

基于子空间追踪重构算法的改进

汪鲁才, 赵延昇, 林海军, 刘文卫

(湖南师范大学 工程与设计学院, 湖南 长沙 410012)

摘 要: 压缩感知的重构算法中, 子空间追踪算法(Subspace Pursuit, SP)是近几年来研究的热点. 针对 SP 算法, 引进前瞻策略和自适应选择步长的方法, 来改进其重构效果. 实验结果表明: 自适应前瞻 SP 算法(Adaptively Look Ahead Subspace Pursuit, ALASP)重构效果优于其他匹配追踪算法.

关键词: 压缩感知; 贪婪追踪算法; SP 算法; 前瞻策略; 自适应

中图分类号: TP391.4

文献标识码: A

文章编号: 1000-7180(2014)07-0151-05

Improved Reconstruction Algorithm Based on Subspace Pursuit

WANG Lu-cai, ZHAO Yan-sheng, LIN Hai-jun, LIU Wen-wei

(College of Engineering and Design, Hunan Norm University, Changsha 410012, China)

Abstract: Subspace Pursuit is one of the hot research topic. To achieve a better performance, developed a look ahead strategy that adaptively chooses the best number of coefficients. Subspace Pursuit algorithm use a support-set expansion by a fixed number of coefficients, leading to restricted performance, experimental results show that the new algorithm which can be call adaptively look ahead subspace pursuit performs better than other greedy pursuits algorithms.

Key words: compressed sensing; greedy pursuit algorithm; subspace pursuit; look ahead; adaptively

1 引言

压缩感知(CS)^[1-2]是目前信号处理技术研究的热点之一, 它是一个线性的欠采样性问题, 要求所采样的原始信号具有稀疏性. 在压缩感知中, 采样的计算量较少, 然而在重构时, 计算量很大. 这与传统的香农压缩采样刚好相反.

压缩感知的难点在于如何在重构效果和计算复杂度之间权衡. 目前出现大量的 CS 重构算法在信号重建上都取得了较好的效果. 它们通常被分为三类^[3]: (1) 贪婪追踪算法; (2) 凸松弛算法; (3) 非凸算法. 第二类和第三类算法从理论上讲, 重构的效果更好, 但面对高维的信号处理问题, 计算量太过复杂和繁重, 在实际问题中往往不实用. 相反, 贪婪追踪算法由于其计算相对比较简单, 需要的测量值比较少且重构效果也不会太差而广泛应用.

贪婪追踪算法核心思想是利用多次迭代来估算出稀疏信号的支撑集合. 贪婪追踪算法在每次迭代时, 选择一个或者更多的原子加入候选原子集直到一个收敛性的标准达到满足. 每次迭代同时选择多个原子加入候选原子集的算法又被称为并行算法. 典型的并行算法如: SP 算法^[2]和 COSAMP 算法^[3].

本文针对 SP 算法每次迭代选择固定的 K 个原子, 导致重构效果受到限制的缺陷, 提出利用前瞻策略和自适应选择加入最优原子集的数目的 ALASP 算法, 并进行实验验证. 结果表明, ALASP 算法的重构效果好于其他贪婪匹配追踪算法.

2 压缩感知模型和并行算法

2.1 压缩感知及贪婪算法

标准的压缩感知信号采集模型, 获取信号稀疏度为 K 的向量 $\mathbf{x} \in \mathbb{R}^{N \times 1}$ 采用的等式是

收稿日期: 2013-09-28; 修回日期: 2013-11-11

基金项目: 国家自然科学基金项目(51205127)

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{w} \quad (1)$$

式中, $\Phi \in \mathbb{R}^{M \times N}$, 表示测量矩阵, $\mathbf{y} \in \mathbb{R}^{M \times 1}$ 表示测量值向量, $\mathbf{w} \in \mathbb{R}^{M \times 1}$ 表示在测量中产生的噪声. 在压缩感知框架中, 由于 $M \ll N$, 那么式(1) 是一个众所周知的病态问题. 研究表明当测量矩阵满足约束等距性 (Restricted Isometry Property, RIP) 时^[4], CS 就能通过较少的测量维数 $M = O(K \log(N/K))$ 近乎精确地重构出稀疏信号 $\hat{\mathbf{x}}$. 重构的过程就是求解下式最优化的过程:

$$\min \|\mathbf{x}\|_0 \quad \text{s. t. } \mathbf{y} = \Phi \mathbf{x} + \mathbf{w} \quad (2)$$

式中, $\|\mathbf{x}\|_0$ 表示稀疏信号 $\mathbf{x}$ 中非零原子的个数.

目前流行的贪婪追踪算法迭代的主要步骤有以下两步:

(1) 原子匹配^[5]. 找出一个或几个与上一轮残差内积绝对值最大的原子, 计算相关系数 u , 并将找出的原子加入原子候选集.

$$u = \{u_j \mid u_j = |\langle \Phi_j, \mathbf{r}_i \rangle|, j = 1, 2, 3, \dots, M\} \quad (3)$$

(2) 更新. 从原子匹配中找出满足条件的列向量构成新的矩阵 Φ_Λ , 即为支撑集. 通过最小二乘法来得到稀疏信号 $\mathbf{x}$ 的估计值 $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = (\Phi_\Lambda^* \Phi_\Lambda)^{-1} \Phi_\Lambda^* \mathbf{y} \quad (4)$$

并更新残差

$$\mathbf{r}_i = \mathbf{y} - \Phi_\Lambda \hat{\mathbf{x}} \quad (5)$$

贪婪追踪算法因为实现简单、重建速度较快而被广泛应用. 典型的 OMP 算法基本思想^[6]是: 从原子库 (感知矩阵) 中选择和残差信号最匹配的原子 (内积最大), 将已选择的原子正交化, 再将测量信号投影到由正交原子所构成的空间, 计算出信号在已选原子上的残差值和分量. OMP 算法虽然迭代次数较少, 计算简单, 但对有些信号不能实现准确重构, 并且对于测量矩阵有比较严格的要求.

2.2 并行算法

为了进一步提高算法效率, 在 OMP 算法的基础上提出回溯的思想, 每次迭代从感知矩阵中选择多个较相关的原子同时对部分原子进行剔除. 这类算法被称为并行算法. 主要的并行算法目前有 COSAMP 和 SP 算法.

COSAMP 算法引入回溯的思想^[3], 每次迭代找出 $2K$ 个最相关原子, 同时剔除掉其中 K 个原子. 其算法如下:

输入: 测量信号 $\mathbf{y}$, 稀疏度 K , 测量矩阵 Φ ;
输出: 重建信号 $\hat{\mathbf{x}}$;

初始化: 残差余量 $\mathbf{r}_0 = \mathbf{y}$, 迭代次数 $t = 1$, 索引值集合 $\Lambda = \varphi, J = \varphi$;

① 计算测量矩阵和残差余量内积绝对值, 将其中最大的 $2K$ 个原子所对应的索引值存入 J_t 中

② $\Lambda_t = J_t \cup \Lambda_{t-1}$, 应 $\hat{\mathbf{x}} = \arg \min \|\mathbf{y} - \Phi_\Lambda \mathbf{x}\|_2$, 并选择取前 K 个最大元素对应的原子放入支撑集 Φ_Λ ;

③ 更新残差余量 $\mathbf{r}_t = \mathbf{y} - \Phi_\Lambda \hat{\mathbf{x}}$;

④ 若 $t > K$, 则停止迭代; 否则令 $t = t + 1$, 转步骤 ①.

子空间匹配追踪算法 (Subspace Pursuit, SP) 的提出, 使得重构信号的计算更加精准. 该算法在已知稀疏度的条件下, 在一次循环迭代内选出与稀疏度相同数目的原子, 通过反复的原子选入和原子剔除的多次计算, 尽量抵消一次循环内选出多个原子造成的误差. SP 算法具体描述如下:

输入: 测量信号 $\mathbf{y}$, 稀疏度 K , 测量矩阵 Φ ;

输出: 重建信号 $\hat{\mathbf{x}}$;

初始化: 残差信号 $\mathbf{r}_0 = \mathbf{y}$, 索引值集 $\Lambda = \varphi, J = \varphi$, 循环次数 $t = 1$;

① 计算测量矩阵与残差内积绝对值, 将其中最大的 K 个元素, 将对应的索引存入 J_t 中;

② 更新支撑集 $\Lambda_t = J_t \cup \Lambda_{t-1}$;

③ 利用式子 $\hat{\mathbf{x}} = \arg \min \|\mathbf{y} - \Phi_\Lambda \mathbf{x}\|_2$, 计算重建信号 $\hat{\mathbf{x}}$, 并选择其中最大的 K 个元素所对应的索引, 来更新支撑集 Λ_t ;

④ 利用式子 $\mathbf{r}_t = \mathbf{y} - \Phi_\Lambda \hat{\mathbf{x}}$, 更新余量;

⑤ 当 $\|\mathbf{r}_t\| > \|\mathbf{r}_{t-1}\|$, 退出循环. 否则令 $t = t + 1$ 并转至步骤 ①

由上可以看出 SP 和 COSAMP 算法之间的主要不同是, SP 算法每次迭代选择 K 个原子, 而 COSAMP 每次迭代选择 $2K$ 个原子. 两者的重构速度都要优于 OMP 算法, 其中 SP 重构的效果要优于 COSAMP 算法, 但两者的重构效果会稍差于 OMP 算法. 文献[6]中指出 SP 算法在对稀疏度以及采样率的要求上比 OMP 算法更为宽松, 是一种性价比较高的贪婪追踪算法.

3 ALASP 算法

为了保留 SP 算法对稀疏度和采样率要求比较宽松的优点, 进一步提高其重构的精度, 文中提出 ALASP 算法. 它引入前瞻策略和自适应的选择每次迭代引入原子集合 K_{grow} 的大小, 确保了在提高重构

精度的同时不会增加太大的计算量。

前瞻策略思想来源于 LAOMP 算法。在 LAOMP 算法中,每次迭代找出 L 个最相关的原子,然后从中挑出使得残差最少那个原子加入支撑集中。LAOMP 算法重构效果得到了明显的改善,但是计算的复杂度大大增加,随着 L 的增大,LAOMP 花费的时间越多。ALASP 算法思想是每次迭代是动态的选择 K_{grow} 个原子($K < K_{\text{grow}} < 2K$),这 K_{grow} 个原子的选择是通过前瞻的策略,选择残差最少的而不是内积最大的 K_{grow} 个原子,来确定每次迭代 K_{grow} 值的大小,然后结合 SP 算法的思想,对 K_{grow} 个原子进行剔除,选择出 K 个最合适的原子加入支撑集。因为每次迭代选择的原子个数为 K 个,所以其重构的时间肯定要短于 LAOMP 算法(每次迭代选择一个最优原子)。

ALASP 算法的原理示意图见图 1,为了对比,SP 算法的原理示意图见图 2。两个算法之间微妙但很重要的不同在于生成 Λ 这个原子支撑集的方法不一样。SP 算法在每次迭代中通过内积最大选择 K 个原子加入 Λ 支持集,然后在每次迭代中对 K 个原子进行更新。ALASP 算法通过前瞻策略选择使得内积最小的 $(K+L^*)$ 个原子加入 Λ 支持集,然后在对 $(K+L^*)$ 个原子进行筛选保留 K 个原子,依次迭代更新。

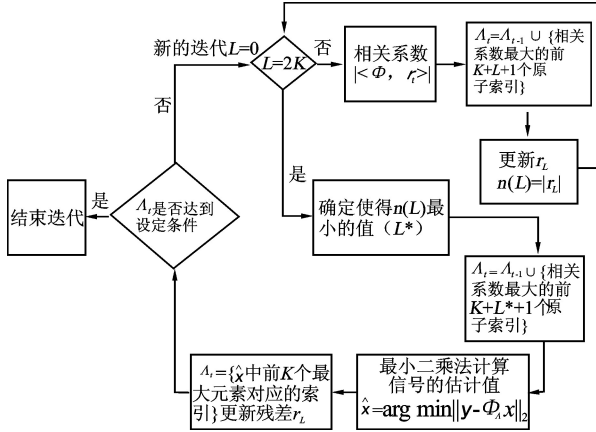


图 1 ALASP 算法示意图

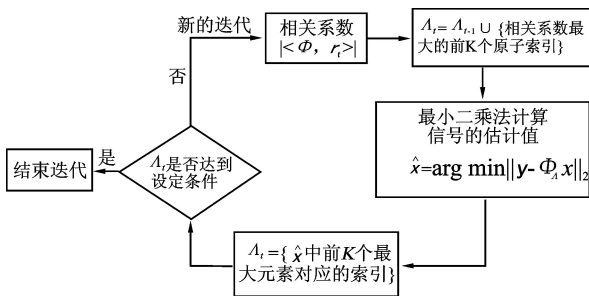


图 2 SP 算法示意图

ALASP 算法部分代码见表 1。

表 1 ALASP 算法

```

输入: 测量信号  $y$ , 稀疏度  $K$ , 测量矩阵  $\Phi$ ;
初始化: 支撑集  $\Lambda = \varphi$ , 残差余量  $r_0 = y$ , 循环次数  $t = 0$ ;
循环执行步骤:
While( $\|r_t\|_2 \leq \|r_{t-1}\|_2$ ) do // 迭代终止条件
     $t = t + 1$ ;  $l = 0$ ; //  $t$  为循环次数    /* 自
适应的选择最佳的  $l(l^*)$  */
    While( $l < K + 1$ ) do
         $l = l + 1$ ;
         $K_{\text{grow}} = K + l - 1$ ;
         $\Lambda = \max_{K_{\text{grow}}} (\Phi, r_{t-1})$ ;
         $\hat{x} = \arg \min \|y - \Phi_{\Lambda} x\|_2$ ;
         $r_t = y - \Phi_{\Lambda} \hat{x}$ ;
         $n(l) = \|r_t\|_2$ ;
    end
     $n(l^*) = \min(n(l))$ ; // 前瞻策略选择最合适的  $l$ 
     $K_{\text{grow}} = K + l^* - 1$ ; // 确定  $K_{\text{grow}}$  的大小
     $\Lambda = \max_{K_{\text{grow}}} (\Phi, r_{t-1})$ ;
     $\hat{x} = \arg \min \|y - \Phi_{\Lambda} x\|_2$ ;
     $\Lambda = \max\_indices(\hat{x}, K)$ ;
    // 确定其中  $K$  个原子加入候选原子集合
     $r_t = y - \Phi_{\Lambda} \hat{x}$ ; // 更新残差
end
输出: 重构的信号  $\hat{x}$ 

```

4 实验结果及其分析

4.1 一维理想信号重构

通过理想信号的稀疏重建分析 OMP, COSAMP, SP 及本文 ALASP 算法对信号的恢复准确率。对于高斯稀疏信号, 实验中采用的一维稀疏信号为高斯稀疏信号和二值稀疏信号。由文献[7]可知 OMP 算法对高斯稀疏信号重构效果较好, 但对于二值稀疏信号恢复重构效果很差。实验参数设置: 令信号维数 $N=500$, 稀疏度 $K=20$, 测量矩阵采用高斯随机矩阵。实验结果如图 3 所示。

图 3(a)中给出了四种不同的重建算法对高斯稀疏信号重构的效果对比结果。当采样率小于 0.13 时, 四种恢复算法的重构效果相当。当采样率大于 0.14 后, ALASP 算法的重构效果最好, OMP 算法其次, SP 和 COSAMP 算法相对较差; 当采样率大于 0.23 后, OMP 算法相较其他算法最差。

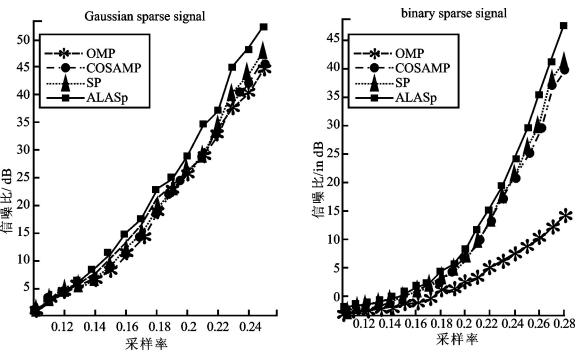


图 3 一维信号重构信噪比

针对二值稀疏信号,OMP 算法重构的效果非常差,当采样率大于 0.24 后,ALASP 算法的重构信噪比相比 OMP 算法高出 10 db 以上.相比 SP 和 COSAMP 算法,ALASP 重构信噪比高出 5 db 以上.

4.2 二维图像重构

本文对图像 house 和 lena 进行测试,采用高斯随机矩阵对其测量,稀疏矩阵为离散余弦变换矩阵.利用分块重构思想,将 256 * 256 的 house 图像和 lena 图像分成 16 * 16 大小的子块,高斯矩阵为 128 * 256 对每块进行测量,然后分别使用 OMP,COSAMP,SP 以及 ALASP 算法来重构图像,结果见图 4、图 5 及表 2 和表 3.

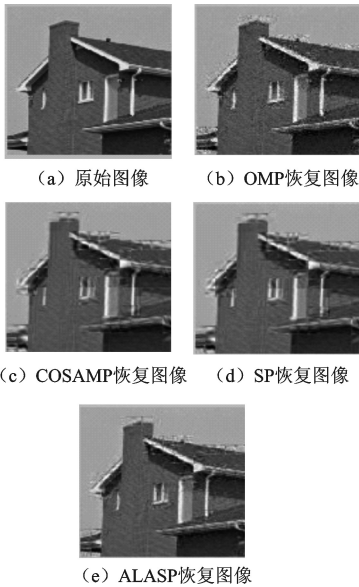


图 4 House 图像重构结果图

表 2 House 图像各算法重构质量对比

算法	均方误差	信噪比/db	结构相似度	运行时间/s
OMP	30.020 3	33.356 7	0.758 7	14.054 058
COSAMP	31.072 6	33.207 0	0.739 4	6.921 323
SP	30.390 9	33.303 4	0.740 1	7.070 489
ALASP	24.405 1	34.256 7	0.801 5	14.984 830



图 5 Lena 图像重构结果图

表 3 Lena 图像各算法重构质量对比

算法	均方误差	信噪比/db	结构相似度	运行时间/s
OMP	35.554 3	32.621 9	0.729 3	14.160 109
COSAMP	37.282 6	32.415 7	0.710 4	6.517 219
SP	37.183 5	32.425 7	0.711 7	6.880 259
ALASP	29.566 6	33.428 7	0.783 5	16.063 306

由图 4 和表 2 可以看出,OMP 算法对 House 图像重构峰值性噪比达到 33.356 7,但对图像的一些边缘信息的重构出现明显的噪声,而且该算法需预设稀疏度,由于稀疏度未知,本文设置 OMP 最大迭代次数为 64 次,OMP 算法的运行时间也较长,为 14 s;SP 和 COSAMP 算法在处理该图像时虽然用时短都接近为 7 s,但图像的重构效果较 OMP 算法有所变差,块效应比较明显.本文提出的 ALASP 算法相比 SP 和 COSAMP 算法时间有所变长,但和 OMP 算法时间相差不多,同时峰值信噪比得到了提高,图像的重构效果更好.和图 5 和表 3 对比发现,对 Lena 图像的重构效果稍差于对 House 图像的重构.因为 Lena 图像包含的细节更多,各个频段的能量更加丰富.但几种算法相比较表明,ALASP 算法还是优于其他几种贪婪算法.

5 结束语

文中提出的 ALASP 算法结合了前向预测和自适应选择迭代引入原子步长的优点,通过理论分析和实验仿真,ALASP 算法虽然相较 SP、COSAMP 算法在重构时间上花费有所变长,但其重构效果上得到明显改善.尤其相比 OMP 算法,在重构时间上相差不多

的情况下,其重构精度更高且对不同的信号重构的效果都很好。

参考文献:

- [1] Donoho D L. Compressed sensing[J]. IEEE Transactions on Information Theory, 2006, 52(4): 1289-1306.
- [2] Dai W, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction[J]. IEEE Transactions on Information Theory, 2009, 55(5): 2230-2249.
- [3] Needell D, Tropp J A. CoSaMP: iterative signal recovery from incomplete and inaccurate samples[J]. Applied and Computational Harmonic Analysis, 2009, 26(3): 301-321.
- [4] Candes E J, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information[J]. IEEE Transactions on Information Theory, 2006, 52(2): 489-509.
- [5] Candes E J, Tao T. Decoding by linear programming

[J]. IEEE Transactions on Information Theory, 2005, 51(12): 4203-4215.

- [6] 裴文炯,李少东,杨军. 压缩感知中四种贪婪类算法重构信号性能研究[J]. 空军雷达学院学报, 2012, 5(4): 145-149.
- [7] Tropp J A, Gilbert A C. Signal recovery from random measurements via orthogonal matching pursuit [J]. IEEE Transactions on Information Theory, 2007, 53(12): 4655-4666.

作者简介:

汪鲁才 男, (1969-), 博士, 教授. 研究方向为图像处理与模式识别.

赵延昇 男, (1987-), 硕士. 研究方向为数字图像处理.

林海军 男, (1980-), 博士, 副教授. 研究方向为图像处理与模式识别.

刘文卫 男, (1986-), 硕士. 研究方向为数字图像处理.

(上接第150页)

- [9] 申丽君, 刘丽, 陆锐. 基于改进免疫进化算法的云计算任务调度[J]. 计算机工程, 2012, 38(9): 208-210.
- [10] Xu B M, Zhao C Y, Hua E Z, et al. Job scheduling algorithm based on Berger model in cloud environment [J]. Advances in Engineering Software, 2011(42): 419-425.
- [11] 左利云, 左利锋. 云计算中基于预先分类的调度优化算法 [J]. 计算机工程与设计, 2012, 33(4): 1357-1361.
- [12] 刘愉, 赵志文, 李小兰, 等. 云计算环境中优化遗传算法的资源调度策略[J]. 北京师范大学学报: 自然科学版, 2012, 48(4): 378-384.

[13] Karaboga N. A new design method based on artificial bee colony algorithm for digital IIR filters [J]. Journal of the Franklin Institute, 2009, 346(4): 328-348.

[14] 胡珂, 李迅波, 王振林. 改进的人工蜂群算法性能 [J]. 计算机应用, 2011, 31(4): 1107-1110.

作者简介:

卓涛 女, (1978-), 讲师. 研究方向为软件工程与云计算技术.

詹颖 女, (1978-), 讲师. 研究方向为计算机网络.