^n\| \leq \|\epsilon\|$ 或迭代数达到最大迭代数 500, 判断一次恢复是否成功的条件为 $\|x - x^n\| \leq 10^{-2}\|x\|$. 为了更好地体现支撑集捕获基数 N_0 的作用, 在每次实验开始前我们令 $N_0 = \min\{k, N_0\}$.

4.1. 无噪声数值结果

本小节主要给出的是 IHT, CoSaMP, MPHTP 在无噪声时的表现. 随机生成高斯矩阵 $A \in R^{m \times N}$ ($m = 200, N = 1000$) 作为观测矩阵、高斯向量 $x \in R^N$ 作为原信号. 同时稀疏度 k 变化范围为区间 $[0, 90]$, 步长为 2; 对于每一个稀疏度 k 生成 50 个高斯矩阵且对于每个矩阵生成 10 个高斯向量.

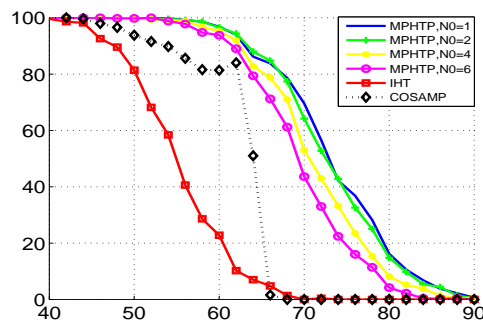


图 1 矩阵维数 $M=200, N=1000$ 时纯高斯, 无噪设置下不同算法的恢复成功率比较 (X 轴稀疏度变化区间; Y 轴恢复成功率)

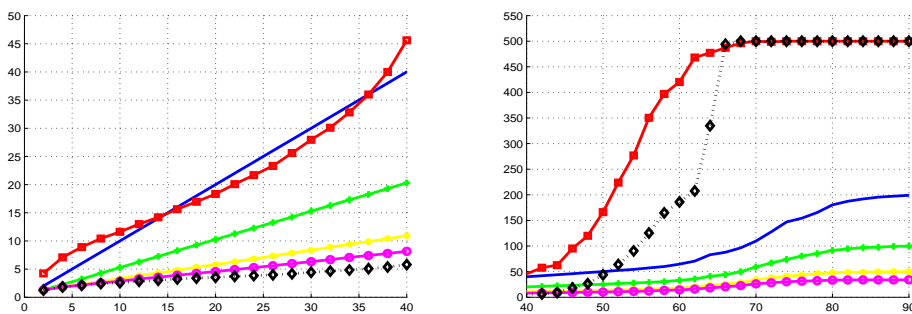


图 2 矩阵维数 $M=200, N=1000$ 时纯高斯, 无噪设置下不同算法所需平均迭代数: (左) 所有算法恢复成功率为 100% 区间的详细信息; (右) 恢复成功率小于 100% 区间的详细信息 (X 轴稀疏度变化区间; Y 轴平均迭代数)

²⁾ 此处 IHT 为修正后用于压缩感知的 IHT_k , 可在 Thomas Blumensath 个人主页 (<http://www.personal.soton.ac.uk/tb1m08/index.html>) 下载

图 (1) 描绘了纯高斯设置下不同算法的恢复成功率 (500 次实验的平均值), 其中包含了 $N_0 = 1, 2, 4, 6$ 的矩阵广义逆硬阈值追踪算法 (MPHTP). 我们可以看到在该设置下矩阵广义逆硬阈值追踪算法整体比 CoSaMP、IHT 表现的好, 其恢复成功率为 100% 的稀疏度区间是最大的 $([0, 55])$. 稀疏度 $k \in (0, 45)$ 时, 几乎所有的算法都可以以很高的概率 (100%) 恢复出原始信号. 稀疏度 $k \in (45, 60)$ 时, MPHTP 仍然保持很高的恢复成功率 (90% 以上), 相比之下 IHT 的恢复成功率则迅速下降、CoSaMP 同比也有所下跌. 当信号稀疏度 $k \in (70, 90)$ 时, MPHTP 也在大部分的情况下以相对较高的概率成功恢复原信号, 而另外两种算法恢复成功率为零. 我们也可以清楚地看到此时随稀疏度增大 MPHTP 算法成功率下降较缓.

同时图 (2) 中给出这几种算法收敛所需平均迭代数. 需要指出的是我们给出的是最差的情况, 也即在整个稀疏度变化区间上所有实验所需迭代数的平均值. 显然在恢复成功率为 100% 的区间上, CoSaMP 算法所需的迭代数是上述所有算法中最少的, 但随着稀疏度不断增大该算法恢复成功率下降的同时其收敛所需要的迭代数也不断增大. 当恢复成功率为零之后该算法基本无法收敛 (达到最大迭代数目 500). IHT 算法也展现了类似的趋势, 但与 CoSaMP 算法相反的是该算法收敛迭代数几乎是所有算法中所需最大的 (无论恢复成功率是否为 100%). 反观 MPHTP 的表现, 我们可以清楚地看到 $N_0 = 1$ 的 MPHTP 收敛所需的迭代数几乎等于原信号的稀疏度, $N_0 = 2, 4, 6$ 的 MPHTP 所需的迭代数少很多. 整体上来看, 矩阵广义逆硬阈值算法收敛所需迭代数大约为 k/N_0 , 即使在算法无法收敛的情况下迭代数最多的 $N_0 = 1$ 的 MPHTP 收敛所需的迭代数也不超过 200.

4.2. 加噪声数值结果

本小节主要给出的是这几种方法在有噪声时的表现. 观测矩阵、原信号生成方式与上一小节相同, 在观测时我们加入了模量为原始信号模量百分之一的高斯噪声, 即测量值 $y = Ax + e$, 其中 $\|e\| = 0.01\|x\|$.

图 3 说明在有噪声情况下与无噪声情况下相比, MPHTP 硬阈值算法较为稳定 (恢复成功率有所下降), 这说明 MPHTP 可以强健的恢复出原始信号. 压缩采样匹配追踪算法恢复成功率下降较多 (恢复成功率 100% 区间缩短到 $[0, 30]$); 而迭代硬阈值算法在我们的实验中表现是最差的, 仅在信号高度稀疏时才有可能以低于 20% 的概率恢复出原始信号. 但是在收敛所需迭代数方面, 三种算法所需迭代数大不相同: IHT 在稀疏度 $k < 40$ 的区间上所需的迭代数很少残差的数量级便可达到误差水平的数量级, 因此该算法迭代停止过早; 随后随着稀疏度不断增大, 迭代数不断上升直到给定的最大迭代数 500. CoSaMP 所需迭代数的趋势与 IHT 相同, 但在整个稀疏度变化区间上其所需迭代数均是几种算法中最大的. 而即使是所需迭代数最多的 $N_0 = 1$ 的矩阵广义逆硬阈值算法其迭代数也在 150 以下.

4.3. 运行时间

本小节报告了无噪声情况下各算法的平均运行时间. 由图 5 可知, 与 CoSaMP、IHT 相比, MPHTP 算法所需时间略多, 这与计算 A^\dagger 有关. 当 A 稀疏或具有某些特殊结构时可以采用快速算法, 如向量性算法^[16]等. 若直接求解 $A^\dagger y$ 时也可采用 Greville 递推法. 同时可以看到随着 N_0 的增大, MPHTP 算法的运行时间逐渐减少. 当 $k \in [0, 50]$ 时, CoSaMP 算法的运行时间是最少的, 但当稀疏度 $k > 50$ 时, 其所需的时间远远超过其他算法.

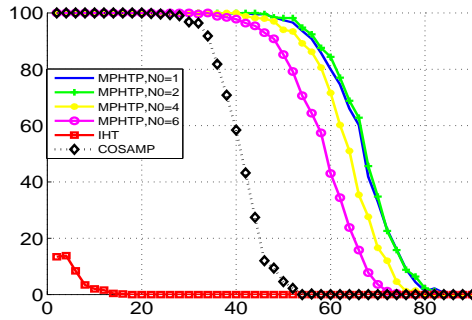


图 3 矩阵维数 $M=200, N=1000$ 时纯高斯、有噪设置下不同算法的恢复成功率比较 (X 轴稀疏度; Y 轴恢复成功率)

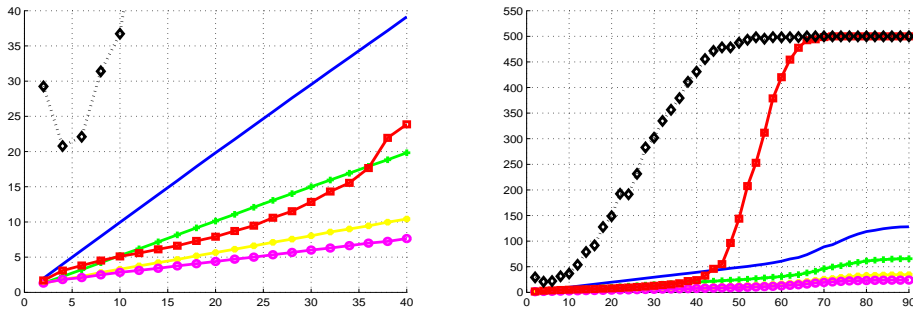


图 4 矩阵维数 $M=200, N=1000$ 时纯高斯、无噪设置下不同算法所需平均迭代数: (左) 稀疏度区间 $[0, 40]$ 的详细信息; (右) 整个稀疏度区间的详细信息 (X 轴稀疏度变化区间; Y 轴平均迭代数)

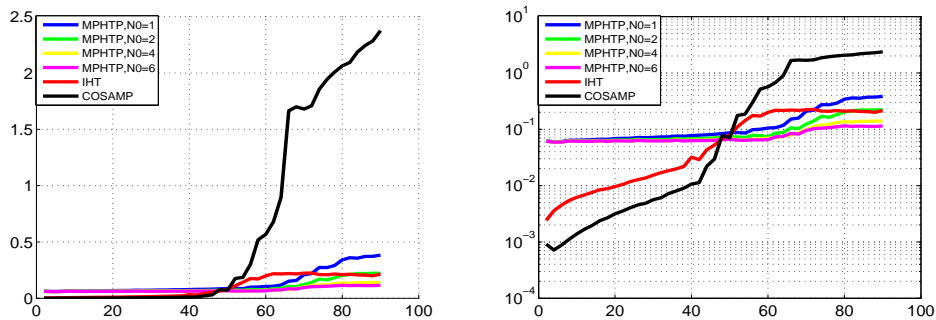


图 5 矩阵维数 $M=200, N=1000$ 时纯高斯、无噪设置下不同算法所需平均运行时间: (左) 平均运行时间; (右) 图为运行时间的对数图 (X 轴稀疏度变化区间; Y 轴平均运行时间)

5. 结 论

我们研究了用于压缩感知稀疏恢复的矩阵广义逆硬阈值追踪算法 (Matrix Pseudoinverse

Hard Thresholding Pursuit Algorithm, MPHTP), 证明了算法在观测误差存在时该算法在 RIP 条件下的收敛性. 用于 k -稀疏恢复时, 该算法在一定的条件下可以找到所有的 k -稀疏解, 同时极大程度地减少收敛所需迭代数. 数值实验表明矩阵广义逆硬阈值追踪算法具有很强的稀疏恢复能力, 并且恢复是强健的. 需要指出的一点是虽然随着支撑集捕获基数 N_0 的增大矩阵广义逆硬阈值追踪算法收敛所需的迭代数不断减小 (约为 k/N_0), 但其稀疏恢复能力逐渐衰退. 这可能是由于每次选择的支撑集基数较大时所包含错误的指标数目也会相对较多, 导致了算法表现较差. 这暗示着每次选择的支撑集基数增加速度不宜太快, 也即 $N_0 \leq \{k, \sqrt{m}\}$ 且选取不应过大. 与迭代硬阈值算法及压缩采样匹配追踪算法相比, 该算法的缺点在于需要计算系统 $Ax = y$ 的最小二乘解 $A^\dagger y$ (可以采用快速算法来求解). 但当信号较稠密或者测量误差存在的情况下, 鉴于该算法较高的恢复成功率及较少的迭代数其不失为一个好的选择.

参 考 文 献

- [1] Natarajan B K. Sparse approximation Solutions to linear systems[J]. SIAM Journal on Computing, 1995, 24(2): 227-234
- [2] Candès E J, Romberg J K and Tao T. Stable signal recovery from incomplete and inaccurate measurements[J]. Communications on Pure and Applied Mathematics, 2005, 19(5): 410-412
- [3] Donoho D L. Compressed Sensing[J]. IEEE Transactions on Information Theory, 2006, 52(4):1289-1306
- [4] Donoho D L. High-Dimensional Centrally Symmetric Polytopes with Neighborliness Proportional to Dimension[J]. Discrete and Computational Geometry, 2006, 35(4): 617-652
- [5] Tropp J A, Gilbert A C. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit[J]. IEEE Transactions on Information Theory, 2008, 53(12): 4655-4666
- [6] Needell D, Tropp J A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples [J]. Applied and Computational Harmonic Analysis, 2008, 26(3): 301-321
- [7] Wang J, Kwon S, Shim B. Generalized Orthogonal Matching Pursuit[J]. IEEE Transactions on Signal Processing, 2011, 60(12): 6202-6216
- [8] Blumensath T, Davies M E. Iterative hard thresholding for compressed sensing[J]. Applied and Computational Harmonic Analysis, 2009, 27(3): 265-274
- [9] Donoho D L. Denoising by soft-thresholding[J]. IEEE Transactions on Information Theory, 1994, 41(3): 613-627
- [10] Yin W, Osher S, Goldfarb D, et al. Bregman iterative algorithms for L1-minimization with applications to compressed sensing. SIAM J Imaging Sci 1:143-168[J]. Siam Journal on Imaging Sciences, 2008, 1(1): 143-168
- [11] Yin W, Osher S. Error Forgetting of Bregman Iteration[J]. Journal of Scientific Computing, 2013, 54(2-3): 684-695
- [12] Foucart S. Hard thresholding pursuit: an algorithm for compressive sensing[J]. Siam Journal on Numerical Analysis, 2011, 49(6): 2543-2563.
- [13] Bouchot J L, Foucart S, Hitczenko P. Hard thresholding Pursuit Algorithms: Number of Iterations[J]. Applied and Computational Harmonic Analysis, To appear
- [14] Li H, Fu Y, Zhang Q, et al. A Generalized Hard Thresholding Pursuit Algorithm[J]. Circuits Systems and Signal Processing, 2014, 33(4): 1313-1323

- [15] 李娟, 李维国, 郑昭静. 求解稀疏最小二乘问题的新型 Bregman 迭代正则化算法 [J]. 信号处理, 2012, 28(8): 1164-1170
- [16] Moriya K, Nodera T. A new scheme of computing the approximate inverse preconditioner for the reduced linear systems[J]. Journal of Computational and Applied Mathematics, 2007, 199(2):345-352

MATRIX PSEUDOINVERSE HARD THRESHOLDING PURSUIT AND SPARSE RECOVERY PROBLEM

Shi Zhanglei Li Weiguo

(College of Science, China University of Petroleum(East China), Qingdao 266580, China)

Abstract

In this paper, based on pseudoinverse and a newly introduced capture cardinality of support set, the new algorithm, Matrix Pseudoinverse Hard Thresholding Pursuit, is proposed to find the sparse solutions of underdetermined linear systems. The convergence is established under restricted isometry property (RIP) condition in the situation where observation error exist. Numerical experiments show that the proposed algorithm not only significantly reduce the iteration number required for convergence, but also can robustly recover the original signal when measurements contain errors.

Keywords: compressive sensing; sparse recovery; hard thresholding pursuit; pseudoinverse

2000 Mathematics Subject Classification: 65F10, 15A29, 94A12