快速最佳子集选择:坐标下降和局部组合优化算法

Hussein Hazimeh 和 Rahul Mazumder †

麻省理工学院

2018 年 3 月

抽象的

我们考虑典型的L0 正则化最小二乘问题（又称最佳子集）,它通常被视为许多稀疏学习方案的"黄金标准"。尽管存在最坏情况下的计算难处理性结果,但最近的工作表明,对于特征数量 $p \approx 10^3$ 的情况,混合整数优化方面的进步可用于获得该问题的接近最优解。虽然这些方法导致估计器具有出色的统计特性,但计算时间的急剧增加通常是要付出代价的,尤其是与用于稀疏学习的高效流行算法（例如,基于L1 正则化)相比时扩展到更大的问题规模。弥合这一差距是本文的主要目标。我们研究了一系列带有额外凸惩罚的 L0 正则化最小二乘问题的计算方面。我们为这些问题提出了一个必要最优条件的层次结构。我们开发了基于坐标下降和局部组合优化方案的新算法,并研究了它们的收敛特性。我们证明了算法的选择决定了所获得解决方案的质量;和基于局部组合优化的算法通常会产生高质量的解决方案。我们凭经验证明,与更简单的启发式算法相比,我们提出的框架对于 $p \approx 10^6$ 的问题实例相对较快,并且在优化和统计属性（例如,预测、估计和变量选择)方面运行良好。与最先进的稀疏学习方案（如 glmnet 和 ncvreg)相比,我们的算法的一个版本可实现三倍加速（p 高达 10^6 ）。

1 简介

我们考虑通常的线性回归设置,响应 $y \in R^n$ 和回归系数 $\beta \in R^p$。我们将假设 X 的列以均值为中心并标，模型矩阵 $X \in R^{n\times p}$，准化为具有单位 ℓ_2-norm 并且 y 为中心。在具有 p [illegible] 作为数据保真项,这导致了众所周知的约束形式的最佳子集选择问题 [25]：

$$\hat{\beta}^{L0} \in \arg\min_{\beta\in R^p} \frac{1}{2}\|y - X\beta\|_2^2 \quad \text{英石} \quad \|\beta\|_0 \leqslant k, \tag{1}$$

*H。 Hazimeh 的研究得到了 ONR-N000141512342 的部分支持。电子邮件:hazimeh@mit.edu

† R。 Mazumder 的研究得到了 NSF-IIS-1718258 和 ONR-N000141512342 的部分支持。电子邮件：rahulmaz@mit.edu

其中,$\|\beta\|_0 \overset{定义}{=} \sum_{i \in [p]} 1[\beta_i \neq 0]$ 表示 β 的L0伪范数,k 控制模型大小。

$\hat{\beta}_{L0}$的统计特性是众所周知的,参见例如 [13, 14, 30, 37](以及其中的参考文献),并且该估计量被广泛认为是稀疏回归的“黄金标准”(假设它可以计算)。假设数据是从真正的线性模型 $y = X\beta^0 +$ 生成的,其中 β iid N(0, σ2)。众所周知,当信噪比 (SNR)[1]较高时,$\hat{\beta}_{L0}$具有出色的统计特性(变量是稀疏的以及选择、估计和预测误差)。事实上,在几个方案中,与计算友好的方案(例如,基于 L1 正则化)相比,$\hat{\beta}^0$

会下降 [24, 12]。最近 [illegible] $\hat{\beta}_{L0}$相比可以提供更好的预测模型。 [24] 建议通过考虑以下形式的正则化变体来规避 $\hat{\beta}_{L0}$的这种不利行为:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_q \quad \text{st} \quad \|\beta\|_0 \leqslant k \tag{2}$$

其中 $\|\beta\|_q$, $q \in \{1, 2\}$ 是 β 的Lq范数,λ 控制收缩量。 [24] 证明(从理论上和经验上)估计器(2)与 Lasso/ridge 相比具有更好或相当的预测准确性,并且它通常会导致非零值更少的解决方案。因此,在本文中,我们考虑了正则化子集选择估计器的惩罚版本[2]

上面介绍过:

$$\min_{\beta \in \mathbb{R}^p} F(\beta) \overset{定义}{=} f(\beta) + \lambda_0\|\beta\|_0 \tag{3}$$

其中, $\lambda_0 > 0$ 和 f(β) 是具有附加凸惩罚的最小二乘项:

$$f(\beta) \overset{定义}{=} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2、 \tag{4}$$

和$\lambda_1, \lambda_2 \geqslant 0$ 是控制连续收缩年龄量的先验已知调谐参数。在本文中,我们将至少设置调整参数λ1、 λ2中的一个为 0,并关注以下情况:
(i) $\lambda_1 = 0$ 且$\lambda_2 > 0$,问题 (3) 表示为(L0L2) (ii) $\lambda_1 > 0$ 和$\lambda_2 = 0$,其中 (3) 表示为(L0L1)和 (iii) $\lambda_1 = \lambda_2 = 0$,其中 (3) 表示为(L0)。

当前的计算环境:我们讨论问题 (3) 的一个关键方面:它的计算特性。计算 $\hat{\beta}_{L0}$是已知的 NP-hard [26] - 事实上,众所周知的 R 包跳跃可以计算 $n \geqslant p \approx 30$ 的问题 (1) 的解决方案。最近,[5] 表明显着的计算进步在混合整数优化 (MIO) 中,可用于计算问题 (1) 的近似最优解,因为实例比自 [11] 以来在更广泛的统计社区中认为可能的实例大得多。 [5] 证明,对于 $p \approx 1000$ 的问题实例,使用 Gurobi 的(商业级)MIO 求解器在几分钟内可以为最佳子集问题获得高质量的解决方案,当使用离散的良好热启动进行初始化时一阶算法类似于迭代硬阈值 (IHT)[3] 。这种基于 MIO 的框架与其他启发式方法的不同之处在于它能够通过双重边界提供(近似)最优性的证书,但代价是更长的计算时间。 [24] 将 [5] 的方法应用于问题 (2)。 [4] 提出了一种令人印象深刻的切割平面方法(使用 Gurobi)

---

[1] 对于$y_i = \mu_i + _i$的线性模型,$i \in \{1, . . . , n\}$ 我们定义 SNR=Var(μ)/Var(·)。

[2] 惩罚版本主要是从计算角度选择的。

[3] 通常,Gurobi 可能需要 5-30 分钟才能为特定子集获得接近最佳的解决方案
使用通过离散一阶方法获得的解决方案热启动时的大小

用于子集选择,它适用于温和的样本相关性和足够大的 n。 [22]在 Dantzig 选择器的 L0 变体的上下文中演示了 MIO 的使用。复杂的基于数学优化的技术（例如 MIO)似乎对近实时训练不是最重要的应用很有用,但能够获得具有最优性证书的高质量解决方案是最重要的。另一方面， L1 正则化 (Lasso) 求解器的极其高效和优化的实现（例如 glmnet [10])通常可以在不到一秒的时间内计算出整个正则化路径（具有一百个调整参数值） .事实上,与 Lasso 等流行的学习方案相比,使用最佳子集 [5] 的计算时间似乎要付出高昂的代价。正如 [17] 所指出的,增加的计算时间可能会阻止从业者对问题 (1) 采用基于全局优化的求解器来进行日常数据分析。然而,众所周知 [20, 5, 38, 23, 32] 可以通过 Lasso（及其变体)和非凸子集选择的近似最优解决方案实现的解决方案的统计质量存在显着差距类型程序。

此外,正如我们在本文中所探讨的,算法的选择会显着影响所获得解决方案的质量:在许多情况下,在优化非凸子集选择标准 (3) 方面表现更好的算法会产生高质量的统计数据。估算器（例如,在支持恢复方面）。在我们的实验中,我们观察到,当潜在的统计问题很困难时,几乎所有用于稀疏学习的最先进算法（套索、迭代硬阈值、逐步回归、MCP 惩罚回归等)都在

热门包,未能恢复β的支持 ^{0} .然而,使用我们在此开发的局部组合优化方法对问题 (3) 进行更好的优化,似乎可以在可比较的运行时间内改善这个问题。

我们的贡献:上述讨论表明,为子集选择类型问题开发算法框架至关重要,从而在与Lasso 的快速坐标算法相当的时间内为 $p \approx 103$ -106的问题提供接近最优的解决方案/nonconvex (MCP) 惩罚回归,例如。为此,我们重新审视了针对问题 (1) 的流行启发式方法[illegible]我们提出的问题最优属性？ (ii) 例如,这些算法能否在速度上与 Lasso 的有效算法相媲美?解决这些问题是本文的主要目标。

我们从稀疏回归 [7, 23, 10, 27] 中普遍使用的基于坐标下降 (CD) 的算法的效率中汲取灵感。然而,与 Lasso 等凸问题不同,问题 (3) 是非凸问题,因此,研究解决方案的质量更加精细。由于我们寻求创建运行时间可与 glmnet 和 ncvreg 相媲美的算法（例如）,因此与基于全局优化的 MIO 框架密切相关的全局最优性概念似乎实际上是不现实的。因此,我们需要研究较弱的最优性概念。为此,解决方案 β 对问题 (3) 最优的必要条件激发了各种平稳性或局部最优性的概念（正如我们在本文中定义的那样）。

事实上,正如我们在本文中所讨论的,平稳解的概念与用于获得问题 (3) 解的算法类型密切相关。换句话说,解决方案的质量取决于用于问题 (3) 的算法。例如（参见第 2 节）,如果 $S \subset \{1, 2, \ldots, p\}$ 然后 $\min\{f(\beta) \mid \beta_i = 0, i \notin S\}$ 导致问题 (3) 的平稳解。然而,正如我们所展示的,使用更高级的优化程序可以获得更好的（就较小的目标值而言)解决方案。由于问题 (3) 是平滑凸损失和非平滑正则化器的总和,它在 β 的坐标上是可分离的,因此可以应用 IHT 类型的方法 [24, 6]；

甚至坐标算法 [29] 以获得问题 (3) 的良好解决方案。这些算法的不动点对应于问题 (3) 的固定解的受限概念 与 IHT 类型算法相关的不动点总是包括与坐标算法相关的不动点（参见第 2 节）。事实上，坐标平稳解对我们是否对每个坐标执行完全最小化很敏感。通过从局部组合优化算法中汲取灵感，人们可以获得更精细的固定解。为此，我们引入了交换不可避免最小值的概念（第 2 节） 换句话说，这些是固定解决方案，不能通过（局部）扰乱固定解决方案的当前支持并优化新支持来改进。

这为问题 (3) 引入了必要最优条件的层次结构，并激发了问题 (3) 的新算法。我们证明，当一个人在层次结构中向上移动时，与其他几种类似的稀疏正则化技术相比，可以获得（i）问题（3）的更好解决方案，运行时间略有增加，以及（ii）具有优越统计特性的估计器计算可扩展性和运行时间。

我们将我们的贡献总结如下：

1. 我们为问题 (3) 引入了一系列必要的最优性条件，导致了一系列固定解。层次结构中较高的类具有更高的质量，并且包括无法通过局部扰动来改进以支持当前解决方案的静态解决方案。

2. 我们开发了一个算法框架，依靠循环 CD 和局部组合优化，使我们能够获得这些类别的固定解决方案。我们探讨了问题 (3) 的优化算法的选择如何影响获得的解决方案的质量。

   我们对坐标算法的变体建立了一种新颖的收敛分析，该算法对每个坐标执行完全优化。我们的局部组合优化算法基于高度结构化的 MIO 公式，当 p 大约为 103到 105时，该公式可以在几秒到几分钟内运行。

3. 我们提供了一个开源且可扩展的 C++ 工具包 L0Learn，带有 R 接口4 ，实现了本文中的算法。我们的实现特别注意几个微妙的计算细节，并利用问题结构来实现运行时间，这些运行时间通常比套索 (glmnet)、非凸惩罚回归 (ncvreg) 的高效实现更快。对于 p 高达 106和 n ≈ 103 ，我们的算法版本在真实和合成数据集上的典型加速在 25% 到 300% 之间。

4. 在真实和合成数据集的一系列实验中，我们证明了本文提出的算法在优化问题（3）方面做得更好，并且在估计、预测和变量选择精度方面与流行的相比具有更好的统计性能使用最先进的方法进行稀疏学习。根据经验，我们的算法获得的解决方案的质量与通过 MIO 对完整问题 [5] 获得的解决方案的质量相似，但运行时间明显更短且更实用。

1.1 相关工作

有大量关于稀疏线性回归算法的文献 参见例如 [5, 1] 的概述。最佳子集选择（即问题（1））的流行启发式是（贪婪）逐步

4可在http://github.com/hazimehh/L0Learn获得

回归 [17, 16] - 然而,一旦特征数量达到数万个数量级,这就会变得非常昂贵。 IHT 或近端梯度类型算法 [6, 5] 也是问题 (1) 及其拉格朗日版本的流行选择。然而,IHT 类型的方法需要在每次迭代中进行完整的梯度评估,这使得它在计算上的吸引力不如坐标下降 (CD) 类型的方法 我们也经历了对问题 (3) 的类似观察。

实际上,在 Lasso 的情况下,CD 类型算法在计算上比近端梯度方法更具吸引力 [10, 27]。此外,我们在第 2 节中展示了与 IHT 类型算法相关的平稳解严格包含与 CD 类型算法相关的那些。

[1] 对问题 (1) 和 [29] 也对 IHT 和 CD 类型算法进行了类似的观察。此外,对于最小二乘损失 [23, 7] 提出了用于连续(非凸)正则化器(如 MCP、SCAD)的有效循环 CD 算法,采用了 [35] 的框架,该框架在每个坐标块中使用完全最小化。我们注意到问题(3)中的目标函数在每个坐标中都不是拟凸的,因此不能保证完全最小化的 CD 收敛 [35]。 [29] 对问题(1)的惩罚版本使用了随机 CD,其步长较为保守。保守的步长和坐标的随机选择有助于算法的收敛性分析。然而,保守的步长会导致一类包含完全最小化生成的平稳解(见第 3 节)。此外,根据我们的经验证据,发现每个坐标都完全最小化的(部分贪婪的)循环 CD 优于在解决方案质量和运行时间方面具有保守步长的随机 CD 方法。由于其具有竞争力的性能,我们使用(部分贪婪的)循环 CD 算法来获得问题 (3) 的坐标平稳解。

然而,证明该算法对坐标平稳解的收敛性并不简单 严格建立收敛性是我们工作的重要贡献。我们注意到 [2] 提倡使用循环 CD 规则而不是随机 CD 来解决凸问题,因为它们的收敛速度更快。此外,在随机 CD 算法的上下文中,对大规模问题的随机数生成器的调用可能成为计算负担 [27]。

我们工作的一个重要方面使其不同于早期关于最佳子集选择的类 CD 算法的工作 [1, 29] 是探索局部组合优化方案以定义更精细的固定解决方案类别。 [1] 为问题 (1) 提出了一种特殊情况,即单坐标交换的形式 然而,我们的方法存在重要差异,因为我们考虑了更小的固定解决方案类别,通过 (local) 组合优化方案。此外,我们的工作仔细考虑了大型问题的计算效率和可扩展性,这是类似算法以前没有探索过的一个方面(参见第 2 节和第 5 节中的讨论)。

1.2 符号和预备知识

我们在本文中使用以下符号。我们表示集合 {1, 2, . . . , p} by [p], R p by e1,的规范基础。 . . , ep和标准`2欧几里得范数由 k · k。对于 β ∈ R p Supp(β) 表示它的支持,即具有非零条目的索引。对于 S ⊆ [p], βS ∈ R |S|表示 β 的子向量,其索引在 S 中。类似地, XS表示 X 的子矩阵,列索引为S S。U 表示 ap × p 矩阵,如果 i ∈ S,则其第 i 行为ei ,否则为零。因此,对于任何 , (U Sβ)i = βi如果 i ∈ S 并且 (U Sβ)i = 0 如果 i /∈ S. β ∈ R p

我们使用简写: (i) L0L2表示问题 (3),其中λ1 = 0 且λ2 > 0; (ii) L0L1表示问题 (3),其中λ1 > 0 和λ0; (iii) L0 表示问题 (3),其中λ1 = λ2 = 0。
此外,对于问题 (3),我们假设λ0 > 0。

2 必要的最优条件

我们研究了问题 (3) 的必要最优条件的不同概念。我们从平稳解的基本概念开始,随后在第 2.2 节和第 2.3 节中完善这个概念。

2.1 固定解决方案

对于函数 g : R p → R 和 g 的向量 d ∈ R p在 β 方向 d 上：，我们表示（下)方向导数[3, 35]

$$G^{0}(\beta; d) \overset{\text{定义}}{=} \liminf_{\alpha \downarrow 0} \frac{g(\beta + \alpha d) - g(\beta)}{\alpha} .$$

方向导数在描述优化问题的必要最优性条件方面起着重要作用[3]。例如,让我们考虑当 g 连续可微时的无约束最小化。这里,众所周知的一阶平稳条件,∇ g(β) = 0

强加 g $^{0}(\beta; d) = \nabla g(\beta)$ 吨对于任何 d,d ⩾ 0。请注意,β → f(β) 是凸的,任何次梯度由 ∇ f(β) ∈ R p表示。如果 β 支持 S,则函数在βS→处的梯度S 处可微。因此， ∇ Sf(β) ∈ R |S|是 f(uS)

尽管（问题 (3) 的)目标 F(β) 不是连续的,但使用方向导数的概念来得出问题 (3) 的平稳性的基本定义是很有见地的

定义 1.（平稳解)如果对于每个方向向量 d ∈ R p ,向量 β ∈ R p是问题（3)的平稳解
，下方向导数满足： F $^{0}(\beta; d) \geqslant 0$。

下一个引理给出了 F 的更明确的表征 $^{0}(\beta; d)$ 是非负的。

引理 1. 令 β ∈ R p有支持 S。那么,β 是问题 (3) 的平稳解,当且仅当∇ Sf(β) = 0。

证明。对于任何 d ∈ R ^{d}, 我们将证明 F $^{0}(\beta; d)$ 由下式给出：

$$F^{0}(\beta; d) = \begin{cases} \langle \nabla Sf(\beta), dS \rangle & \text{如果 } dSc = 0 \\ \infty & \end{cases} \tag{5}$$

令 d 为 R p中的任意向量。然后，

$$F^{0}(\beta; d) = \lim_{\alpha \downarrow 0} \frac{F(\beta + \alpha d) - F(\beta)}{\alpha}$$

$$= \liminf_{\alpha \downarrow 0} \underbrace{\frac{f(\beta + \alpha d) - f(\beta)}{\alpha}}_{\text{第一学期}} \underbrace{\sum_{i \in S} \frac{k\beta i + \alpha d i k_0 - 1 k_{\alpha d j k_0} + \lambda_0 X + \lambda_0 X}{\alpha}}_{\text{第二学期}} \underbrace{\sum_{j \in S}}_{\text{第三学期}} .$$

首先我们注意到lim$\alpha \downarrow 0$ 项II = 0,因为对于任何 i ∈ S,对于足够小的 α ,$k\beta i + \alpha d i k_0 = 1$。 (βS; dS) = 假设dSc = 0。那么,f 的连续性意味着lim$\alpha \downarrow 0$ 项I = f,其中第二个等式通过观察βS → $^{0}\langle \nabla Sf(\beta), dS \rangle$, f(βS)是连续可微的（在βS 附近）。此外,Term III = 0。因此,我们有：

$$F^{0}(\beta; d) = \lim_{\alpha \downarrow 0} \text{项 I} + \lim \text{项 II} = \langle \nabla Sf(\beta), dS \rangle 。 \quad \alpha \downarrow 0$$

我们现在考虑dSc 6= 0 的情况。在这种情况下， limα↓0 项III = ∞;并且由于 Term I 的极限是有界的,我们有 F (β; d) = ∞。因此,我们已经证明（5)成立。从（5），我们有 F

°(β; d) ⩾ 0 对于所有 d iff ∇Sf(β) = 0。 □

请注意,对于每个 i ∈ Supp(β),∇if(β) = 0 可以等效地写为：

$$\lambda 1\ \beta i = 符号(\beta ei)和\ \frac{|\beta ei| -}{|\beta ei|} > \lambda 1,\ 1 + 2\lambda 2 \qquad 对于所有\ i \in Supp(\beta), \tag{6}$$

其中,βei def= hy − P j6=i Xjβj , Xii。表征 (6) 表明平稳解 β 不依赖于正则化参数λ0。此外,(6) 没有对β 支持之外的坐标施加任何条件。在接下来的评论中,我们展示了一个固定解也是问题 (3) 的局部最小值。

备注 1. 我们注意到平稳解 β 是问题 (3) 的局部最小值。我们在下面给出这个结果的证明。

由于 f 的连续性,存在一个正标量 δ 和一个非空球 R = {β ∈ R p | ||β −β|| < δ} 使得对于每个 β ∈ R,我们有 |f(β) − f(β)| < λ0。令 S = Supp(β)。我们假设 wlog δ 足够小,因此如果 i ∈ S,那么对于每个 β ∈ R,我们有 i ∈ Supp(β)。对于任何 β ∈ R,如果βi 6= 0 对于某些 i /∈ S,我们有 (||β ||0 − ||β||0) ⩽ -1。这意味着

$$F(\beta^*) - F(\beta) = f(\beta^*) - f(\beta) + \lambda 0(||\beta^*||0 - ||\beta||0) \leqslant |f(\beta^*) - f(\beta)| + \lambda 0(-1) < \lambda 0 - \lambda 0 = 0。$$

否则,如果 Supp(β) = S,则 β 的平稳性和 f 的凸性意味着 f(β) ⩽ f(β) 并且因此 F(β) ⩽ F(β)。因此,对于任何 β ∈ R,我们有 F(β) ⩽ F(β)。

我们现在介绍上面介绍的固定解决方案类的改进。

2.2 坐标最小值

我们考虑了一类受坐标算法启发的固定解决方案[35,1,3]。如果针对每个单独坐标的优化不能改善目标,则静止点 β 是问题 (3) 的坐标最小值。定义如下： 定义 2.（坐标方向 (CW) 最小值)如果对于每个 i ∈ [p]， β 是F(β) wrt 第 i 个坐标（其他坐标固定）,即

$$\beta^*\ \beta\ )。 \in \arg\min_{\beta i \in R} F(\beta 1^*, \beta\ i-1, \beta i, \beta\ i+1, \ldots, \ p \tag{7}$$

令 i ∈ [p] 和 βei是定义为 βei def = hy − P j , Xii 的标量。由于 X 的列具有单位 6=i Xjβj ℓ2-范数,问题 (7) 的解决方案可以通过简单的计算给出：

$$Te(\beta ei, \lambda 0, \lambda 1, \lambda 2) \overset{定义}{=} \arg\min_{\beta i \in R}\ \frac{1 + 2\lambda 2}{n} \left(\beta ei\ \beta i - \frac{1}{1+2\lambda 2}\right)^2 + \lambda 1|\beta i| + \lambda 01[\beta i\ 6= 0]o, \tag{8}$$

其中,（调整参数） {λi}提供了 Te(βei , 和 βei都是固定的,并且 Te(βei , λ0, λ1, λ2)是集值的。引理 2 λ0, λ1, λ2)的明确表征。

引理 2. (单变量最小化) 设 Te 为 (8) 中定义的阈值算子。然后，

$$
T_e(\tilde{\beta}_i, \lambda_0, \lambda_1, \lambda_2) = \begin{cases} \text{符号}(\tilde{\beta}_i)\dfrac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} & \text{如果 } \dfrac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} > \sqrt{\dfrac{2\lambda_0}{1+2\lambda_2}} \\ \{0\} & \text{如果 } \dfrac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} < \sqrt{\dfrac{2\lambda_0}{1+2\lambda_2}} \\ \left\{0, \text{符号}(\tilde{\beta}_i)\dfrac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2}\right\} & \text{如果 } \dfrac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} = \sqrt{\dfrac{2\lambda_0}{1+2\lambda_2}} 。\end{cases}
$$

证明。令 g(u) 表示 (8) 中最小化的目标函数，即

$$
g(u) := \frac{1+2\lambda_2}{2}\left(u - \frac{\tilde{\beta}_i}{1+2\lambda_2}\right)^2 + \lambda_1|u| + \lambda_0 1[u \neq 0]。
$$

如果 $|\tilde{\beta}_i| > \lambda_1$，则 $\min_{u\neq 0} g(u)$ 由 $\hat{u} = (|\tilde{\beta}_i| - \lambda_1)\,\text{符号}(\tilde{\beta}_i)/(1+2\lambda_2)$ 得到（这就是众所周知的软阈值运算符）。现在，$g(\hat{u}) < g(0)$ 等价于 $\frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} > \sqrt{\frac{2\lambda_0}{1+2\lambda_2}}$。因此，$\hat{u}$ 是 $g(u)$ 的最小化，当 $\frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} > \sqrt{\frac{2\lambda_0}{1+2\lambda_2}}$。如果 $\frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} = \sqrt{\frac{2\lambda_0}{1+2\lambda_2}}$，则 $\hat{u}$ 和 0 都是 g(u) 的极小值。如果 $\frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} < \sqrt{\frac{2\lambda_0}{1+2\lambda_2}}$，函数 g(u) 在 u = 0 处最小化。这样就完成了最后证明时。 □

引理 2 和定义 2 在引理 3 中提供了 CW 最小值的明确表征，并为所有 i ∈ [p]定义了 $\tilde{\beta}_i = \langle y - \sum_{j\neq i} X_j\beta_j, X_i\rangle$。然后，引理 3. 令 $\beta \in \mathbb{R}^p$，β 是一个CW 最小 iff

$$
\beta_i^* = \text{符号}(\tilde{\beta}_i)\frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} \text{ 和 } \left|\beta_i^*\right| \geqslant \sqrt{\frac{2\lambda_0}{1+2\lambda_2}}，\text{对于每个 } i \in \text{Supp}(\beta^*)
$$
$$
\text{和 } \frac{|\tilde{\beta}_i|-\lambda_1}{1+2\lambda_2} \leqslant \sqrt{\frac{2\lambda_0}{1+2\lambda_2}} \text{ 对于每个 } i \notin \text{Supp}(\beta^*)。 \tag{9}
$$

比较 (9) 和表征 (6) 的平稳解，我们看到平稳解的类包含 CW 最小值类，并且对于一般 X 的包含是严格的。

2.3 交换不可避免的最小值

我们现在考虑使用局部组合优化的概念进一步细化 CW 最小值类的固定解决方案。给定一个 CW 最小值 β，人们可能会考虑通过“交换”操作获得另一个目标较低的候选解，如下所述：我们将 β 中的一些非零值设置为零，并将一些原来为零的 [illegible] 化新支持：

为非零。

- 部分优化：这里我们只优化从支持外部添加的坐标 这会导致部分交换不可避免的最小值，如第 2.3.1 节所述。
- 全面优化：在这里我们优化了新支持中的所有坐标 这导致到完全交换不可避免的最小值，如第 2.3.2 节所述。

如果与 β 相比，得到的解决方案导致更低的目标，则从 β 到更好的解决方 ，那么我们已经成功逃脱，无法通过上述策案。如果 β [illegible] 的最小值，其中他们的

使用部分优化对单个坐标进行交换操作。然而,这里研究的问题,即问题 (3) 是不同的 我们认为是L0 惩罚版本并且 f(β) 是非平滑的。此外,我们的交换不可避免的最小值类允许交换多个坐标。我们还允许对子问题进行部分和完全优化。

2.3.1 部分交换不可避免（PSI)最小值

我们正式介绍了部分交换不可避免 (PSI) 最小值。换句话说,这些固定解决方案（或最小值)不能通过交换任何两个坐标子集（从支持内部和外部)和对新支持执行部分优化（即,我们仅优化添加到支持的新坐标)来逃避）。我们回想一下,对于任何 L ⊆ [p],符号 U Lβ 表示如果 i ∈ L 且 (U Lβ)i = 0 如果 i /∈ L 则具有 i 坐标 (U Lβ)i = βi的向量。

定义 3. (PSI 最小值) 令 k 为正整数。具有支持 S的向量 β 是 k 阶的部分交换不可避免的最小值,用 PSI(k) 表示,如果它是一个平稳解并且对于每个S1 ⊆ S 和S2 ⊆ S c

，使得|S1| ⩽ k 和|S2| ⩽ k,以下成立

$$F(\beta^{*}) \leqslant \min_{\beta S2} F(\beta^{*} - U_{S1}\beta^{*} + U_{S2}\beta)。$$

以下引理描述了 1 阶 PSI 最小值,即 PSI(1)。

引理 4. 一个向量 β ∈ R p是一个 PSI(1) 最小值 iff

$$\beta^{*}_{\cdot} = \text{符号}(\beta ei)\,|\beta ei| - \lambda 1 \quad \text{和} |\beta^{*}$$

$$| \geqslant \text{最大}\ (q\ 2\lambda 0^{\frac{}{1+2\lambda 2}}, \quad \text{最大限度}\ \frac{|\beta eij| - \lambda 1}{1+2\lambda 2}), \text{对于 } i \in \text{Supp}(\beta)$$

和

$$\frac{|\beta ei| - \lambda 1}{1+2\lambda 2} \leqslant q\ 2\lambda 0^{\frac{}{1+2\lambda 2}}, \quad \text{对于 } i /\in \text{Supp}(\beta)$$

其中 βei = hy - P j6=i Xjβj , Xi和βeij = hy - P l6=i,j Xlβl ， Xj i。

证明。结果可以很容易地从引理 2 和定义 3 推导出来。 □

引理 3 和 4 表明,与 CW 最小值相比,PSI(1) 最小值对非零系数的大小施加了额外的限制。我们还注意到,CW 最小值类包含任何 k 的 PSI 最小值。此外,随着 k 的增加,PSI(k) 最小值的类变得更小 直到它与问题 (3) 的全局最小值类一致。 4.1 节介绍了一种算法,该算法结合了基于 MIO 的坐标下降和局部组合优化,以实现任何给定 k ∈ [p] 的 PSI(k) 最小值。

2.3.2 全交换不可避免 (FSI) 最小值

我们定义了完全交换不可避免 (FSI) 最小值 它们与 PSI 最小值的不同之处在于,在交换坐标后允许对新支持进行全面优化。

定义 4. (FSI 最小值) 令 k 为正整数。具有支持 S的向量 β 是 k 阶 FSI 最小值,用 FSI(k) 表示,如果对于每个S1 ⊆ S 和 S2 ⊆ S c使得|S1| ⩽ k 和|S2| ⩽ k,以下成立 ,

$$F(\beta^{*}) \leqslant \text{最小值}_{\beta(S\backslash S1)\cup S2} F(\beta^{*} - U_{S1}\beta^{*} + U_{(S\backslash S1)\cup S2}\beta)。$$

我们注意到,对于一个固定的 k,FSI(k) 最小值类包含在 PSI(k) 最小值类中（这是定义的结果）。随着 k 的增加,FSI(k) 最小值的类变得更小,直到它与问题 (3) 的全局最小值集重合。 4.2 节介绍了一种将坐标下降与 MIO 相结合以生成 FSI(k) 最小值的算法。

2.4 迭代硬阈值 (IHT) 激发的平稳性

近端梯度类型算法（如 IHT)广泛用于L0 约束和L0惩罚最小二乘问题 [6]。考虑与 IHT 相关的固定解决方案类别并研究它们与 CW 最小值的比较是很有见地的。令fd(β) := λ2kβk ∇fd(α)k ⩽ Lkβ - αk 对于所有 β, α ∈ R p 。然后得出[28]

$\frac{1}{2}$ky - Xβk ^{2} +

^{2}. fd(β)的梯度,即∇fd(β)是Lipschitz,参数为L（比如）,即k∇fd(β) -

$$QL(\beta;\alpha) := \frac{大号}{2}k\beta - \alpha k^2_2 + h\nabla fd(\alpha), \beta - \alpha i + fd(\alpha) \geqslant f(\beta) \qquad \forall \beta, \alpha \in R\,p\cdot$$

如果 表示第 k 次迭代时 β 的值,然后,为了获得第 (k+1) 次迭代,IHT 最小化
β 是问题 (3) 形式的上界： QL(β; β 更新序列：)+λ1kβk1+λ0kβk0。这导致以下情况

$$\beta^{k+1} \in \arg\min_{\beta\in Rp} \frac{1}{2\tau} k\beta - (\beta - \tau\nabla fd(\beta))k^2 + \lambda 1k\beta k1 + \lambda 0k\beta k0\,, \tag{10}$$

其中,τ > 0 是固定步长。如果 β 到 β = α ,我们说 α ∈ R p是更新 (10) 的不动点。这也为问题 (3) 定义了另一个平稳性概念,它不同于前面描述的CW最小值。因此,我们将以下定理1建立的收敛性上的不动点 = α 导联 k+1 定义如

$$\beta^*_i = 符号(\beta ei)\,|\beta ei| - \lambda 1 \frac{1}{1+2\lambda 2} \quad 和\ |\beta| \geqslant \sqrt{2\lambda 0\tau} 对于\ i \in Supp(\beta)$$

$$和 \quad \frac{|\beta ei| - \lambda 1}{1+2\lambda 2} \leqslant q\ 2\lambda 0 (1+2\lambda 2)\ 2\tau \quad 对于\ i \notin Supp(\beta) \tag{11}$$

其中 βei = hy - P j , Xii。提供 Xjβ 定理的证

明,因为它类似于 [24, 5] 的证明（考虑了问题 (3) 的基数约束版本） 另见 [21] 的证明当目标可微时,IHT 用于基数约束优化问题。表征 (11) 提出了一类受 IHT 启发的最小化器,定义如下。

定义 5. 向量 β 是问题 (3) 的 IHT 最小值,如果它满足 (11) 的 τ < $\frac{1}{升}$。

下面的注释表明 IHT 极小类包含 CW 极小族。

备注 2. 设 M 为XT X 的最大特征值,取 L = M+2λ2。根据定理 1,任何 τ < 确保更新 (10) 的收敛。此外,由于 X 的列被归一化,我们有 M ⩾ $\frac{1}{M+2\lambda}$ 1。比较表征（11）(IHT 最小值)与表征（9）(CW 最小值)我们看到 IHT 最小值的类别包括 CW 最小值的类别.实际上,通常对于高维问题,M 远大于 1,使得 IHT 最小值的类比 CW 最小值大得多（参见第 6.2 节的数值示例）。

下面我们总结了本节介绍的固定解的层次结构。

$$\text{等级制度} \qquad \underset{\text{极小值}}{\text{FSI(k)}} \subseteq \underset{\text{极小值}}{\text{PSI(k)}} \subseteq \underset{\text{极小值}}{\overset{\text{顺时针}}{\text{}}} \subseteq \underset{\text{极小值}}{\text{IHT}} \qquad \underset{\text{解决方案}}{\text{固定}\subseteq} \tag{12}$$

最后,我们注意到当 k 很大时,FSI(k) 最小值和 PSI(k) 最小值的类与问题 (3) 的全局最小值一致。第 2 节介绍了问题 (3) 的平稳条件的分层类。我们现在讨论收敛到这些固定类的优化算法:CW 最小值、PSI 最小值和 FSI 最小值。第 3 节讨论了保证收敛到 CW 最小值的坐标系算法;第 4 节讨论了到达对应于 FSI(k)/PSI(k) 最小值的静止点的局部组合优化算法。

3 循环坐标下降:算法收敛

在本节中,我们介绍了循环 CD 的一种变体,它对每个坐标执行完全最小化 [3]。我们分析了它的收敛行为 特别是,我们证明了一个新的结果,该结果建立了一个独特的 CW 最小值(取决于初始化)和渐近线性收敛速度。我们注意到,如果我们避免完全最小化并使用保守的步长,则由于每次坐标更新后目标值都充分减小,收敛的证明变得简单明了。但是,对问题 3 使用具有保守步长的 CD 会对解决方案质量产生不利影响。实际上,通过表征固定点,可以证明次优步长会导致一类包含 CW 最小值的平稳解。虽然之前已经使用具有最小二乘数据保真度项的连续正则化器 [23, 7] 研究了循环 CD,但据我们所知,对问题 (3) 的循环 CD(以及相关的收敛性分析)的研究是新颖的。

回想一下,循环 CD 按照先验指定的 {1, 2, . . . , p}。在深入研究我们的算法的正式处理之前,我们简要讨论为什么 CD,特别是循环 CD,似乎非常适合我们的问题 尤其是考虑到它在我们的实验中具有出色的计算性能。

为什么选择循环 CD? Cyclic CD 实际上已被证明是 Lasso [10] 和连续非凸正则化器(如 MCP、SCAD 等)[23、7] 中最快的算法之一。循环 CD 中的坐标更新成本低,并且可以利用稀疏性 特别是通过稀疏残差更新和活动集收敛[10]。这使得它非常适合处理 np 和 p 数量级为数万到数百万的高维问题。另一方面,对于具有相似大小的问题实例,需要评估完整梯度的方法(例如,近端梯度下降、贪婪坐标下降等)的计算成本更高。

例如,近端梯度下降方法不利用基于稀疏性的结构以及基于 CD 的方法 [10, 27]。我们还注意到,根据我们的经验实验,随机 CD 在实践中似乎表现出较慢的收敛,(见第 6.2 节) 另见 [2] 对凸问题的相关讨论。

为了补充早期研究中报告的上述计算优势,我们的数值经验表明,循环 CD 在优化目标(第 6.2 节)和统计性能(参见第 6.3、6.4 和 6.6 节)方面都优于竞争算法。 (L0)、 (L0L1)和(L0L2)问题的解决方案通常预计会更少

5这样的观察出现在建立 IHT 类型算法的收敛性时 例如,参见 [1, 21, 5]

3.1 收敛性分析

6我们注意到,这个约定是出于技术原因使用,是我们证明定理 2 所必需的

7由于技术原因引入了间隔台阶;我们的 CD 收敛证明依赖于此来确保
算法限制点的平稳性。

算法 1:带间隔台阶的坐标下降 (CDSS)

输入:初始解 β k ← 0 while Not ^{0}, 正整数 C
Converged do for i in 1 to p do

$\beta^{k+1}_{k+1} \leftarrow \beta$
β minβi∈R F(βi),使用 (14) k ← k + 1 Count[Supp(β // 非间隔步骤
if Count[Supp(β
)] ← 计数[补充(β)] + 1
)] = Cp 然后 ←
β^{k+1} SpacerStep(β)] = 0) // 间隔步骤
Count[Supp(β k
← k + 1 返回 (β)

子程序 1:SpacerStep(β)

输入: β
for i in Supp(β) do
βi ← arg minβi∈R f(β1, . . . , βi , . . . , βp) using (14) with λ0
= 0 return (β)

引理 5. 算法 1 是下降算法,F(β) ↓ F 对于某些 F * ⩾ 0。

证明。如果在间 是非间隔步骤的结果,则 F(β) ⩽ F(β $^{k-1}$) 根据定义成立。如果 β)。由于间隔隔步骤之后获得 β,则 f(β 的 β 的支持大小) ⩽ F(β 是) ⩽ f(β $^{k-1}$ 台阶不能增加非递增的并且有界低于 $^{k-1}$(零,这意味着 kβ k0 ⩽ kβ k0,因此 F(β $^{k-1}$) 。由于 F(β) *

□

对于本节的其余部分,我们将做出以下小假设,以显示(L0)和(L0L1)问题收敛到唯一极限。 (L0L2)问题不需要这样的假设。

假设 1. 设 m = min{n, p}。 X 中的每 m 列是线性独立的。
假设 2. (初始化)如果 p > n,我们假设初始估计 β 满足 ^{0}

- 在(L0)问题中:F(β ^{0}) ⩽ λ0n。
- 在(L0L1)问题中:F(β ^{0}) ⩽ f(β $^{\ell 1}$) + λn 其中 f(β $^{\ell 1}$) = minβ $\frac{1}{2}$ky - Xβk ^{2} + λ1kβk1。

下面的评论表明假设 2 是相当小的。

备注 3. 假设 p > n 且假设 1 成立。对于(L0)问题,令 S ⊆ [p] 使得 |S| = n。如果β ^{0} 定义为 β = 0;然后 F(β) = λ0n 是最小二乘解决方案,其余为零,则满足假设 2。对于(L0L1)问题,我们注意到始终存在最优 lasso 解 βb 使得 kβbk ⩽ n(例如,参见 [34])。因此,βb 满足假设 2。

在下文中,我们假设假设 1 和 2 适用于(L0)和(L0L1)问题,并且我们不对(L0L2)问题做任何假设。

下面的引理表明,在(L0)和(L0L1)问题中,由算法 1 获得的任何 β 的支持大小都不能超过 n 和 p 的最小值。

引理 6. 对于(L0)和(L0L1)问题,对于所有 k,{β k} 满足 kβ kk0 ⩽ min{n, p}。

证明。如果 p ⩽ n,则结果很简单。假设 p > n。在(L0)问题中,假设 2 表明 F(β
) ⩽ λ0n。由于算法 1 是下降法(由引理 5),我们有 F(β)
这意味着对于所有 k,f(β kβ kk0 ⩽ n。类似地,对于(L0L1)问题,假设 2 和 [illegible] 对于每个 k, λ0n ,

暗示 F(β) ⩽ f(β `1) + λ0n可以等效地写为 f(β ) ⩾ 0,这导致 kβ kk0 ⩽ n。 ) − f(β `1
但是 lasso 解的最优性意味着 f(β) − f(β `1 □

以下引理表明算法 1 生成的序列是有界的。

引理 7. 序列 {β k} 是有界的。

证明。对于(L0L1)和(L0L2)问题,对于所有 k,β R p | F(β) ⩽ F(β 有界,因此, 属于水平集 G = {β ∈ 是一个初始解。因为在这
{β k} 是有界的。)} 其中β 两种情况下 F(β) 都是强制的,所以 G 是

我们现在研究L0问题。首先,如果 p ⩽ n,则(L0)问题的目标函数是强制性的(在假设 1 下),并且适用于(L0L1)/(L0L2)的先前参数。
否则,假设 p > n。回想一下引理 6,对于所有 k ⩾ 0,我们有 kβ kk0 ⩽ n;并且根据假设 2,我们有 F(β 每 k 个。因此,可以得出
β) ⩽ λ0n。此外,由引理 5,我们有 F(β) ⩽ λ0n为
∈ A 其中,

$$\text{一个} = [\quad \frac{1}{2}\, ky - XS\beta Sk \quad \leqslant \lambda n,\ \beta Sc = 0\}。$$

请注意,在每个AS 中, β 的唯一可能非零分量在βS 中。假设 ⩽ λn} ⊆ R |S|是有界的,这意味着AS是有界的。 1、水平集{βS|
− ky - XSβSk 由于 A 是有限数量的有界集合的并集,因此它也是有界的。
□

下一个引理描述了算法 1 的极限点。证明在 A 节。

引理 8. 令 S 为由非间隔步骤无限频繁生成的支持,令 {β l}l∈L为由 S 生成的间隔步骤序列。那么,以下成立:

1. 存在一个整数 N,使得对于所有 l ∈ L 和 l ⩾ N,我们有 Supp(β 2。存在一个 {β l}l∈L的子序列,它) = S。

收敛到一个平稳解 β ,其中, β 小号
是 minβS f(βS)和 β 的唯一极小值 Sc = 0。

3.支持 S的 {β k}k⩾0的每个子序列都收敛到 β (如上文第 2 部分所述)。

4. β 满足 |β | ⩾q $\sqrt{2\lambda 0}$ 对于 S 中的每个 j。

引理 9 表明,对应于 {β k} 的任何极限点的支持无限频繁地出现。

引理 9. 设 B 为 {β k} 的极限点,Supp(B) = S,则 Supp(β many k.) = S 表示无限

证明。我们通过使用矛盾论证来证明这个结果。为此,假设支持 S 只出现有限多次。由于只有有限多个支持,{β k} 的支持 } 满足:
Supp(β S ; 和 → ∞。但是,这在引理 8 的第 3 部分是不可能的。
与 S 6=S;和一个子序列 {β) = 小号 对于所有 k
β → B 为 k □

引理 10 是技术性的,在算法 1 的收敛性证明中将需要。证明在 A 节中。

引理 10. 令B(1)和B(2)是序列 {β k} 的两个极限点,分别支持S1和S2 。假设S2 = S1 ∪ {j} 对于一些 j /∈ S1。然后,恰好满足以下条件之一:

1. 如果存在 i ∈ S1使得 hXi , Xj i 6= 0,则 |B | > q 2λ0 | =
2. 否则,如果 hXi , Xj i = 0 对于所有 i ∈ S1,则 |B q 2λ0 1+2λ2 。此外,对于 Supp(β) = S2的 β ∈ R p ,我们有 |T(βej , λ0, λ1, λ2)| = q 2λ0

当在算法 1 期间将非零坐标设置为零时,以下建立了目标值减少的下限。

引理 11. 令 β 为将坐标 用 β j迭代算法 1 对于某些 j ∈ [p],6= 0。让β 对应 j 更新为 0 的非间隔步骤,即 β j = 0。那么,以下成立:

$$F(\beta) - F(\beta k+1) \geqslant \frac{1 + 2\lambda 2}{2} |\beta| - r\, 2\lambda 0\, 1 + 2\lambda 2\, !2\,。 \quad (15)$$

证明。 F(β) − F(β k+1)可以通过注意到 β 来简化 = β 对于所有 i 6= j 和 β j = 0:

$$F(\beta) - F(\beta k+1) = -\beta ek\, \beta j + j \frac{1 + 2\lambda 2}{}(\beta) + \lambda 0 + \lambda 1|\beta j|$$

$$\geqslant -|\beta ek||\beta\ 2\lambda 2 | + (\beta 2) + \lambda 0 + \lambda 1|\beta|$$

$$\geqslant -|\beta (|\beta ek| - \lambda 1) + (\beta \frac{1 + 2\lambda 2|}{2}) + \lambda 0, \quad (16)$$

其中 βek k+1是一个非间隔步骤,将坐标 j 设置为 并分解,我们得到引理的结果。由于 β 阈值算子 (14) 的定义意味着 |βek | − λ1 ≤ p 2λ0(1 + 2λ2),将这个界限代入 (16) □

最后,下面的定理(证明见 A 节)确定了算法 1 收敛到唯一的 CW 最小值。

定理 2。以下适用于算法 1:

1. {β k} 的支持在有限次迭代后稳定,即存在一个整数 m 和一个支持 S 使得 Supp(β) = S 对于所有 k ⩾ m。
2. 序列 {β k} 在 Supp(B) = S 的情况下收敛到 CW 最小值 B。

0 j 0 0 ≠0 j j j j j(2) 1+2λ2 (2) 2 k+1 j 1+2λ2 ≠0 j 0 k+1 k+1 0

3.2 收敛速度

在本节中,我们将展示算法 1 表现出渐近线性收敛速度。
定理 2 意味着迭代的支持在有限数量的迭代中稳定。在支撑稳定到支撑 S（比如说)之后,间隔和非间隔步骤导致相同的坐标更新。因此,算法 1 可以看作是一个循环 CD（每个坐标都有完全优化）,其中我们循环应用算子:T(βei , 0, λ1, λ2)定义在 (14) 中,对于每个 i ∈ S。

我们将在第 3.2 节中使用一些新的符号来解释该理论 术语全循环将指代原版 CD 在 S 中的所有坐标上的单次通过。我们使用 β 表示在执行 K 个全循环后生成的迭代光盘。

定理 3。假设与定理 2 相同的假设成立。令 {β K} 为算法 1 生成的全周期迭代,B 为支持 S 的极限。令mS和MS分别表示XT XS的最小和最大特征值。然后,有一个整数 N,使得对于所有 K ⩾ N,以下成立:

$$\frac{F(\beta^{K+1}) - F(B)}{F(\beta^{K}) - F(B)} \leqslant 1 - \frac{2(\text{毫秒} + 2\lambda_2)}{1 + |S|\left(\frac{MS+2\lambda_2}{1+2\lambda_2}\right)^2} \tag{17}$$

证明。根据定理 2,我们有 β 和 Supp(B) → B 和 ∃M 使得对于所有 K ⩾ M,我们有 Supp(β K) = S) = 与 = S。因此,存在整数 N ⩾ M 使得对于 K ⩾ N,sign(β sign(Bi)对于每个 i ∈ S。对于 K ⩾ N,它可以很容易地看出,通过 。最小化以下目标生成的迭代 β

$$g(\beta_S) = \frac{1}{2}\|y - X_S\beta_S\|^2 + \lambda_1 \sum_{i\in S, B_i>0} \beta_i - \lambda_1 \sum_{i\in S, B_i<0} \beta_i + \lambda_2\|\beta_S\|^2, \tag{18}$$

1使用带步长的坐标下降并从初始解开始 β 1+2λ2βS7→ g(βS)是连续可微的,其梯度是 Lipschitz 连续ñ 。功能
的,参数 L=MS+2λ2。此外,它是强凸的 [28],具有强凸参数σS = mS +2λ2。 [2]（见定理 3.9)证明了当应用于强凸和连续可微函数时,循环 CD 的线性收敛速度。在我们的上下文中应用 [2] 的结果可以得出定理的结论。

□

4 局部组合优化算法

受第 2.3 节中介绍的 Swap Inescapable minima 类的启发,我们提出了算法来实现属于这些类的解决方案。

4.1 PSI 最小值的算法

我们介绍了一种导致 PSI(k) 最小值的算法。在第 th 次迭代中,算法执行两个步骤:1) 运行算法 1 以获得 CW 最小值 β 和 2) 通过解决以下组合优化问题找到“下降移动”： ,

$$\min_{\beta,S1,S2} F(\beta^{-US1\setminus\beta + US2}) \text{ st } S1 \subseteq S,\ S2 \subseteq S^{c},\ |S1| \leqslant k,\ |S2| \leqslant k \tag{19}$$

其中,S = Supp(β F(βb) `）。请注意,如果存在满足问题 (19) 的可行解 βb,则 βb 可能不是 CW 最小值。为此,βb可用于 < F(β 算法 1 以获得问 `化
题 (3) 的更好解。否则,如果 βb 不存在,则 β 是 PSI(k) 最小值,因为它满足定义 3 . 算法 2（又名 CD-PSI(k))总结了该算 `
法。

算法 2:CD-PSI(k) βb0 ← β for

` = 0, 1, 0 . . do ← 用 βb 初始化的算法 1 的输出`

β `+1

如果问题 (19) 有一个可行解 βb 满足 F(βb) < F(β` +1)则

别的 βb`+1 ← βb

停止

返回β

定理 4。令 {β `} 为算法 2 生成的迭代序列。对于(L0)和(L0L1)问题,假设假设 1 和 2 成立。然后,算法 2 以有限次数的迭代终止,输出为 PSI(k) 最小值。

证明。算法 2 导致支持 S 上的序列 {β i} β CW 最小值，` 使得 F(β `) < F(β `-1 ) < · · · < F(β 0）。因为，对 `β 以 1, 问题是否都是算 in 1 的输出,这都是 CW 最小值（根据定理 2）。任何的凸性,支持 S 上的所有静止解都具有相同的目标（因为它们都对应） 6= Supp(β j) 对于任何 i, j ≤ ` 使得

到 minβS f(βS)) 的最小值。因此,我们有 Supp(β i 6= j。因此,在算法
2 的过程中支持最多出现一次。由于可能的支持的数量是有限的,我们得出结论,算法 2 在有限次数的迭代中终止。最后,我们注意到算法 2 终止,如果 (19) 没有可行解 βb 满足 F(βb) < F(β 是 (19) 的最小值,因此是 PSI(k) 最小值（根据定义 3）。

`）。这意味着 β `

□

我们现在讨论用于计算组合优化问题 (19) 的解决方案的公式和算法。

问题 (19) 的 MIO 公式:问题 (19) 承认混合整数二次优化

化 (MIQO) 公式由下式给出：

$$\min_{\theta,\beta,z} \quad f(\theta) + \lambda_0 \sum_{i \in [p]} z_i \quad (20a)$$

$$\text{s.t. } \theta = \beta - \sum_{i \in S} e_i \beta_i (1 - z_i) + \sum_{i \in S^c} e_i \beta_i \quad (20b)$$

$$-Mz_i \leqslant \beta_i \leqslant Mz_i, \quad \forall i \in S^c \quad (20c)$$

$$\sum_{i \in S^c} z_i \leqslant k \quad (20d)$$

$$\sum_{i \in S} z_i \geqslant |S| - k \quad (20e)$$

$$\beta_i \in \mathbb{R}, \forall i \in S^c \quad (20f)$$

$$z_i \in \{0, 1\}, \forall i \in [p], \quad (20g)$$

变量是 θ ∈ R p , βi ,其中 β 是固定的,M 是控制 βSc 的导致问题 (20) 的解决方案发生在 (20) 中的 MIO 求解器的运行时间),式表示。因此,问题 (20) 是一个具有线性约束和凸目标的 MIQ 优化问题,通过线性不等

我们现在解释问题 (20) 中的约束以及它们与问题 (19) 的关系。为此， − U S1 + U S2令S1和S2为 (19) 中定义的子集。令删除 β，并且我们有θi = 0（参见（20b））。如果zi = 1,则 β（见（20b））。注意|S1| θ = β 并且如果在 (20e) 中对于 i ∈ S,我们考虑任何

没有从 θ 中删除,我们有θi = β 6=

i∈S 0 (1−zi) = |S| -P i∈S zi 。条件|S1| ⩽ k,因此被编码

在约束 P i∈S zi ⩾ |S| - (20e) 中的 k。因此我们有kθSk0 = P i∈S zi 。

任何二元变量zi ,其中 i ∈ S c βi在 [−M,M] 中自由变化。这意味着θi = βi,那么如果zi = 0,则我们观察到。现在考虑 i∈Sc zi = |S2|,并且约束|S2| ⩽ k 表示为 (20d) 中的 P i∈Sc zi ⩽ k。注意到目标 (20a) 中的函数是 F(θ),因为λ0 P i∈ [p] zi = λ0kθk0。

备注 4. 我们注意到,与完整问题 (3) 的 MIO 公式相比,MIO 问题 (20) 的搜索空间大大减少（组合）。因此,对于较小的 k 值,求解 (20) 通常比问题 (3) 快得多。此外,请注意,我们使用 MIO 框架 (20),因此它可以快速提供目标比当前解决方案更小的可行解决方案 与通过匹配对偶建立最优性相比,通常可以在非常短的运行时间内实现界限。为此,如果不存在具有较小目标值的可行解决方案,则 MIO 框架可以通过对偶边界来证明不存在。

第 6 节提供了示例,其中上述 MIO 框架在获得的解决方案的质量方面导致更高质量的解决方案 - 从优化和统计性能的角度来看。

第 4.1.1 节讨论了上述 MIO 公式的一个特例,其中 k = 1,我们可以推导出解决问题 (19) 的有效算法。

子程序 3:问题 (19) 的有效实现,k = 1。

S ← Supp(β)
for i ∈ S do for
j ∈ S c do
计算 v 在 O(1) 中使用 (25)
|
如果 |v * | > |β |然后
βb ← β − eiβ
休息

备注 5. 由于 CD-PSI(1)（算法 2,k = 1)计算效率高,在算法 2（k > 1)中,CD-PSI(1) 可用于代替算法 1。在我们的数值实验中,发现这在较低的运行时间和获得更高质量的解决方案（就客观价值方面)方面效果很好。此修改还保证收敛到 PSI(k) 最小值（因为定理 4 的证明仍然适用于此修改版本）。

4.2 FSI 最小值的算法

为了获得 FSI(k) 最小值,需要修改问题 (19) 我们将优化变量 U S2 β 替换为 U (S\S1)∪S2 β。这会导致以下问题： + U (S\S1)∪S2 β) st S1 ⊆ S, S2 ⊆ S

最小 β,S1,S2 F(β − U S1 β , |S1| ⩽ k, |S2| ⩽ k, (29)

其中,S = Supp(β 问题)。类似地,算法 2 通过考虑问题 (29) 而不是 (19)。通过定理 4 的证明中使用的相同论点,这种修改保证了算法 2 在有限次数的迭代中收敛到 FSI(k) 最小值。

问题 (29) 可以表示为 MIQO 问题。为此,问题 (20) 需要在 (20c)、(20b) 和 (20f) 行中进行修改,并具有以下约束：

θ = β - P (1 − zi) + P i∈SC eiβi + P = [p] i eiβi

−Mzi ⩽ βi ⩽ Mzi , ∈ S ∪ S c

i ∈ S ∪ S c = [p]

通过上述修改,问题（29)可以表示为以下 MIQO 问题：

最小 θ,z f(θ) + λ0 X (30a)

st − Mzi ⩽ θi ⩽ Mzi , ∀i ∈ [p] (30b)

X (30c)

X zi ⩾ |S| - k (30天)

(30e)

← argmax |v i * βi ∈ e v* i∈S zi∈ {0,1}, ∀i ∈ [p] i∈S

换句话说,上述公式从当前支持 S 中删除由S1索引的变量,添加与索引S2 ∈ S c对应的变量,然后在新支持上优化 (S \ S1) ∪ S2 – 坐标的选择通过不等式表示对二元变量的约束,出现在问题 (30) 中。问题 (30) 有 p 个二元变量和 p 个连续变量。

备注 6. 与 PSI 最小 ima 的公式 (20) 相比,公式 (30) 具有更大的搜索空间,这是由于连续变量的数量增加所致。与配方 (20) 相比,这导致运行时间增加。然而,我们注意到对于问题 (3),这个公式的求解速度明显快于 MIO 公式（原因与备注 4 中讨论的相同）。

在第 6.5 节中,我们展示了将不同 k 值的 FSI(k) 最小值的质量与其他类别的最小值进行比较的实验。

5 正则化路径的高效计算

我们设计了 L0Learn:一个具有[8]R 接口的可扩展 C++ 工具包,实现了本文讨论的所有算法。与其他流行的稀疏学习工具包（例如,glmnet 和 ncvreg)相比,该工具包通过利用一系列计算技巧,例如延续、调整参数网格的自适应选择、活动集更新、（部分)贪婪循环,实现了更低的运行时间。坐标排序、相关筛选以及利用β中的最小二乘损失和稀疏性对浮点运算进行仔细计算。我们强调,根据我们的经验,发现上述计算启发式算法起着至关重要的作用 它们共同影响所获得解决方案的质量,并导致运行时间更快。我们注意到该工具包利用了快速线性代数库 Armadillo [31],它直接调用 BLAS（基本线性代数子程序）,从而显着加快了线性代数运算。下面我们对上述策略进行更详细的说明。

继续:从统计的角度来看,对于每个选择λ1、 λ2的正则化参数 {λ 网格,希望获得问题 (3) 的解决方案。如果λ0值排序 > λ2 i来初始化算法（都 > · · · > λm ,我们使用从 λ 0 0 获得的解。这有助于加速算法的收敛,也鼓励算法避免低质量平稳解。我们注意到我们不使用跨λ1、 λ2 的延拓;并且限制我们自己使用跨λ0 的延拓。

为:λ 0 1 0 0, 我+1

算法 1 和 2) 用于 λ

调整参数的自适应选择:在使用延续时,选择良好的λ0值网格非常重要。选择的网格可能取决于λ1、 λ2 的选择。如果λ 的i个连续值相距甚远,则可能会错过好的解决方案。另一方面,如果两个0太接近,它们可能会导致相同的解决方案（由于问题的组合性质）。为了避免这个问题,我们推导了选择λ0值的条件（据我们所知,这些条件是新颖的）,以确保算法 1 在使用延续时导致不同的解决方案。为此,我们提出以下引理,其中我们假设λ1、 λ2是先验固定的。

λ 的值 0

引理 12. 假设 β (i)是算法 1 的输出,其中λ0 = λ 0 .令 S = Supp(β (i)), r =

[8] https://github.com/hazimehh/L0Learn 9

我们还注意到问题 (3) 预计会导致解决方案比那些解决方案更稀疏（非零更少）可通过 Lasso 和 MCP 惩罚回归获得。这也有助于减少运行时间。

y − Xβ(i)表示残差,并且

$$Mi = 2(1 \frac{1}{+ 2\lambda 2)} \max_{j \in Sc} |hr, Xj \| - \lambda 1^{2}. \quad (31)$$

然后,针对 λ 运行算法 1 β (i+1) 6= β $^{我+1}_{0}$ 如果$_{0}$λ 在 β(Mi),以初始化导致解 β (i+1)满足： < λi并且 β (i+1) = β (i) (i) if λ $^{我+1}_{0}$ <密 , $^{我+1}_{0}$ $_{0}$]。

证明。让我们考虑这样的情况,λ $^{我+1}_{0}$ <弥。由 (31) 得出：

$$\max_{j \in Sc} \frac{| \|hr, Xj \| - \lambda 1|}{1 + 2\lambda 2} > s \, 2\lambda \frac{\overline{}}{1 + 2\lambda 2}, \quad (32)$$

这意味着对于给定的 λ,β (i)不是 CW 最小值（参见 (9)）。根据定理 2,算法 1 收敛到 CW 最小值。因此,用 β (i)初始化的算法 1导致 β (i+1) 6= β (i)

我们现在考虑这样的情况,λ $^{我+1}_{0}$ ∈ (Mi , λi$_{0}$]。那么 (31) 意味着

$$\max_{j \in Sc} \frac{| \|hr, Xj \| - \lambda 1|}{1 + 2\lambda 2} < s \, 2\lambda \frac{\overline{}}{1 + 2\lambda 2} \leqslant s \, 2\lambda \frac{\overline{}}{1 + 2\lambda 2}. \quad (33)$$

此外,由于 β (i)是 λ = λ 的 CW 最小值 $^{\rightarrow}_{0}$, 我们对于每个 j ∈ S

$$^{(一)}|\beta| \geqslant s \, 2\lambda \frac{\overline{}}{1 + 2\lambda 2} \geqslant s \, 2\lambda \frac{\overline{}}{1 + 2\lambda 2}, \quad (34)$$

其中,从 λ 不等式 (33) 和 (34) 得出的第二个不等式 $^{我+1}_{0}$ 我≤λ 0 .条件∇Sf(β) = 0 以及 $^{(一世)}_{小号}$ 意味着 β (i)是问题 (3) 的 CW 最小值,在λ0 = λ β (i)是算法 1 的不动点。 $^{我+1}_{0}$.所以,

□

引理 12 提出了一个简单的方案来计算调整参数的网格 {λ $^{\rightarrow}_{0}$ }。假设我们 = 计算了直到λ0 = λ的正则化路径,其中 α 是 (0, 1) 中的固定标$^{\rightarrow}_{0}$, 然后 λ 0 $^{i+1}$ 可以计算为 λ $^{我+1}_{0}$ αMi , 量。此外,我们注意到Mi （定义在 (31) 中)可以在没有显式计算 hr、 Xii的情况下计算每个 i ∈ S c因为这些点积可以在内存中维护,同时运行λ0 = λ 和因此i+1 的算法 1 λ , $^{\rightarrow}_{0}$.因此,计算Mi ,

0 ,只需要 O(|S c |) 操作。

（部分)贪心循环顺序:假设算法 1 用解 β let r 初始化 ^{0} 和 ^{0} = y - Xβ0 。在运行算法 1 之前,我们根据 , Xii|的排列对坐标进行排序对于 i ∈ [p] 以降序排列10。我们注意数量 |hr 在算法^{0} 到这个排序只执行一次 1 开始之前,这与贪婪 CD 不同,贪婪 CD 需要找到 |hr 的最大值 ^{0},十二|每次坐标更新后。

这种坐标排序鼓励算法 1 优先考虑与残差高度相关（绝对值)的坐标。此外,我们注意到当使用延续时,数量 |hr ^{0},十二|可以通过算法 1 保存在内存中

10由于 X 的列有单位`2-norm,更新索引 arg maxi |hr ^{0},十二|将导致最大的减少 在目标函数中。

在先前的λ0值（在网格中)计算解。而不是对 p 值进行完全排序 |hr 排序,其中仅对顶部 t 坐标进行排序,而其余的则保持其初始顺序。0, Xii|, i ∈ [p];我们观察到执行部分计算在计算上是有益的

可以使用基于堆的实现在 O p log(t) 操作中完成部分排序。在我们的实验中,我们发现将 t 设置为 p 的 5% 会导致类似于完全排序的结果,同时大大减少了计算时间。我们将此方案称为（部分)贪婪循环顺序。

在第 6.2 节中,我们将算法 1 与上述（部分)贪心循环顺序、算法 1 的普通版本（我们以先验固定顺序在 p 坐标上循环)和随机 CD [29] 进行比较。我们的研究结果似乎表明,本文提出的（部分)贪婪循环规则在运行时间和优化性能方面都具有显着优势。

相关筛选:当使用延续时,除了与当前残差高度相关的一小部分（例如,5%)其他坐标之外,我们通过将算法 1 的更新限制为支持热启动来执行筛选 [33] 这些高度相关的坐标很容易作为（部分)贪婪循环排序规则的副产品获得,如上所述。在筛选的支持上收敛后,我们检查支持外的任何坐标是否违反 CW 最小值的条件。如果存在违规,我们从当前解决方案重新运行算法。通常,从筛选集获得的解决方案结果是 CW 最小值,并且只在所有坐标上完成一次通过 - 这通常会减少整体运行时间。

活动集更新:根据经验,我们观察到算法 1 生成的迭代通常可以在不到 10 个完整周期内实现支持稳定。定理 2 进一步支持了这一点,它保证了支持必须在有限次数的迭代中稳定。如果支持在多个连续的完整周期中没有改变,那么我们将算法 1 的后续更新限制为当前支持。在受限支撑上收敛后,我们检查支撑外的任何坐标是否违反 CW 最小值的条件。如果有违规,我们再次运行从当前迭代初始化的算法 这将一直持续到我们达到 CW 最小值。这种启发式方法在减少计算时间方面非常有效,尤其是当 p n 时。

快速坐标更新:让 r 当前迭代 β βei = hr 可以 = y - P j∈[p] Xjβ更新坐标 i,我们需要在应用阈值算子 (14) 之前进使用以下任一利用稀疏性的算法更新 r：

+西β 我, Xii = 小时 , Xii + β 气

(i) 残差更新:我们维护残差 r 在整个算法中,我们计算（在更新 时),成本 O(n)

βei = hr , Xii + β 使用 O(n) 操作。一旦β k+1 是稀疏的,在算法过程中许多坐标保持为

ith 坐标),我们通过以下方式更新残差： r 操作。由于 β k+1 k ← r + Xi(β 气 - β k+1零， k+1 = 0 意味着

即 β 。因此,更新一个完整通道的残差需要花费 O((p − 通道的残差需要花费 O((p − t)n),其中 t 是在整个循环中保持为 0 的坐标数 注意 p ≈ |S|,其中 S 是在整个周期结束时获得的支持。 (ii) 物化更新:我们提出了另一种更新 βei 的方法,它不需要 。请注意,算法 1 在开始时为所有 i ∈ [p]计算 hy, Xii

使用预先计算的 r （使用延续
时仅执行一次）。如果坐标 ` 进入支持

11回想一下,一个完整的周期是指以循环顺序更新所有p坐标。

第一次,我们通过计算和存储所有 j ∈ [p]的数量 hX` , Xj i “实现 `”。
因此,在迭代 k + 1 中,我们利用支持的稀疏性计算 βei ,如下所示:

$$βei = hy, Xii -X \quad hXi , Xj iβ 。 \tag{35}$$

(35) 中的更新需要 O(kβ kk0) 次操作。该成本小于使用上述规则 (i) 计算 βei 的 O(n) 成本;假设 kβ kk0 < n。如果 β 6= 0 并且坐标 i 之前没有被物化,那么我们用成本 O(np) 物化它(源于点积计算)O(|S|np) 的成本,其中 S 是遇到的最大支持。此外,计算由物化新坐标产生的点积需要 O(|S|np) 的成本。然而,这些计算可以在算法期间存储,不需要在每次迭代中重复。

当算法 1 遇到的支持小于 n 时,方案 (ii) 很有用。它对于 PSI(1) 算法的有效实现也很有用,因为它存储了 (26) 中所需的点积。然而,当 CD 遇到的支持与 n 相比相对较大时(例如,n 的 10%),则方案 (i) 可以变得明显更快,因为可以使用对 BLAS 库的调用来加速点积计算。我们建议为上述两种方案保留一个选项,并选择运行速度更快的方案。

6 计算实验

在本节中,我们研究了我们提出的算法的优化和统计性能,并将它们与其他流行的稀疏学习算法进行了比较。为方便起见,我们提供了本节的路线图。

第 6.2 节介绍了我们提出的算法与 CD 和 IHT 的其他变体之间的比较。
第 6.3 节经验性地研究了我们提出的算法中可用的估计量与其他样本量不同的估计量的统计特性。第 6.4 节研究 SNR 变化时的相变。第 6.5 节对不同的 k 值在 PSI(k)/FSI(k) 算法中进行了深入研究。第 6.6 节介绍了实证研究,包括对一些大规模实例(包括真实数据集)的时序比较。

6.1 实验装置

数据生成:我们考虑了一系列针对各种问题规模和设计的合成数据集的实验。我们生成一个多元高斯数据矩阵Xn×p MVN(0, Σ)。
我们为 ∈ {0, 1} 和 β [p] 中的 k 个等间距的非零条目 † = 1 的β† 生成 ε,然后我们通过 (SNR) 来定义响应向量 (y = Xβ† + ε) 的实例:
, 在哪里 iid N(0, σ2)。我们定义信号

- 常数相关性:我们为每个 i 6= j 设置σij = ρ,对所有 i ∈ [p] 设置σii = 1。 |i-j|对于所有 i,j,约定00 = 1。
- 指数相关:我们设置σij = ρ

我们通过最小化单独验证集上的预测误差来选择调整参数,该验证集是在固定设计设置下生成的 y ,其中,
= Xβ † + iid N(0, σ2)。

i:βi 6=0 † 0 k+1

竞争算法和参数调整:除了我们提出的算法,我们在实验中比较了以下最先进的方法:

- Lasso:在图中我们用“L1”表示。
- 松弛套索:这是[17] 中考虑的松弛套索版本。给定 Lasso = γβlasso + (1 − γ)β 由支持的最小二乘解给出,并解 β lasso,松弛 Lasso 解定义为 β 轻松 β LS β 的非零分量,在实验中用 且 γ ∈ [0, 1] 是一个调整参数。我们使用自己的实现轻松套索 “L1Relaxed”表示。 套索
- MCP:这是[36] 的MCP 惩罚。我们使用 R 包“ncvreg”[7]提供的基于坐标下降的实现。
- 前向逐步选择:我们使用[17] 提供的实现,在实验中我们用“FStepwise”表示。
- IHT:我们使用我们的 IHT 实现,我们选择一个恒定的步长,它等于 为XT X的最大特征值的倒数。

对于所有涉及一个调整参数的方法,我们调整了 100 个参数值的网格,除了前向逐步选择,我们允许运行多达 250 个步骤。对于具有两个参数的方法,我们调整 100×100 值的二维网格。对于我们的算法,根据第 5 节生成调整参数λ0 。对于(L0L2)惩罚,如果 SNR ⩾ 1,我们在 0.0001 和 10 之间扫描λ2 ,如果 SNR ⩽ 1,我们扫描 100。对于(L0L1)惩罚,我们扫描λ1从 kXT yk∞下降到 0.0001 × kXT yk∞。对于 Lasso,我们将λ1从 kXT yk∞扫描到 0.0001 × kXT yk∞。对于松弛套索,我们使用与 Lasso 相同的λ1值,我们在 0 和 1 之间扫描 γ。对于 MCP,第一个参数 λ 的范围由 ncvreg 选择,我们在 1.5 和 25 之间扫描第二个参数 γ .

性能指标:我们使用以下指标来评估通过一种方法获得的解决方案的质量（例如,βb）。

- 预测误差:这与 [5] 中使用的度量相同,定义为

$$\text{预测误差} = \|X\beta b - X\beta\dagger\|_2 / \|X\beta\dagger\|^2.$$

  完美模型的预测误差为 0,空模型（β = 0)的预测误差为 1。

- L∞范数:这是估计误差的 L∞ 范数,即 kβb - β †k∞。 · 完全支持恢复:我们研究 β †的支持是否完全被 β 恢复,即 1[Supp(β †) = Supp(βb)],其中,1[·] 是指标函数 我们看此数量在多次复制中的平均值,从而估计完全支持的概率 恢复。
- True Positives:βb 和 β 的支持之间的共同元素的数量† · False Positives:βb 的支持中但不在 β · 的支持中的元素数量† · Support Size:βb 中非零的数量. · 目标:目标函数F(βb) 的值。 ·

6.2 CD变体与IHT的比较:优化性能

我们研究了不同算法的优化性能。我们研究 IHT 和 CD 的以下变体获得的解决方案的客观值:

- 循环CD:这是具有默认循环顺序的算法1。
- 随机 CD:这是随机版本的 CD,其中要更新的坐标为从 [p] 中均匀随机选择。这是 [29] 中考虑的版本。
- 贪心循环CD:这是我们提出的算法1,具有部分贪心循环排序坐标,在第 5 节中描述。

我们生成了一个具有指数相关性、ρ = 0.5、n = 500、p = 2000、SNR = 10 和支持大小 k † = 100 的数据集。我们生成了 50 个随机初始化,每个[illegible] Cyclic CD、Greedy Cyclic CD [illegible]的目标函数值以及直到收敛的迭代次数。对于随机 CD,我们在每次初始化时运行该算法 10 次,并对目标值和迭代次数进行平均。对于上述所有算法,当目标的相对变化 $< 10^{-7}$时,我们声明收敛。图 1 显示了结果。图 1 显示了贪婪循环 CD 产生的目标值

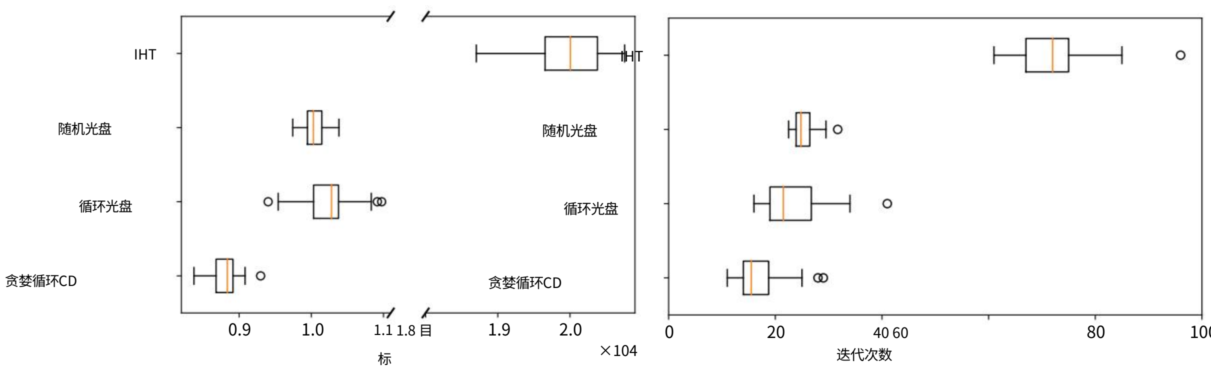

图 1:显示目标值分布和迭代次数的箱线图(这里,对所有 p 坐标的一次完整遍历定义为一次迭代)直到 CD 和 IHT 的不同变体收敛,对于每种算法,我们使用 50 个随机初始化(如文中所述)。箱线图的刻度代表四分位距的 1.5 倍。

明显低于其他方法;平均而言,我们的目标比随机 CD 提高了大约 12%,比 IHT 提高了 55%。根据我们在第 2.4 节中的讨论,可以部分解释这一发现,我们观察到 Lipschitz 常数 L 控制 IHT 返回的解的质量。在这种高维设置中,L ≈ 11 远非 1,因此 IHT 可能会陷入相对较弱的局部最小值。直到收敛的迭代次数也有利于贪婪循环 CD,它比随机 CD 需要大约 28% 的迭代,比 IHT 少 75% 的迭代。

6.3 不同样本数的统计性能

在本节中,我们研究不同的统计指标如何随样本数量而变化,而其他因素(p、 k † 、SNR、Σ)保持不变。我们考虑算法 1 和 CD-PSI(1)

(L0)、 (L0L1)和(L0L2)问题;除了 Lasso、Relaxed Lasso、MCP 和 Forward Stepwise 选择。在实验 1 和 2（如下）中,我们以 100 为增量在 100 和 1000 之间扫描 n。对于每个 n 值,我们生成了 20 个具有指数相关性的随机训练和验证数据集,p = 1000,k ＋ = 25 和 SNR =10。

6.3.1 实验一:高相关性

这里我们选择 ρ = 0.9　这是一个难题,因为样本中的特征之间的相关性很高。图 2 总结了结果。在图 2 的顶部面板中,我们展示了(L0)的算法 1、 (L0L2)的 CD-PSI(1)和其他竞争算法。在底部面板中,我们对所有情况(L0)、 (L0L1)和(L0L2) 的算法 1 和 CD-PSI(1) 的可用结果进行了详细比较。

从顶部面板（图 2）,我们可以看到 CD-PSI(1) (L0L2) （即算法 CD-PSI(1) 应用于(L0L2)问题)在所有考虑的方面实现了最佳性能测量和所有的 n 值。当 n 在 200 和 300 的范围内时,CD-PSI(1) (L0L2)完全恢复的概率急剧增加,并且在 n = 500（这是 p 的一半)时达到概率 1。

请注意,对于所有 n 值,Lasso 和 Relaxed Lasso 永远无法实现完全支撑恢复　并且相应的线与水平轴对齐。此外,即使在 n = p = 1000 时,MCP 和 FStepwise 的概率也小于 0.5　这表明它们在这种情况下未能正确地进行支撑恢复。 CD-PSI(1) (L0L2)明显占主导地位的预测误差和L∞范数也会出现类似的现象。

下图显示 CD-PSI(1) (L0)、 CD-PSI(1) (L0L1)和 CD-PSI(1) (L0L2)的行为相似;他们的表现明显优于不进行互换的表亲。很明显,这里介绍的局部组合优化方案在性能上具有优势。这一发现表明,在存在高度相关的特征的情况下,CD很容易陷入质量较差的解决方案中,因此组合搜索方案在引导其找到更好的解决方案方面起着重要作用。互换似乎有助于实现真正的基础解决方案,即使基础统计问题变得相对困难　n 值很小。

6.3.2 实验二:轻度相关

在这个实验中,我们保持与前一个实验完全相同的设置,但我们将相关参数 ρ 降低到 0.5。在图 3 中,我们展示了(L0)的算法 1、 (L0L2)的 CD-PSI(1)和其他竞争方法的结果。我们注意到,我们其余方法的结果与算法 1 (L0)和 CD-PSI(1) (L0L2) 具有相同的配置文件,但是,我们不包括空间限制图。直观地说,与 ρ = 0.9 的实验 1 相比,从统计角度来看,这种设置相对容易。因此,我们希望所有方法（总体而言)表现更好,并在较小的样本量下显示相变（与实验 1 相比）。实际上,如图 3 所示,算法 1 (L0)和 CD-PSI(1) (L0L2)具有大致相同的配置文件,并且它们优于其他方法;他们仅使用 200 个样本就完全恢复了真正的支持。在这种情况下,我们方法的交换变体似乎不会导致对非交换变体的显着改进,这可以通过我们的假设来解释:当统计问题很容易时,算法 1 成功地恢复了真实的潜在信号– 不需要交换。 MCP 和 FStepwise 也表现出良好的性能,但它们的转换发生在更大的 n 值上,并且 MCP 似乎并不稳健。 Lasso 在这种情况下永远不会

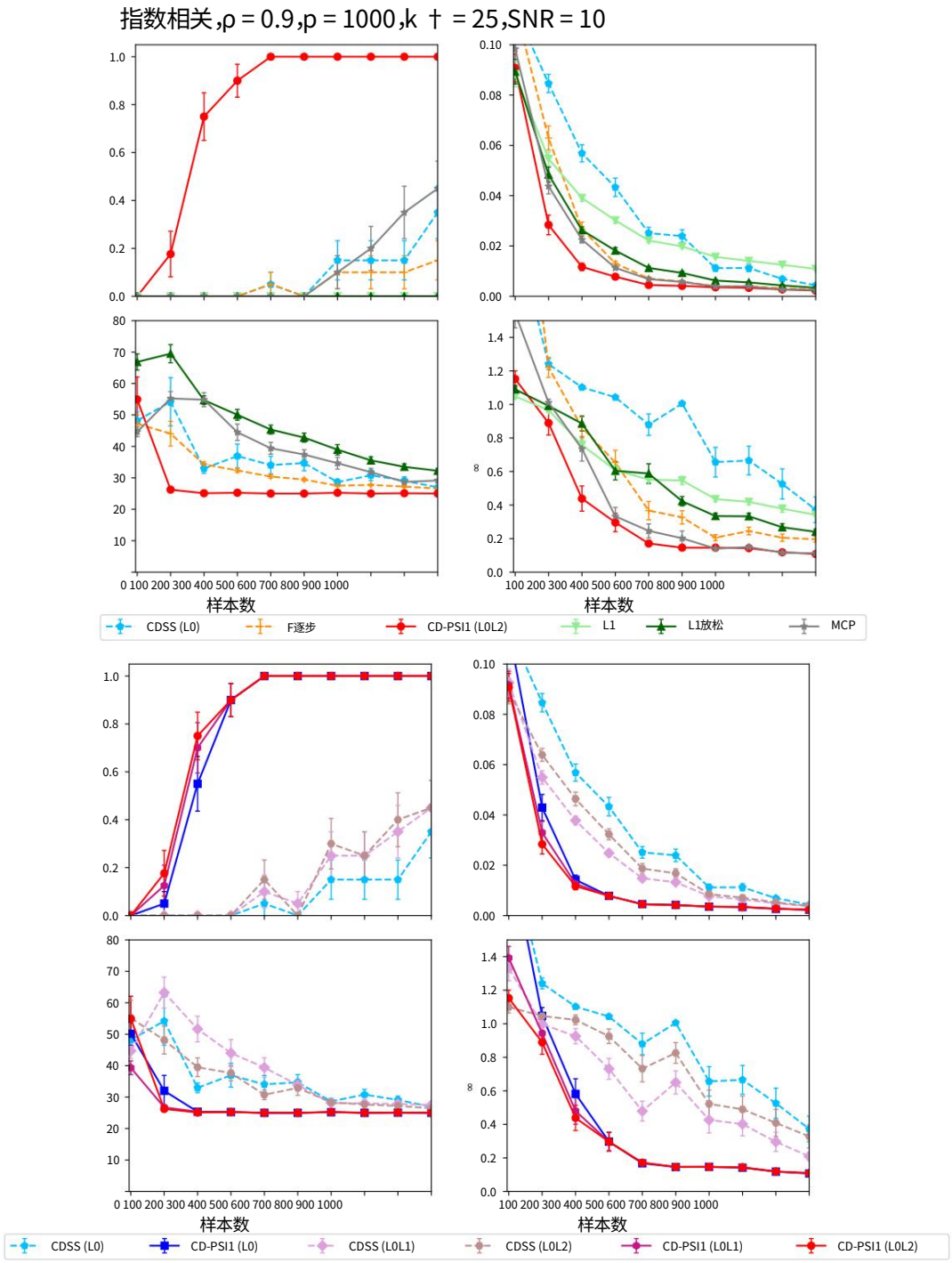

图 2:样本数量 n 在 100 和 1000 之间变化时的性能测量。上图比较了我们的两种方法 (a) CDSS(L0)（即,针对(L0)问题的算法 1），(b) CD-用于(L0L2)问题的 PSI(1)和其他最先进的算法。下图比较了所有三个问题的算法 1 和 CD-PSI(1)。

恢复真正的支持,并且此属性由松弛套索继承,需要至少 900 个样本才能完全恢复支持。

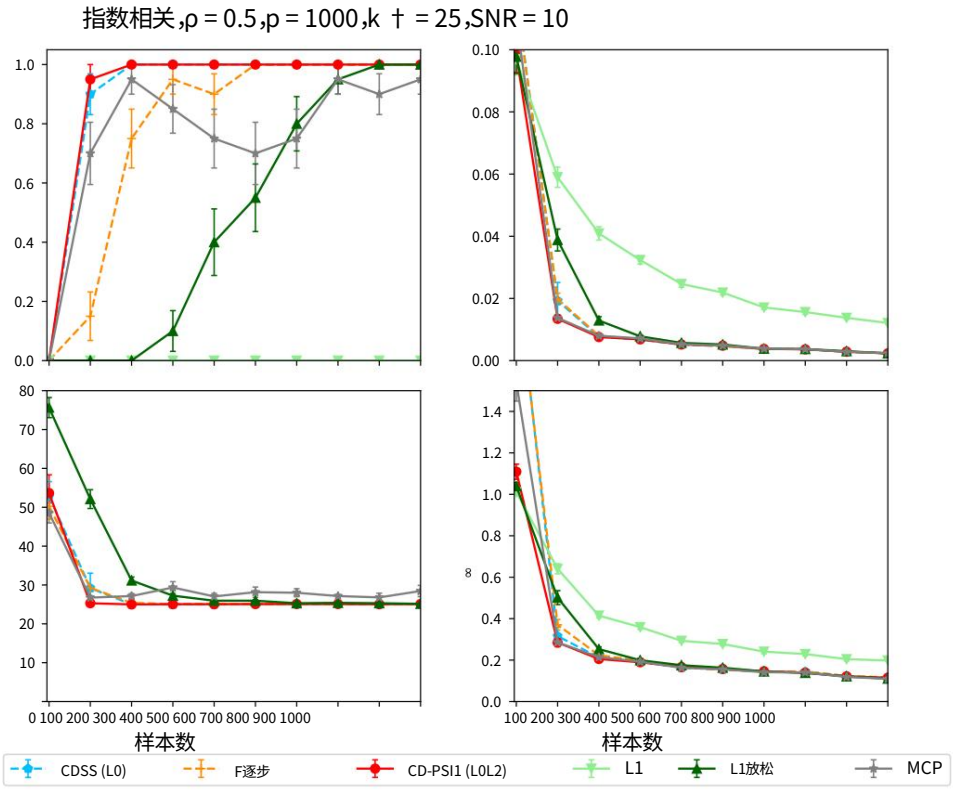

图 3:样本数 n 在 100 到 1000 之间变化时的性能测量。该图比较了我们的两种方法 (a) CDSS(L0)（即(L0)问题的算法 1），(b) CD-PSI (1) 对于(L0L2)问题,以及其他最先进的算法。

6.4 不同 SNR 的统计性能

我们进行了两个实验来研究改变 SNR 对不同性能测量的影响。在每个实验中,我们将 SNR 扫描在 0.1 和 100 之间,以获得对数刻度上等距的 10 个值。对于每个 SNR 值,我们生成了 20 个随机数据集,我们在这些数据集上平均了结果。我们观察到,对于低 SNR 值,岭回归在预测性能方面似乎工作得非常好。因此,我们在结果中包含岭回归 (L2)。

6.4.1 实验一:常数相关

我们生成了具有恒定相关性的数据集,ρ = 0.4,n = 1000,p = 2000 和 k = 50。
我们在图 4 中报告了算法 1 (L0)、 CD-PSI(1) (L0L2)和所有其他最先进算法的结果。

图 5 表明:CD-PSI(1) (L0L2)在整个 SNR 范围内的完全恢复、预测误差和L∞范数方面明显优于所有其他方法。对于低 SNR,其预测误差接近 L2 的预测误差,这在低 SNR 设置中非常有效。在 SNR = 100 时,CD-PSI(1) (L0L2)完全恢复支持,而算法 1 (L0)具有恢复概率

0.4。然而,即使对于高信噪比,其他考虑的方法都不能完全恢复。经过

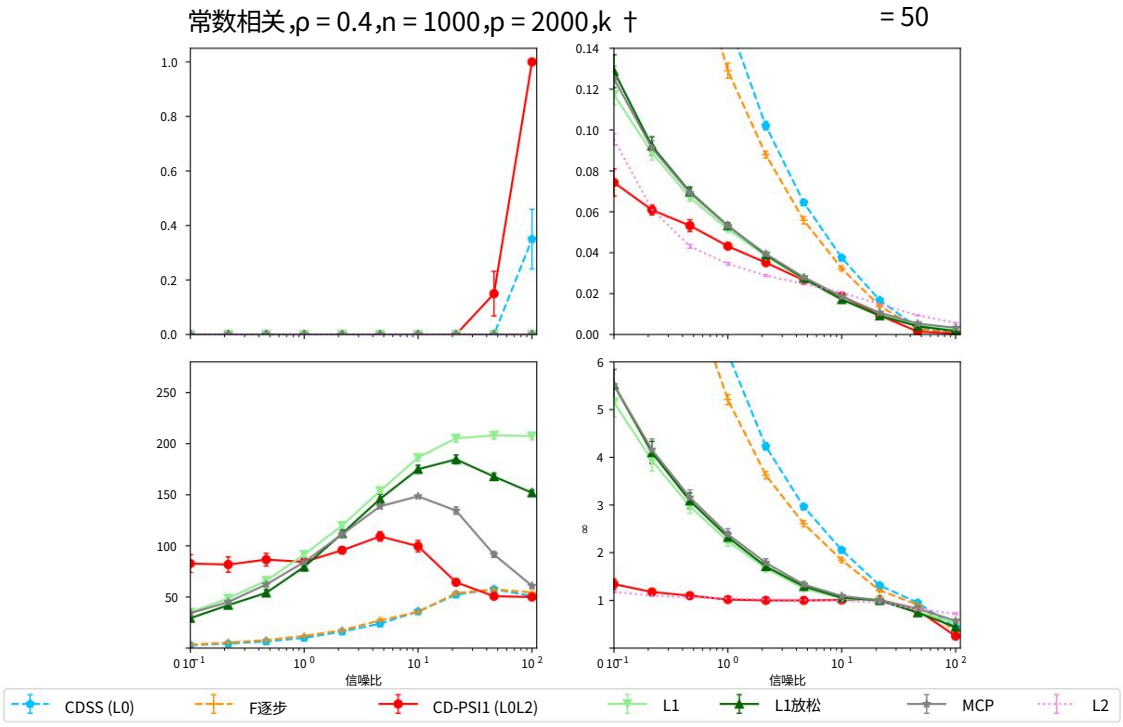

图 4:当信噪比 (SNR) 在 0.1 和 100 之间变化时的性能测量。该图比较了我们的两种方法 (a) CDSS(L0)（即针对(L0)问题的算法 1）， (b) 针对(L0L2)问题的 CD-PSI(1) 和其他最先进的算法。

检查L∞范数,我们可以看到 Lasso、Relaxed Lasso 和 MCP 表现出非常相似的行为,尽管它们针对不同的目标进行了优化。

6.4.2 实验二:指数相关

我们生成的数据集与 ρ = 0.5、n = 1000、p = 5000 和 k † = 50 具有指数相关性。我们报告了算法 1 (L0)、 CD-PSI(1) (L0L2)和所有其他算法的结果图 5 中的竞争算法。

看起来这个实验中的优化任务比第 6.4.1 节中的实验相对容易,因为变量之间的相关性呈指数衰减。因此,与第一个实验相比,我们观察到算法之间的差异较小 见图 5。CD-PSI(1) (L0L2)在所有测量和所有 SNR 值方面再次占主导地位。算法 1 (L0)的性能也优于 FStepwise,尤其是在选择概率方面。我们注意到,即使在这种相对简单的情况下,Lasso 也从未完全恢复支持。 MCP 似乎在完全康复方面也受到影响。

6.5 PSI(k)/FSI(k)之间的比较

在这里,我们检查了论文中介绍的各种最小值之间的差异:CW、PSI(k) 和 FSI(k) 最小值。为了理解这些差异,我们考虑了一个相对困难的常数相关设置,其中 ρ = 0.9,n = 250,p = 1000,支持大小为 k †

= 25. 我们

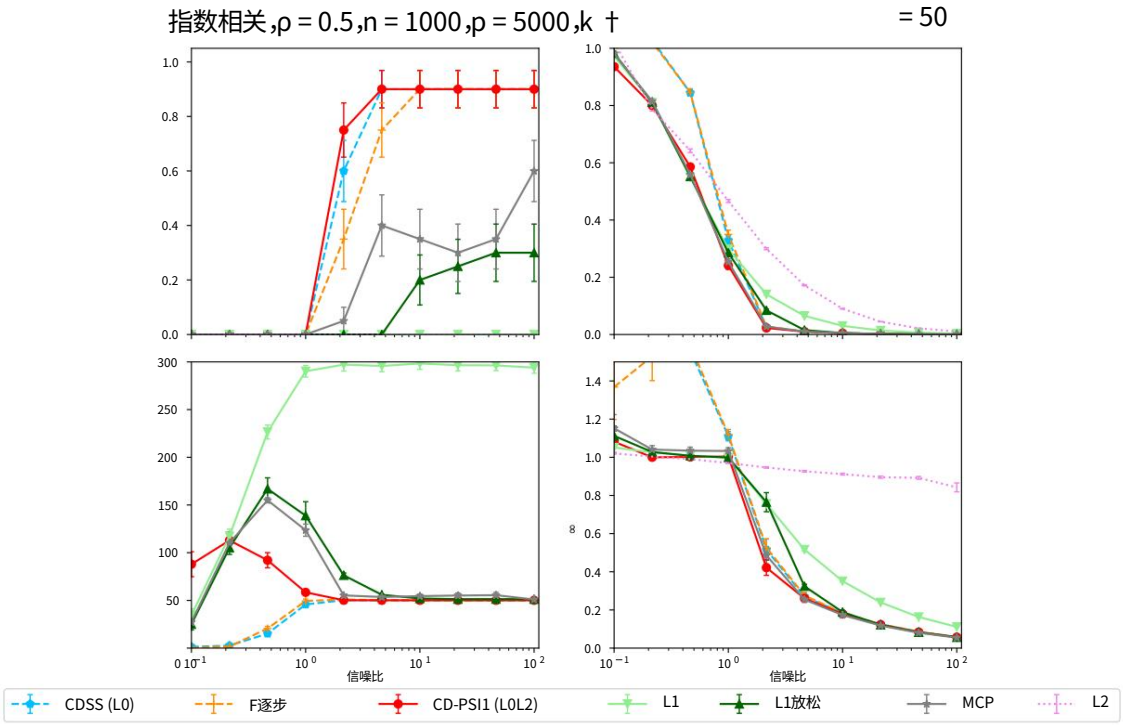

图 5:信噪比 (SNR) 在 0.1 和 100 之间变化时的性能测量值。该图比较了我们的两种方法 (a) CDSS(L0)（即针对(L0)问题的算法 1）， (b) 针对(L0L2)问题的 CD-PSI(1) 和其他最先进的算法。

设置 SNR= 300 以允许完全支持恢复。我们在此设置下生成了 10 个随机训练数据集,并使用(L0)惩罚为 k ∈ {1, 2, 5} 运行算法 1 和算法 2 的 PSI 和 FSI 变体。所有算法都用零向量初始化。对于算法 2,当 k > 1 时,我们使用 Gurobi (v7.5) 来解决 MIQO 子问题 (20) 和 (30)。

图 6 显示了箱形图,显示了每个算法和 10 个数据集记录的目标值、真阳性和假阳性的分布。与算法 1（导致 CW 最小值)相比,PSI(1) 和 FSI(1) 最小值导致目标显着降低。随着 k 的增加,我们确实观察到了进一步的减少,但收益不那么明显。在这种情况下,CW 最小值平均包含大量误报（> 35)和很少的真阳性　这可能是由于所有特征之间的高度相关性,这使得优化任务可以说非常具有挑战性。 PSI 和 FSI 最小值都显着增加了真阳性的数量。对误报数量的仔细检查表明,与 PSI 最小值相比,FSI 最小值在误报方面做得更好。这当然是以解决相对更困难的优化问题为代价的。

在图 7 中,我们展示了针对同一数据集运行算法 2 的 FSI(5) 变体时解决方案的演变。该算法从 CW 最小值开始,并在运行 CD-PSI(1) 和找到通过使用 MIO 解决优化问题 (30) 来改进目标的交换之间迭代。从运行 CD-PSI(1) 获得的 PSI(1) 最小值用红色圆圈标记,通过使用 MIO 解决问题 (30) 获得的结果用蓝色方块表示。

图 7 显示,在大多数情况下,在 MIO 获得的解决方案之上运行 CD-PSI(1) 会在更好的目标值方面取得重要收益。这也印证了我们的直觉

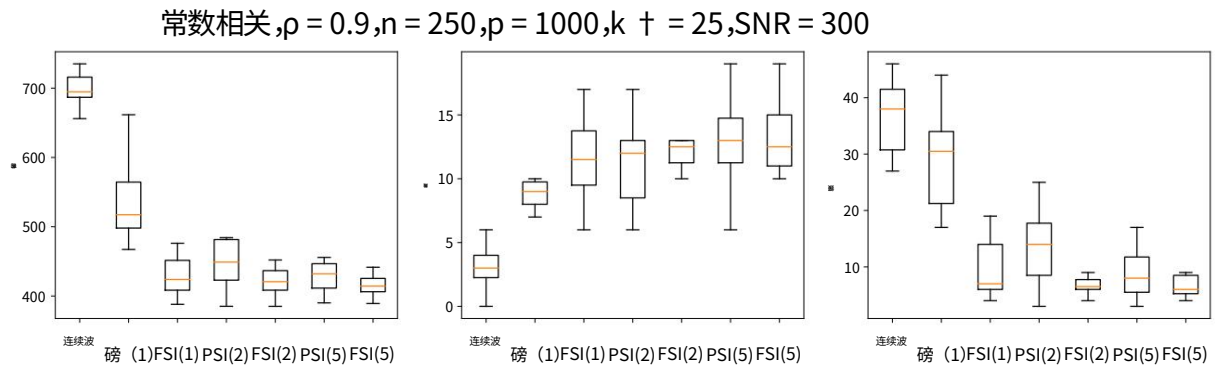

图 6:显示目标值分布、真阳性数和阳性数的箱形图
不同类别的局部最小值的误报。

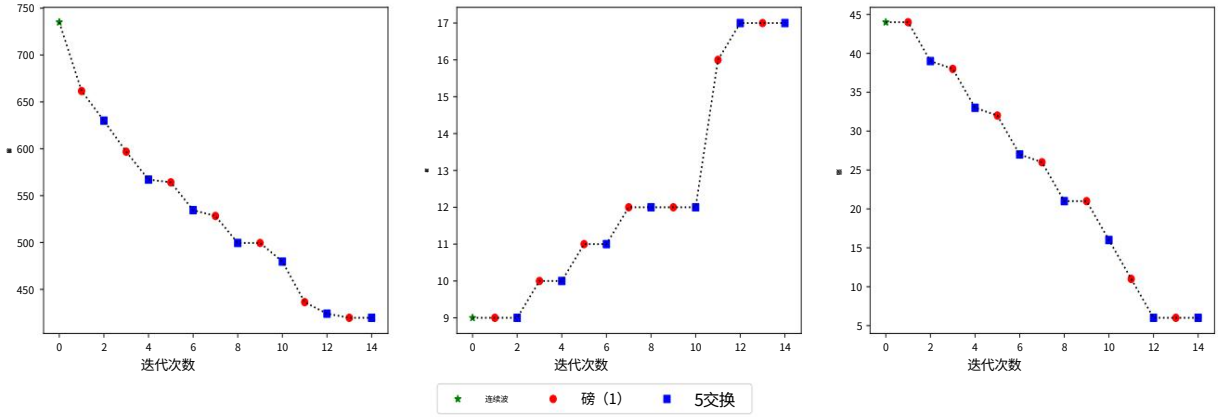

图 7:我们连续运行的算法 2 变体过程中解决方案的演变
CD-PSI(1) 并解决组合问题 (30) 以生成 FSI(5) 最小值。

MIO 可能会导致无法通过 PSI(1) 获得的解决方案。从真阳性图和
假阳性,我们可以看到 CD-PSI(1) 通过增加真阳性来改进解决方案
而 MIO 通过消除误报来改进解决方案。这一观察证实了
我们在图 6 中注意到的行为,其中 PSI(1) 最小值成功获得了一个好的数字
真阳性,但在假阳性方面受到影响。

6.6 大型高维实验

综合实验:在这里,我们研究了不同算法在以下情况下的性能
pn:我们在以下设置下进行了两个具有大量特征的实验：

- 设置 1:指数相关,ρ = 0.5,n = 1000,p = 105/2,k = 100,SNR = 10
- 设置 2:常数相关,ρ = 0.3,n = 1000,p = 105 ，k = 50,信噪比 = 100

每个实验重复 10 次,结果取平均值。我们报告了
表 (1) 中设置 1 和 2 的结果。在表1中,算法1与（L0L1)和（L0L2）
惩罚完全恢复真实支持并获得最低的预测误差。没有任何一个
其他方法能够完全支持恢复； Lasso 和 Relaxed Lasso 捕获了大部分

	设置 1				设置 2			
方法	kβk0	TP	FP	聚乙烯×102	kβk0	TP	FP PE×103	
阿尔格 2 (L0)	160 ± 24	79 ± 9	81 ± 33	5±1.6	69 ± 18	47 ± 3	22 ± 22	1.6±1
算法 1 (L0L2)	100 ± 0	100 ± 0	0 ± 0	0.97 ± 0.05	50 ± 0	50 ± 0	0 ± 0	0.5 ± 0.02
算法 1 (L0L1)	100 ± 0	100 ± 0	0 ± 0	1±0.05	50 ± 0	50 ± 0	0 ± 0	0.5 ± 0.02
L1	808 ± 7	95 ± 1	712±7	7.9±0.17	478 ± 11	50 ± 0	428 ± 11	4.7 ± 0.1
L1放松	602 ± 40	95 ± 1	508 ± 41	7.9 ± 0.19	385 ± 12	50 ± 0.2	335 ± 13	4.4 ± 0.2
MCP	102 ± 1	100 ± 0	2.3 ± 1	0.97 ± 0.05	65 ± 3	50 ± 0	15 ± 3	75 ± 2 50 ± 0 25 ± 2 3.5±0.13 1.1±0.07
F逐步	216 ± 17	64 ± 7	152 ± 23	8.9 ± 1.3				

表 1:设置 1 和 2 下不同算法的性能测量。TP、FP 和 PE 表示
分别为真阳性、假阳性和预测误差。均值的标准误是
在每个值旁边报告。

真正的阳性,但包括大量的假阳性。 MCP居中
在(L0L1)/(L0L2)和 Lasso 之间 它捕获了所有的真阳性并添加了更多的假阳性
积极的一面。我们还注意到,在如此高的 SNR 设置中,我们预计不会出现收缩（由
L1/L2处罚)以导致重大的统计改进。因此,性能差异
(L0)和(L0L1)/(L0L2)之间的关系可以通过以下事实来解释:连续正则化器
帮助算法 1 为问题 (3) 获得更好的解决方案。

时序和样本外性能:我们使用我们的工具包 L0Learn 运行算法 1 和
比较了运行时间和预测性能与 glmnet 和 ncvreg,在各种
真实和合成数据集。我们注意到 L0Learn、glmnet 和 ncvreg 正在解决不同的问题
优化问题 这里提供的运行时间是为了证明一个主要的
与高效的最新技术相比,我们提出的算法的主力具有竞争力
稀疏学习的实现。下面我们提供有关数据集的一些详细信息:

- 房价:p = 104, 000 和 n = 200。我们使用二阶多项式展开
  流行的波士顿房价数据集 [15] 获得 104 个特征。然后,我们添加了随机
  通过将 1000 个随机排列附加到数据矩阵中来“探测”（又名噪声特征）
  每一列。验证集和测试集分别有 100 个和 206 个样本。

- 亚马逊评论:p = 17、580 和 n = 2500。我们使用了 Amazon Grocery and Gourmet
  食品数据集 [19] 用于预测每条评论的有用性（基于其文本）。具体来说,
  我们将每条评论的有用性计算为赞成票数与
  否决票,我们通过使用 Scikit-learn 的 TF-IDF 转换器获得 X（同时移除
  停用词）。验证集和测试集分别有 500 个和 1868 个样本。我们也
  创建了这个数据集的增强版本,我们在其中添加了随机探针
  数据矩阵每列的 9 个随机排列得到 p = 174, 755。

- 美国人口普查:p = 55、537 和 n = 5000。我们使用了从 2016 年提取的 37 个特征
  美国人口普查计划数据库预测邮件退回率12 [9]。我们附加了数据
  每列有 1500 个随机排列的矩阵,我们随机抽样了 15,000 个
  行,均匀分布在训练集、测试集和验证集之间。

- Gaussian 1M:p = 106和 n = 200。我们生成了一个具有独立的合成数据集
  标准的正常条目。我们设置 k † = 20,SNR=10 并执行验证和测试为

12我们感谢美国人口普查局的 Emanuel Ben David 博士在准备此数据集方面提供的帮助。

在第 6.1 节中描述。

对于所有真实数据集,我们在单独的验证集和测试集上进行了调整和测试。时间安排
在具有 i7-4800MQ CPU 和 16GB RAM 运行 Ubuntu 16.04 和
OpenBLAS 0.2.20。对于所有方法,我们报告获得 100 个网格所需的训练时间
解决方案。对于(L0L2)、 (L0L1)和 MCP,我们分别提供固定λ2、 λ1和 γ的时间
（这些参数已设置为通过验证集调整超过 10 获得的最佳值
调整参数的值）。表 2 显示了所有四种方法的运行时间 每种方法
在 100 个调整参数的网格上计算解决方案。

表 2 中的结果显示如下： L0Learn 比 glmnet 和 ncvreg 更快
在所有考虑的数据集上,例如,在亚马逊评论数据集上的速度是其两倍以上。这
加速可以归因于 L0Learn 的精心设计（如第 5 节所述）,并且由于
对于L0正则化的性质,它通常选择比通过以下方法获得的支持更稀疏的支持
L1或 MCP 正则化。此外,对于(L0L2)和(L0L1)问题,L0Learn 提供
与其他工具包相比,更稀疏的支持和竞争性测试 MSE。最后,我们
请注意,我们的方法的预测误差可以通过使用算法 2 来改善,在
稍微增加计算时间的成本。

亚马逊评论

(p = 17, 580, n = 2500)

工具包	时间	MSE	×102 kβk0
glmnet (L1) 7.3	L0Learn	4.82	542
(L0L2) 3.3	L0Learn (L0L1) 2.8	4.77	159
ncvreg (MCP) 10.9		4.79	173
		6.71	1484

亚马逊评论（+探针）

(p = 174, 755, n = 2500)

工具包	时间	MSE	×102 kβk0
glmnet (L1) 49.4	L0Learn	5.11	256
(L0L2) 31.7	L0Learn (L0L1)	5.18	37
29.5	ncvreg (MCP) 67.3	5.20	36
		5.33	318

美国人口普查 (p

= 55, 537, n = 5000)

工具包	时间	MSE	kβk0
glmnet (L1) 28.7	61.3 L0Learn (L0L2)		222
19.6	60.7 15		
L0Learn (L0L1) 19.5	60.8 ncvreg (MCP)		11
	32.7	62.02	16

房价

(p = 104, 000, n = 200)

工具包	时间	MSE	kβk0
glmnet (L1) 2.3	L0Learn	100	112
(L0L2) 1.8	L0Learn (L0L1) 1.8	94	59
ncvreg (MCP)		104	74
	3.9	102	140

高斯1M

（p = 106 ， n = 200）

工具包	时间	MSE	kβk0
glmnet (L1)	22.5	4.55	185
L0学习（ L0L2 ）	16.5	4.64	11
L0Learn (L0L1)	16.7	5.12	15
ncvreg (MCP)	36.5	4.85	147

表 2:各种高维数据集的训练时间（以秒为单位）、样本外 MSE 和支持大小。训练时间用于获得具有 100 个解的正则化路径。

A 附录:证明和技术细节

A.1 引理 8 的证明

l

l

l−1

l l−1

l−1 l l−1 l l−1 l

k0 - kβ k0) ⩾ λ0。

⩾0 ⩾1

l

l−1

l l−1 l−1

l

l⁰

0 0 0 l

0−1 ⩾ N,这意味着 β

l⁰,1 ⁰,2 ⁰

,J都有支持 S（这来自这个引理的第 1 部分）。接下来,我们将 β 证明 β l,2也收敛到 β

0 0

卄°,1 卄°,2 l⁰,1 l⁰,2 l⁰,1 l⁰,2 2

小号 小号 2 2

l⁰,2

2

千

0

l⁰,1 l⁰,2 2 l⁰,1 l⁰,2 0 l⁰,1

2 2 2 2

l⁰,2 l⁰,2 l⁰,i和 β l⁰,i+1对于每个 → β

2 l⁰

l⁰

0 0

l⁰

βe我

l⁰

0

其中,βel 0 j6=i Xjβ | ⩾λ1 。

* l⁰ 1+2λ2

= lim l

0→∞

βe*

1+2λ2

}l 0∈L0收敛到 β 。通过第 1 部分,对于每个

lim
l 0→∞

}k 0∈K0,其中K0 ⊆ {0, 1, 2, . . . },使得非间隔步骤
}k 0∈K0通过矛盾法。
}k 0∈K0有一个不等于 β 的极限点 βb

林k
00→∞

气

| ⩾q 2λ0−2λ2 。根据这个引理的第 3 部分,{β } 收敛到平稳解 β

| ⩾q 2λ0, | ⩾q 2λ0。 □

A.2 引理 10 的证明

(2)

有一个子序列 $\{\beta k$ ⁰ ⁰ $) = S1$ 对于每个 =
.由于 $k \to \infty$,我们得到： F 有 F F(B(2))。我们
因此， F(B(1)) = F(B(2)),等价于

$$k0。(1)\ (1)\ (2)\ (2)\ f(B_{S1}) + \lambda 0 kB_{S1}\ k0 = f(B_{S2}) + \lambda 0 kB_{S2}$$

由于 $kB_{S2}\ k0 = kB_{S1}\ k0 + 1$ 我们可以简化上面得到：

$$f(B_{S1} \quad {}_{S2} \tag{39}$$

项 $f(B_{S2})$ 可以重写如下（使用基本代数操作）

$$_{S2} \quad \frac{1}{2} ky - XS2B_{S2} + \lambda 1 kB_{S2} \quad {}_{S2}$$

$$= \frac{1}{2} ky - XS1B_{S1} + \lambda 1 kB_{S1}\ (2)\ (2)\ k1 + \lambda 1 |B| + \lambda 2 kB_{S1} + \lambda 2(B$$

$$= \frac{1}{2}\ (2)\ ky - XS1B_{S1}\ k + \lambda 2 kB\ k_{S1}\ (2)\ (2) + \lambda 1 kB\ k1_{S1}$$

$$- hy - XS1B\ S1, Xj\ iB\ \frac{1}{-} + \lambda 2(B)\ (2)\ (2) + \lambda 1 |B|$$

$$_{S1}\ S1, Xj\ iB\ - \ (乙) + \lambda 2(B)\ (2)\ (2) + \lambda 1 |B|\ . \tag{40}$$

从引理 8 我们知道B(2)是一个固定解。使用（6)中固定解的表征并重新排列术语,我们得到：

$$S1, Xj\ i = (1 + 2\lambda 2)B + \lambda 1 sign(hy - XS1B\ S1, Xj\ i) \quad (2)\ |hy - XS1B\ S1, Xj\ i| > \lambda 1。 \tag{41}$$

将上面的第一个方程乘以 B 并使用事实 hy - XS1B 具有相同的符号（从上面的系统中可以看出）,我们得出和 j (2)

$$hy - XS1B\ S1, Xj\ iB\ (2)\ (2) = (1 + 2\lambda 2)(B)\ (2) + \lambda 1 |B|。 \tag{42}$$

将上述表达式代入 (40) 的 rhs 的第二项并使用kXjk
= 1 我们得到

$$(2)\ (2)\ (2)\ f(B_{S2}) = f(B_{S1}) \frac{1 + 2\lambda 2}{} (乙)\ 2\ . \tag{43}$$

将 (43) 代入等式 (39) 并重新排列各项,我们得出

$$|乙 \quad | = r\ 2\lambda 0 \overline{\frac{}{1 + 2\lambda 2} + \frac{2}{1 + 2\lambda 2}\ f(B_{S1}\ (2)\ (1)\ ) - f(B)_{S1}}。 \tag{44}$$

第 1 部分。)我们考虑第 1 部分,其中存在 $i \in S1$ 使得 hXi , Xj i 6= 0。根据引理 8 (1)我们有B(1)是一个固定解。因此,B ∈ $\arg\min_{S1} \beta S1\ f(\beta S1)$,以下成立

$$f(B)。_{S1}\ (1)\ (2)\ f(B)_{S1} \leqslant \tag{45}$$

乙。 2 (2) f(B) = (2)(2) f(B) − λS1B ... 满足 S(2) p(β i i i (2) 2 i * j(2)) 2

我们将证明上面的不等式是严格的。为此,假设 (45) 等式成立。

引理 9 暗示S1出现在迭代序列中。但是函数 f(βS1) 是强凸的（这对于(L0L2)来说是微不足道的,并且由于(L0)和(1)的假设 1 和引理 6 而成立

（L0L1)问题）。因此,B 是 f(βS1) 的唯一极小值。因此,必须是(1) (2) (1) (2)的情况 B = B 并且特别是 B = B 。通过S1 S1中静止解的表征， ii

(6) 我们有：

$$\text{符号}(hy - XS1\backslash\{i\}B^{(1)}_{S1\backslash\{i\}}, Xii)\ \frac{\overbrace{|hy - XS1\backslash\{i\}B^{(1)}_{S1\backslash\{i\}}, Xii| - \lambda 1}^{\geqslant 0}}{1 + 2\lambda 2\ (2)}$$

$$\lambda 1 = \text{sign}(hy - XS2\backslash\{i\}B^{(2)}_{S2\backslash\{i\}}, Xii)\ \frac{\underbrace{|hy - XS2\backslash\{i\}B_{S2\backslash\{i\}}, Xii| -}_{\geqslant 0}}{1 + 2\lambda 2} \quad (46)$$

观察到 (46) 中的两个符号项相等,我们可以将上述简化为：

$$hy - XS1\backslash\{i\}B^{(1)}_{S1\backslash\{i\}}, Xii = hy - XS2\backslash\{i\}B^{(2)}_{S2\backslash\{i\}},\text{十二}$$
$$\text{(2) (2)} = hy - XjB_j - XS1\backslash\{i\}B_{S1\backslash\{i\}},\text{十二} \quad (47)$$

其中,(47) 中的第二行紧随其后的是XS2\{i}B (2) {i} (2) (1)在 (47) - XjB S2\j (2) + XS1\{i}B S1\{i} (2) 。替代品

中,我们得到了不等式 i(=40),这与第1部分中的假设的这个结果相矛盾。因此,我们已经确

得出结论：$|B^{(2)}_j| > q\ 2\lambda 0\ \overline{1+2\lambda 2}$ 。

第 2 部分。)我们现在考虑所有 i ∈ S1时 hXi , Xj i = 0的情况。在这种情况下,优化(2)

问题minβS2 f(βS2) 分解为变量βS1和βj的优化。请注意,B (2) (1)) = f(B)。因此,从 (44) 我们 S1

和 $B^{(1)}_{S1}$ 都是minβS1 f(βS1) 的最小值;因此 f(B S1 S1

得到 $|B^{(2)}_j| = q\ 2\lambda 0\ \overline{1+2\lambda 2}$ 。最后,我们注意到对于任何满足Supp(β) = S2的 β ∈ R p ,我们有 T(βej , λ0, λ1, λ2) = B j 。这样就完成 (2)

了证明。 □

A.3 定理 2 的证明

证明。设 B 为 {β k} 的最大支撑点,用 S 表示其支撑。

我们将证明 β k → B 作为 k → ∞。

第 1 部分。)根据引理 9,有一个{β k} 的子序列 {β r}r∈R ,它满足： Supp(β r) = S 对于所有 r 和 β r → B（如 r → ∞）。根据引理 8,存在一个整数 N,使得对于每个 r ⩾ N,都是一个非间隔步骤,Supp(β r+1) = Supp(β r+1),我们通过矛盾来证明这一点。对于无限多个r,假设j ∈ Supp(β r+1) 。因此,存在{β r}r∈R的进一步子序列 {β r （其中R0 ⊆ R)使得 Supp(β r 0+1) = S \ {j}。对于每个 r ∈ R0 + 1 是一个非间隔步骤（因为 r ⩾ N）。 我们

0 }r 0∈R0

0 , 请注意,r 0 0

因此,应用引理 11 和 k = r 0 , 我们得到：

$$F(\beta r^0) - F(\beta r\ 0+1) \geqslant \frac{1 + 2\lambda 2}{2}\ |\beta_j r^0|\ \ r\ 2\lambda 0\ \ \overline{\overline{1 + 2\lambda 2}}\ !2\ 。\ |- \quad (48)$$

服用 r → ∞ 在 (48) 中并使用 {F(β)} （引理 5），我们得出结论

$$|\text{北京}| = \text{林} \quad |\beta\, r^{\circ}| = r\, 2\lambda 0 + 2\lambda 2. \tag{49}$$

，

°

我们有那个|Bj | > q 2λ0
对于每个 r ⩾ Nj ， 我们有 j ∈ Supp(β r+1)。

上述论证表明，在序列{βk} 中，没有任何支持 B 的 j 可以被无限频繁地丢弃。由于 S 具有最大的支持大小，因此在序列{βk}中，没有任何坐标可以无限频繁地添加到 S。第 1 部分的证明到此结束。

第 2 部分。）最后，我们证明 {β k} 的极限是 CW 最小值。为此，请注意第 1 部分（上图）和引理 8（第 3 部分和第 4 部分）的结果意味着 β 收敛到满足 Supp(B) = S 的极限 B，并且对于每个 i ∈ S，我们有：

$$\frac{|\text{贝}| - \lambda 1}{1 + 2\lambda 2} \quad \text{和} |\text{毕}| \geqslant r\, 2\lambda 0 \quad \frac{}{1 + 2\lambda 2}. \tag{50}$$

修正一些 j /∈ Supp(B) 并让 {β
j 已更新。对于每 k

$$\frac{|\beta ek\ 0| - \lambda 1}{} < r\, 2\lambda 0 + 2\lambda 2$$

其中 βek 0 = hy - P i , Xj i。取 k Xiβ i6=j → ∞ 在上面，我们有：

$$\frac{|\text{北京}| -}{\lambda 1\ 1 + 2\lambda 2} \leqslant r\, 2\lambda 0 + 2\lambda 2. \tag{51}$$

(50) 和 (51) 一起暗示 B 是 CW 最小值（根据定义）。 □

参考

[1] A. Beck 和 YC Eldar。稀疏约束非线性优化:优化条件和算法。 SIAM 优化杂志,23(3):1480–1509, 2013. doi: 10.1137/120869778。网址https://doi.org/10.1137/120869778。

[2] A. Beck 和 L. Tetruashvili。关于块坐标下降型方法的收敛性。 SIAM 优化杂志,23(4):2037–2060, 2013. doi: 10.1137/120887679。网址http://dx.doi.org/10.1137/120887679 。

[3] D. Bertsekas。非线性规划。雅典娜科学优化和计算系列。 Athena Scientific,2016 年。ISBN 9781886529052。网址https://books.google.com/books?id=TwOujgEACAAJ。

[4] D. Bertsimas 和 B. Van Parys。稀疏高维回归:精确的可扩展算法和相变。 arXiv 预印本 arXiv:1709.10029, 2017。

[5] D. Bertsimas、A. King 和 R. Mazumder。通过现代优化镜头选择最佳子集。志统计,44（2）:813-852,2016。

[6] T. Blumensath 和 M. 戴维斯。压缩感知的迭代硬阈值。应用与通信推定谐波分析,27（3）:265-274,2009。

[7] P. Breheny 和 J. Huang。非凸惩罚回归的坐标下降算法,应用于生物特征选择。应用统计年鉴,5（1）:232,2011。

[8] P. B¨uhlmann 和 S. van-de-Geer。高维数据的统计。斯普林格,2011。

[9] C. Erdman 和 N. Bates。美国人口普查局邮件退回率挑战:众包开发难以计数的分数。统计,第 8 页,2014 年。

[10] J. Friedman、T. Hastie 和 R. Tibshirani。通过坐标下降的广义线性模型的正则化路径。统计软件杂志,33(1):1-22,2010。网址http://www.jstatsoft。 org/v33/i01/.

[11] G. Furnival 和 R. Wilson。以突飞猛进的方式回归。技术计量学,16:499–511,1974。

[12] D. Gamarnik 和 I. Zadik。具有二进制系数的高维回归。估计平方误差和相变。在学习理论会议上,第 948-953 页,2017 年。

[13] E. Greenshtein 和 Y. Ritov。高维线性预测器选择的持久性和优点过度参数化。伯努利,10:971–988,2004。

[14] E.格林施泰因。最佳子集选择,高维统计学习的持久性和`1约束下的优化。统计年鉴,34(5):2367–2386, 2006。

[15] D.哈里森和 DL 鲁宾菲尔德。享乐房价和对清洁空气的需求。环境经济与管理杂志,5(1):81 – 102, 1978. ISSN 0095-0696。 doi: https://doi.org/10.1016/0095-0696(78)90006-2。网址http://www.sciencedirect.com/science/article/pii/0095069678900062 。

[16] T. Hastie、R. Tibshirani 和 J. Friedman。统计学习的要素:预测、推理和数据挖掘（第二版）。施普林格出版社,纽约,2009 年。

[17] T. Hastie、R. Tibshirani 和 RJ Tibshirani。最佳子集选择的扩展比较,前向逐步选择和套索。 ArXiv 电子版,2017 年 7 月。

[18] T. Hastie、R. Tibshirani 和 M. Wainwright。稀疏的统计学习:套索和根泛化。 CRC 出版社,佛罗里达州,2015 年。

[19] R. He 和 J. McAuley。起起落落:用一类协同过滤对时尚趋势的视觉演变进行建模。在第 25 届万维网国际会议论文集上,WWW 16,第 507-517 页,瑞士日内瓦共和国和州,2016 年。国际万维网会议指导委员会。国际标准书号 978-1-4503-4143-1。 doi: 10.1145/2872427.2883037。

网址https://doi.org/10.1145/2872427.2883037。

[20] P.-L.罗和 MJ 温赖特。支持没有不连贯的恢复:非凸正则的一个案例化。统计年鉴,45(6):2455–2482, 2017。

[21] 吕志。 l0 正则化凸锥规划的迭代硬阈值方法。数学编程,147(1):125–154,2014 年 10 月。ISSN 1436-4646。 doi: 10.1007/s10107-013-0714-4。网址https://doi.org/10.1007/s10107-013-0714-4。

[22] R. Mazumder 和 P. Radchenko。离散 Dantzig 选择器:通过混合整数线性优化估计稀疏线性模型。 IEEE Transactions on Information Theory, 63 (5):3053 – 3075, 2017。

[23] R. Mazumder、JH Friedman 和 T. Hastie。稀疏网络:使用非凸惩罚的坐标下降。
美国统计协会杂志,106(495):1125–1138, 2011。doi: 10.1198/jasa.2011。 tm09738。网址https://doi.org/10.1198/jasa.2011.tm09738。 PMID:25580042。

[24] R. Mazumder、P. Radchenko 和 A. Dedieu。带收缩的子集选择:稀疏线性建模
当信噪比低时。 arXiv 预印本 arXiv:1708.03288, 2017。

[25] A.米勒。回归中的子集选择。 CRC 华盛顿出版社,2002 年。

[26] B.纳塔拉詹。线性系统的稀疏近似解。 SIAM 计算杂志,24(2):
227-234,1995。

[27] Y.涅斯捷罗夫。坐标下降法在大规模优化问题上的效率。 SIAM 优化杂志,22(2):341–362,2012。

[28] Y.涅斯捷罗夫。凸优化入门讲座:基础课程。 Kluwer,诺威尔,2004 年。

[29] A. Patrascu 和 I. Necoara。 `0正则化凸优化的随机坐标下降方法。
IEEE Transactions on Automatic Control,60(7):1811–1824,2015 年 7 月。ISSN 0018-9286。 doi: 10.1109/TAC.2015.2390551。

[30] G. Raskutti、M. Wainwright 和 B. Yu。高维线性回归超球的极小极大估计率。信息论,IEEE Transactions on,57（10）:6976–6994,2011。

[31] C. 桑德森和 R. 科廷。 Armadillo:一个基于模板的线性代数 C++ 库。开源软件杂志,2016 年。

[32] W. Su、M. Bogdan 和 E. Candes。错误发现发生在套索路径的早期。年鉴
统计,45（5）:2133-2150,2017。

[33] R. Tibshirani、J. Bien、J. Friedman、T. Hastie、N. Simon、J. Taylor 和 RJ Tibshirani。在套索类型问题中丢弃预测变量的强规则。皇家统计学会杂志:B 系列（统计方法）,74（2）:245–266,2012。

[34] RJ蒂布希拉尼。套索问题和唯一性。电子统计杂志,7:1456–1490,2013。

[35] P.曾。用于不可微最小化的块坐标下降法的收敛。杂志
优化理论与应用,109:475–494,2001。

[36] C.-H.张。极小极大凹惩罚下的几乎无偏变量选择。统计年鉴
抽动症,38（2）:894–942,2010。

[37] C.-H.张和 T. 张。高维稀疏凹正则化的一般理论
估计问题。统计科学,27（4）:576–593,2012。

[38] Y. Zhang、MJ Wainwright 和 MI Jordan。稀疏线性回归的多项式时间算法的性能下限。在学习理论会议上,第 921-948 页,2014 年。