稀疏优化在数独中的应用

陈永鑫, 蔡邢菊*, 姜波

南京师范大学数学科学学院, 南京 210023

E-mail: 180902004@stu.njnu.edu.cn, caixingju@njnu.edu.cn, jiangbo@njnu.edu.cn

收稿日期: 2020-01-07; 接受日期: 2020-03-12; 网络出版日期: 2020-09-08; * 通信作者

国家自然科学基金 (批准号: 11571178, 11871279 和 11971239) 和第三届中国科协青年人才托举工程 (批准号: 2017QNRC001) 资助项目

摘要 数独是一个难以求解的整数规划问题, 可以通过实数编码的方式去除整数约束的限制, 将整数规划模型转化为一个 ℓ_0 范数极小化模型. 已有算法大多是求解松弛的 ℓ_1 范数极小化模型, 只能求解部分数独问题. 本文证明对于数独这样一个特殊的问题, ℓ_q ($0 < q < 1$) 范数极小化模型等价于 ℓ_0 范数极小化模型, 同时用 $\ell_{1/2}$ -SLP (sequential linear programming) 算法求解 $\ell_{1/2}$ 范数极小化模型. 数值实验表明该方法可以求解更多的数独问题, 本文从时间和成功率两方面验证了算法的高效性.

关键词 实数编码 稀疏优化 ℓ_0 范数极小化模型 ℓ_q ($0 < q < 1$) 范数极小化模型 $\ell_{1/2}$ -SLP 算法

MSC (2020) 主题分类 90C25, 90C26, 90C90

1 引言

“稀疏优化”中的“稀疏”指的是变量中绝大多数元素为 0. 近年来, 在信号与图像处理^[1]、压缩感知^[2,3]、统计^[4]和机器学习^[5]等领域涌现出了大量的优化问题, 其特点为优化变量具有某种稀疏结构, 利用稀疏结构设计算法能加快大规模优化问题的计算速度.

求解稀疏优化模型的算法大致分为 3 类: (1) 直接或近似求解 ℓ_0 极小化模型; (2) 将 ℓ_0 范数凸松弛为 ℓ_1 范数并求解; (3) 将 ℓ_0 范数松弛为非凸的 ℓ_q ($0 < q < 1$) 范数并求解. 直接求解 ℓ_0 范数极小化模型主要是贪婪算法, 包括各类匹配追踪 (matching pursuits, MP) 算法^[6]和各类正交匹配追踪 (orthogonal matching pursuit, OMP) 算法^[7,8]. 迭代硬阈值 (iterative hard thresholding, IHT) 算法^[9]也是求解 ℓ_0 范数极小化模型的有效算法. 近似求解 ℓ_0 范数极小化模型主要是用各种光滑的函数近似 ℓ_0 范数^[10,11]. Candès 等^[12-14]证明了在一定条件下 ℓ_0 极小化模型与 ℓ_1 极小化模型概率等价, 所以求解 ℓ_0 极小化可以凸松弛为求解 ℓ_1 极小化. 求解 ℓ_1 极小化有许多高效算法, 如投影梯度算法 (projected gradient method, PGM)^[15,16]、快速迭代阈值算法 (fast iterative shrinkage-thresholding algorithm, FISTA)^[17]、交替方向乘子方法 (alternating direction method of multipliers, ADMM)^[18]和

英文引用格式: Chen Y X, Cai X J, Jiang B. Application of sparse optimization in Sudoku (in Chinese). Sci Sin Math, 2022, 52: 209–222, doi: 10.1360/SSM-2020-0006

Bregman 算法^[19]等. Gorodnitsky 和 Rao^[20]提出用 ℓ_q ($0 < q < 1$) 范数来松弛 ℓ_0 范数, 与 ℓ_1 范数松弛相比, 虽然 ℓ_q ($0 < q < 1$) 范数可以得到更加稀疏的解, 但是 ℓ_q ($0 < q < 1$) 范数的非凸性增大了求解该问题的难度. 目前求解 ℓ_q ($0 < q < 1$) 范数极小化模型的方法主要有仿射缩放方法 (affine scaling methodology, ASM)^[21,22] 和迭代调整权值最小二乘 (iteratively reweighted least squares, IRLS)^[23], 还有求解 $\ell_{1/2}$ 范数的半迭代阈值算法^[24].

数独是一种数字填充游戏, 起源于 18 世纪大数学家 Euler 研究的拉丁矩阵. 数独有 9×9 的方格, 玩家需要在这 81 个方格里填上数字, 游戏开始时有一些格子已经给出提示的数. 玩家需要填写 1 到 9 的数字到格子中, 保证每一行每一列都没有重复的数字, 同时 9 个 3×3 的九宫格也要保证没有数字重复, 1 到 9 的数字各一个, 数独游戏保证解的存在性和唯一性. 文献 [25] 提出了一种编码方式将数独问题用 ℓ_0 极小化模型刻画, 并松弛为 ℓ_1 极小化模型. 文献 [26] 用 ℓ_1 极小化模型求解提示数为 24 (81 个方格中有 24 个方格中的数字已经给出) 以上的数独问题达到了 100% 的成功率, 但数独问题不符合 ℓ_1 模型和 ℓ_0 模型等价的条件, 在求解提示数为 17 的数独问题 (最难数独) 时, 文献 [26] 中的算法达到了 86.4% 的成功率 (求解数量为 1,000 个). 本文用 $\ell_{1/2}$ 模型代替 ℓ_0 模型, 同时用 $\ell_{1/2}$ -SLP 方法求解 $\ell_{1/2}$ 模型, 在求解 45,567 个数独 (提示数为 17) 时, 达到了 95.69% 的成功率, 更快更好地求解了数独问题.

2 背景及模型建立

我们将数独的解看成是一个 9×9 矩阵 X , M_{ij} 表示 X 的已经确定的数即提示数的值, 每一行、每一列和每一个九宫格都是由 1 到 9 组成且每个数字出现一次.

该问题是一个整数规划问题, 文献 [25] 给出一种编码策略, 可以松弛该问题中的整数约束限制.

我们先将矩阵 X 转化为向量 $\text{vec}(X)$, 其中 $\text{vec}(X)$ 表示将 X 的元素按列拉直, 即矩阵 X 为 9×9 矩阵, 则 $\text{vec}(X) = [X_{11}, \dots, X_{91}, X_{12}, \dots, X_{92}, \dots, X_{19}, \dots, X_{99}]^T$, 再将 $\text{vec}(X)$ 中的每个元素 X_{ij} 按照以下方式实数编码转化, 则得到的一个 729×1 的列向量, 记为 x .

我们将 1 到 9 的每一个整数都用一个 9 维列向量来表示, 向量的每个元素都是 0 到 1 的实数, $\text{vec}(X)$ 的每一个元素 X_{ij} 都可以表示为 $[X_{ij1} \ X_{ij2} \ X_{ij3} \ X_{ij4} \ X_{ij5} \ X_{ij6} \ X_{ij7} \ X_{ij8} \ X_{ij9}]^T$, 若 $X_{ij} = k$, 则 X_{ij} 所对应的 9 位编码中 $X_{ijk} = 1$, 其余分量均为 0. 数字 1 到 9 的编码见表 1. 由此可得到 $x = [X_{111}, \dots, X_{119}, X_{211}, \dots, X_{219}, \dots, X_{ij1}, \dots, X_{ij9}, \dots, X_{991}, \dots, X_{999}]^T$. 设 I_n 表示 n 阶的单位矩阵,

表 1 数字 1 到 9 的编码

提示数 X_{ij} 的值	9 位编码 $[X_{ij1} \ X_{ij2} \ X_{ij3} \ X_{ij4} \ X_{ij5} \ X_{ij6} \ X_{ij7} \ X_{ij8} \ X_{ij9}]$
1	[1 0 0 0 0 0 0 0 0]
2	[0 1 0 0 0 0 0 0 0]
3	[0 0 1 0 0 0 0 0 0]
4	[0 0 0 1 0 0 0 0 0]
5	[0 0 0 0 1 0 0 0 0]
6	[0 0 0 0 0 1 0 0 0]
7	[0 0 0 0 0 0 1 0 0]
8	[0 0 0 0 0 0 0 1 0]
9	[0 0 0 0 0 0 0 0 1]

$e_i \in \mathbb{R}^{1 \times 729}$, $v_i \in \mathbb{R}^{1 \times 81}$ 是第 i 个位置上元素是 1、其余位置上元素都是 0 的向量, $\otimes$ 表示 Kronecker 积.

定义 2.1 (Kronecker 积) 如果 $A = (a_{ij})$ 是一个 $m \times n$ 的矩阵, 而 B 是一个 $p \times q$ 的矩阵, 则

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

数独每一列都是由不重复的 1 到 9 组成, 即每个数字对应的九位编码中的 1 所在位置都是不一样的, 所以对应位置的代数和为 1, 我们以第 1 列为例, 要求 x 满足

$$\sum_{i=1}^9 X_{i11} = 1, \quad \sum_{i=1}^9 X_{i12} = 1, \quad \dots, \quad \sum_{i=1}^9 X_{i19} = 1,$$

也可以等价地写为

$$\left(\sum_{i=1}^9 v_i \right) \otimes I_9 x = 1_{9 \times 1}.$$

同理, 第 k ($k = 1, 2, \dots, 9$) 列为

$$\left(\sum_{i=1}^9 v_{9(k-1)+i} \right) \otimes I_9 x = 1_{9 \times 1}. \quad (2.1)$$

每一行由不重复的 1 到 9 组成, 第 k ($k = 1, 2, \dots, 9$) 行要求 x 满足

$$\left(\sum_{i=1}^9 v_{9i-9+k} \right) \otimes I_9 x = 1_{9 \times 1}. \quad (2.2)$$

每一个九宫格由不重复的 1 到 9 组成, 第 k ($k = 1, 2, 3$) 个九宫格要求 x 满足

$$\left(\sum_{i=1}^3 \sum_{j=1}^3 v_{27(k-1)+9i-9+j} \right) \otimes I_9 x = 1_{9 \times 1}, \quad (2.3)$$

第 k ($k = 4, 5, 6$) 个九宫格要求 x 满足

$$\left(\sum_{i=1}^3 \sum_{j=1}^3 v_{27(k-4)+9i-6+j} \right) \otimes I_9 x = 1_{9 \times 1}, \quad (2.4)$$

第 k ($k = 7, 8, 9$) 个九宫格要求 x 满足

$$\left(\sum_{i=1}^3 \sum_{j=1}^3 v_{27(k-7)+9i-3+j} \right) \otimes I_9 x = 1_{9 \times 1}. \quad (2.5)$$

对提示数的约束, 以第一个格子为 1 举例, 要求 x 满足

$$e_1 x = 1.$$

同理, 若 $M_{ij} = k$, 即第 i 行 j 列格子数字为 k , 则 x 的第 $(j-1) \times 81 + (i-1) \times 9 + k$ 个分量为 1, 表示为

$$e_{81(j-1)+9(i-1)+k}x = 1. \quad (2.6)$$

对填充数的约束, 即每一个格子只能填 1 到 9 的数字, 以第一个格子为例, 要求 x 满足

$$\left(\sum_{i=1}^9 e_i \right) x = 1.$$

同理, 第 k ($k = 1, 2, \dots, 81$) 个格子要求 x 满足

$$\left(\sum_{i=1}^9 e_{9k-9+i} \right) x = 1. \quad (2.7)$$

通过实数编码, 数独问题 (17 提示数) 转变成为了满足 (2.1)–(2.7) 组成的线性方程组, 记线性方程组的系数矩阵为 $A \in \mathbb{R}^{729 \times 341}$, $b \in \mathbb{R}^{341}$ 是全 1 的列向量, 则数独问题可以表述为以下优化模型:

$$\begin{aligned} & \text{找 } x \\ & \text{s.t. } Ax = b, \\ & \quad x_i \in \{0, 1\}. \end{aligned} \quad (2.8)$$

利用 ℓ_0 范数对上述模型进行转化:

$$\begin{aligned} & \min \|x\|_0 \\ & \text{s.t. } Ax = b, \end{aligned} \quad (2.9)$$

其中 $\|x\|_0$ 是 x 中非零元的个数, 为了叙述方便, 我们在本文中记作 ℓ_0 范数 (ℓ_0 范数不是真正意义上的范数). 使 $\|x\|_0$ 最小的最优解 x^* 的元素必然只有 0 和 1, 从而保证模型 (2.8) 和 (2.9) 的等价性.

3 稀疏优化算法

3.1 凸松弛为基追踪问题

求解 ℓ_0 极小化模型, 一般将其凸松弛为 ℓ_1 范数极小化模型

$$\begin{aligned} & \min \|x\|_1 \\ & \text{s.t. } Ax = b. \end{aligned} \quad (3.1)$$

模型 (3.1) 也称为基追踪 (basis pursuit, BP) 问题.

下面给出一些 ℓ_0 极小化模型和 ℓ_1 极小化模型等价的充分性保证.

定义 3.1 (k -稀疏) 对于一个长度为 N 的向量 x , 它的 N 个元素中只有 k 个是非零元且 $k \ll N$, 此时称向量 x 是 k -稀疏的.

定义 3.2 (零空间) 已知一个矩阵 $A \in \mathbb{R}^{m \times n}$, A 的零空间表示为

$$\text{Null}(A) = \{y \mid Ay = 0\}.$$

设指标集 S 为 $\{1, 2, \dots, n\}$ 的子集, $\bar{S}$ 是指标为 $\{1, 2, \dots, n\} \setminus S$ 的集合.

定义 3.3 ^[27] (零空间性质) 已知一个矩阵 $A \in \mathbb{R}^{m \times n}$, A 的零空间非零元素如果满足

$$\|h_{\bar{S}}\|_1 < \|h_S\|_1, \quad \forall h \in \text{Null}(A) \setminus \{0\},$$

对所有满足 $|S| \leq k$ 的集合 S 都成立, 我们就称这样的矩阵 A 具有 k 阶零空间性质, 其中 $|S|$ 表示的是集合 S 的大小.

定义 3.4 ^[13] (限定等距性 (restricted isometry property, RIP)) 对于矩阵 $A \in \mathbb{R}^{m \times n}$, 若存在常数 $\delta_k \in (0, 1)$, 对于任意 k -稀疏的向量 x , 满足

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2,$$

则称矩阵 A 满足 k -限定等距性, 其中的常数 δ_k 被称为 RIP 常数.

定理 3.1 ^[13] (1) ℓ_1 极小化模型 (3.1) 存在唯一 k -稀疏解 x^* 当且仅当矩阵 A 具有 k 阶零空间性质.

(2) 如果模型中的矩阵 A 满足 RIP 条件, 则当 $\delta_{2k} < 1$ 时, ℓ_0 极小化模型有唯一的 k -稀疏解; 当 $\delta_{2k} < \sqrt{2} - 1$ 时, ℓ_1 极小化模型 (3.1) 的解也是 ℓ_0 极小化模型的解.

注 3.1 对于数独这样一个特殊的问题, ℓ_0 极小化模型和 ℓ_1 极小化模型不等价.

下面介绍几种常见的求解 ℓ_1 极小化模型的方法.

转化为线性规划的方法 ^[3] 是一种常见的求解 BP 问题的算法, 本文简记为 BP 算法. 引入两个非负向量 z 和 w , 其中向量 z 和 x 分别代表向量 x 的正负部分, 此时模型可以转化为

$$\begin{aligned} \min \quad & 1^T(z + w) \\ \text{s.t.} \quad & A(z - w) = b, \quad z \geq 0, \quad w \geq 0. \end{aligned} \quad (3.2)$$

这是一个线性规划问题, 假设 $(z^{*T}, w^{*T})^T$ 是该线性规划问题的解, 则 $x^* = z^* - w^*$ 就是原问题所要求的解. 我们作进一步的修改, 令 $y = [z^T, w^T]^T$, 则目标函数 $1^T(z + w) = 1^T y$, 等式约束为

$$A(z - w) = [A, -A][z^T, w^T]^T = [A, -A]y,$$

不等式约束为 $z \geq 0, w \geq 0$ 变为 $y \geq 0$, 这样就得到了标准的线性规划模型:

$$\begin{aligned} \min \quad & 1^T y \\ \text{s.t.} \quad & [A, -A]y = b, \quad y \geq 0. \end{aligned} \quad (3.3)$$

交替方向乘子法 (ADMM) ^[18, 28] 是一个求解可分凸优化问题的有效算法, 将模型 (3.1) 变为如下形式:

$$\begin{aligned} \min \quad & I_\Omega(x) + \|z\|_1 \\ \text{s.t.} \quad & x - z = 0, \end{aligned} \quad (3.4)$$

其中 $I_\Omega(x)$ 是 $\Omega = \{x \in \mathbb{R}^n \mid Ax = b\}$ 的指标函数. 其增广 Lagrange 函数为

$$\mathcal{L}_\rho(x, z, y) = I_\Omega(x) + \|z\|_1 + y^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2,$$

其中 y 为 Lagrange 乘子, ρ 为惩罚参数, ADMM 的迭代格式为

$$\begin{cases} x^{k+1} := \operatorname{argmin} \mathcal{L}_\rho(x, y^k, z^k), \\ z^{k+1} := \operatorname{argmin} \mathcal{L}_\rho(x^{k+1}, y^k, z), \\ y^{k+1} := y^k + \rho(x^{k+1} - z^{k+1}). \end{cases} \quad (3.5)$$

定制的临近点算法 (proximal point algorithm, PPA) 也是求解 ℓ_1 极小化问题的行之有效的算法. 模型 (3.1) 的 Lagrange 函数为

$$\mathcal{L}(x, \lambda) = \|x\|_1 - \lambda^T(Ax - b),$$

其中 λ 是 Lagrange 乘子, 对给定的 $r, s > 0$ 和 (x^k, λ^k) , 由定制 PPA 算法产生 $(\tilde{x}^k, \tilde{\lambda}^k)$:

$$\begin{aligned} \tilde{x}^k &= \operatorname{argmin} \left\{ \mathcal{L}(x, \lambda^k) + \frac{r}{2}\|x - x^k\|^2 \right\}, \\ \tilde{\lambda}^k &= \operatorname{argmax} \left\{ \mathcal{L}(2\tilde{x}^k - x^k, \lambda) - \frac{s}{2}\|\lambda - \lambda^k\|^2 \right\}, \end{aligned}$$

取 $x^{k+1} = x^k - \alpha(x^k - \tilde{x}^k)$, $\alpha \in (0, 2)$. 当 $\alpha \in (1, 2)$ 时, 此时的迭代格式被称为外延的 PPA 算法, α 称为松弛因子, 取 $\alpha \in [1.2, 2)$ 一般会加快收敛速度^[29].

3.2 光滑连续函数近似 ℓ_0 范数 (SL0)

最小化模型 (2.9) 中 ℓ_0 范数能够充分体现出信号稀疏程度的性质. 但却因为 ℓ_0 范数本身不连续不可微, 所以可以用连续函数近似 ℓ_0 范数. 这样不仅能保证信号稀疏程度, 又让函数本身的性质较易于去求解.

考虑 Gauss 函数

$$f_\sigma(t) = e^{-t^2/2\sigma^2},$$

显然,

$$\lim_{\sigma \rightarrow 0} f_\sigma(t) = \begin{cases} 1, & t = 0, \\ 0, & t \neq 0. \end{cases}$$

对 $x \in \mathbb{R}^n$, 定义如下函数:

$$F_\sigma(x) = n - \sum_{i=1}^n f_\sigma(x_i) = n - \sum_{i=1}^n e^{-x_i^2/2\sigma^2},$$

此时当 $\sigma \rightarrow 0$ 时, 有

$$\lim_{\sigma \rightarrow 0} F_\sigma(x) = n - \lim_{\sigma \rightarrow 0} \sum_{i=1}^n f_\sigma(x_i) = n - \lim_{\sigma \rightarrow 0} \sum_{i=1}^n e^{-x_i^2/2\sigma^2} = \|x\|_0.$$

那么最小化模型 (2.9) 可以松弛为

$$\begin{aligned} \min F_{\sigma}(x) &= n - \sum_{i=1}^n e^{-x_i^2/2\sigma^2} \\ \text{s.t. } Ax &= b, \end{aligned} \quad (3.6)$$

这个方法称为 SL0 算法 (smoothed ℓ_0 norm minimization)^[11].

上述方法中采用的是 Gauss 函数, 也可以采用双曲正切函数

$$g_{\sigma}(t) = \frac{e^{t^2/2\sigma^2} - e^{-t^2/2\sigma^2}}{e^{t^2/2\sigma^2} + e^{-t^2/2\sigma^2}},$$

此时当 $\sigma \rightarrow 0$ 时, 有

$$\lim_{\sigma \rightarrow 0} G_{\sigma}(x) = \lim_{\sigma \rightarrow 0} \sum_{i=1}^n g_{\sigma}(x_i) = \lim_{\sigma \rightarrow 0} \sum_{i=1}^n \frac{e^{x_i^2/2\sigma^2} - e^{-x_i^2/2\sigma^2}}{e^{x_i^2/2\sigma^2} + e^{-x_i^2/2\sigma^2}} = \|x\|_0.$$

注 3.2 在逼近 ℓ_0 范数时, 双曲正切函数的效果要比 Gauss 函数好. 文献 [10] 给出了证明, 对任意的 σ 和任意的 x_i , 都有 $g_{\sigma}(x_i) \geq 1 - f_{\sigma}(x_i)$.

3.3 ℓ_q ($0 < q < 1$) 极小化模型

ℓ_q ($0 < q < 1$) 最小化模型为

$$\begin{aligned} \min \|x\|_q^q &= \sum_{i=1}^n |x_i|^q \\ \text{s.t. } Ax &= b. \end{aligned} \quad (3.7)$$

ℓ_q ($0 < q < 1$) 模型不再是一个凸优化问题, 因此在大多数情况下只能得到局部最优解, 但它往往可以产生比 ℓ_1 极小化模型更稀疏的解.

根据文献 [24, 30, 31], 有如下的结论: (1) 随着 q 值的降低, ℓ_q ($0 < q < 1$) 正则化生成更稀疏的解. (2) 当 $\frac{1}{2} < q < 1$ 时, 正则化性能随着 q 的减小显著提升; 当 $0 < q < \frac{1}{2}$ 时, 正则化的性能随着 q 的减小没有显著的差别. 因此, $\ell_{1/2}$ 正则化可以作为 ℓ_q ($0 < q < 1$) 正则化的代表.

为了分析 $\ell_{1/2}$ 范数极小化模型是不是可以代替 ℓ_0 范数极小化模型, 首先考虑下面这个问题:

$$\begin{aligned} \min f(x) &= \sqrt{x_1} + \sqrt{x_2} + \cdots + \sqrt{x_n} \\ \text{s.t. } x_1 + x_2 + \cdots + x_n &= k, \\ 0 &\leq x_i \leq 1. \end{aligned} \quad (3.8)$$

对应到数独问题中, $n = 729$, $k = 81$.

命题 3.1 问题 (3.8) 的最优解是 0-1 向量, 最优值为 81.

证明 由约束 $0 \leq x_i \leq 1$ 得 $x_i \leq \sqrt{x_i}$, $i = 1, 2, \dots, n$, 所以对任意可行的 x 有

$$f(x) = \sum_{i=1}^n \sqrt{x_i} \geq \sum_{i=1}^n x_i = k. \quad (3.9)$$

当 x 为 0-1 向量时, 即 x 中有 k 个元素为 1, 其余元素均为 0, 则 $f(x) = k$, 所以问题 (3.8) 的最优值为 k , 且所有的 0-1 可行向量都是最优解.

下证最优解都是 0-1 向量. 假设存在不是 0-1 向量的最优解 $y = (y_1, y_2, \dots, y_n)$, 不妨设 y 中前 m 个元素不是 0-1 的, 且和为 $t \in \mathbb{Z}^+$, 即

$$\begin{aligned} 0 < y_i < 1, \quad i = 1, 2, \dots, m, \\ \sum_{i=1}^m y_i = t. \end{aligned}$$

由 $0 < y_i < 1, i = 1, 2, \dots, m$ 得 $y_i < \sqrt{y_i}, i = 1, 2, \dots, m$, 所以,

$$f(y) = \sum_{i=1}^n \sqrt{y_i} = \sum_{i=1}^m \sqrt{y_i} + \sum_{i=m+1}^n \sqrt{y_i} > \sum_{i=1}^m y_i + k - t = k, \quad (3.10)$$

这与 y 是最优解矛盾, 所以假设不成立, 即问题 (3.8) 的最优解都是 0-1 向量. $\square$

本文用 $\ell_{1/2}$ 正则化模型代替模型 (2.9):

$$\begin{aligned} \min \quad & \|x\|_{1/2}^{1/2} = \sum_{i=1}^n |x_i|^{1/2} \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0. \end{aligned} \quad (3.11)$$

定理 3.2 对于数独问题, 模型 (3.11) 和 (2.9) 等价.

证明 问题 (3.8) 的可行域记为

$$C = \left\{ x \mid \sum_{i=1}^{729} x_i = 81, 1 \geq x_i \geq 0 \right\},$$

模型 (3.11) 的可行域 $D = \{x \mid Ax = b, x \geq 0\}$, 模型 (2.8) 的可行域记为

$$F = \{x \mid Ax = b, x_i \in \{0, 1\}\}.$$

设数独问题的唯一解为 x^* , 是一个 0-1 向量, 所以 x^* 为模型 (2.8) 的唯一可行点, 即 $F = \{x^*\}$. 由于 (2.7) 对 $k = 1, 2, \dots, 81$ 都成立, 可得

$$\begin{aligned} 0 \leq x_i \leq 1, \quad i = 1, 2, \dots, 729, \\ \sum_{k=1}^{81} \left(\sum_{i=1}^9 e_{9k-9+i} \right) x = \sum_{i=1}^{729} x_i = 81. \end{aligned}$$

所以,

$$C \supset D \supset F = \{x^*\},$$

则 x^* 是 D 中唯一 0-1 可行点, 下证这个 0-1 可行点是模型 (3.11) 唯一解点.

设问题 (3.8) 的最优值为 f_1^* , 模型 (3.11) 的最优值为 f_2^* , 由于 $C \supset D$, 所以,

$$f_1^* \leq f_2^* \leq \|x^*\|_{1/2}^{1/2} = 81,$$

由命题 3.1 知, $f_1^* = 81$, 所以 $f_2^* = 81$, 因此, x^* 是模型 (3.11) 的最优解, 且模型 (3.11) 的解一定是问题 (3.8) 的解.

最后证明模型 (3.11) 的最优解是唯一的. 假设存在 $\tilde{x}^*$ 也是模型 (3.11) 的解且 $\tilde{x}^* \neq x^*$, 则 $\tilde{x}^*$ 也是问题 (3.8) 的解. 由命题 3.1 得 $\tilde{x}^*$ 是 0-1 向量, 那么 $\tilde{x}^*$ 是模型 (2.8) 的可行点. 这与数独解唯一是矛盾的, 所以, x^* 是模型 (3.11) 的唯一解.

令 ℓ_0 模型 (3.11) 的解集为 Ω_s^* , $\ell_{1/2}$ 模型 (2.9) 的解集为 Ω_h^* , 则有以下结论:

$$\{x^*\} = \Omega_s^* = \Omega_h^*,$$

即求解数独这样一个特殊的问题, $\ell_{1/2}$ 模型 (3.11) 和 ℓ_0 模型 (2.9) 等价. $\square$

求解 $\ell_{1/2}$ 极小化模型, 文献 [24, 30, 31] 提出了半迭代阈值算法和求解一系列 LASSO (least absolute shrinkage and selection operator) 问题的方法, 但本文中子问题是一个标准线性规划问题, 更易于求解. 因为 $\ell_{1/2}$ 可以近似看成加权的 ℓ_1 问题, 其加权矩阵为 W , 是一个对角矩阵, 设 $W = \text{diag}(W_1, W_2, \dots, W_n)$, 其中 $W_i = (\varepsilon + |x_i|)^{-1/2}$, 当 $\varepsilon \rightarrow 0$ 时, 有

$$\|Wx\|_1 = \sum_{i=1}^n |W_i x_i| \longrightarrow \sum_{i=1}^n |x_i|^{1/2} = \|x\|_{1/2}^{1/2},$$

所以可以用如下模型近似求解:

$$\begin{aligned} \min \quad & \|Wx\|_1 \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0. \end{aligned} \quad (3.12)$$

接着用算法 1 求解模型 (3.12), 步 4 中 x^{k+1} 不一定是 0-1 向量, 将 x^{k+1} 重新编码为 0-1 向量, 即将 x^{k+1} 中的每组 9 位编码中分量最大的数令为 1, 其余 8 位编码都为 0, 这样得到的 0-1 向量记为 $\tilde{x}^{k+1}$.

算法 1 $\ell_{1/2}$ -SLP 算法

准备:

给定 $\beta \in (0, 1)$, $\varepsilon^0 < 1$, 给定矩阵 A 和向量 b , 最大迭代次数 MaxIter , 误差容限 TOL .

给定一个初始点 x^0 , 设 $k := 0$.

步骤:

- 1: 如果 $k < \text{MaxIter}$, 进行步 2-6.
- 2: 令 $\varepsilon^{k+1} = \beta \varepsilon^k$, 加权系数 $w_i^{k+1} = (\varepsilon^{k+1} + |x_i^k|)^{-1/2}$, $i = 0, 1, 2, \dots, n$.
- 3: 令 $c = [w_1^{k+1}, w_2^{k+1}, \dots, w_n^{k+1}]^T$.
- 4: 求解以下子问题得到 x^{k+1} :

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0. \end{aligned}$$

5: 将 x^{k+1} 编码得到 $\tilde{x}^{k+1}$.

6: 如果 $A\tilde{x}^{k+1} = b$ 或者 $\|x^{k+1} - \tilde{x}^{k+1}\| < \text{TOL}$, 停止并返回 $\tilde{x}^{k+1}$, 否则 $k := k + 1$, 转步 1.

注 3.3 $\ell_{1/2}$ 也可以近似看成加权的 ℓ_2 问题, 其加权矩阵为 $\bar{W}$, 是一个对角矩阵, 设

$$\bar{W} = \text{diag}(\bar{W}_1, \bar{W}_2, \dots, \bar{W}_n),$$

其中 $\bar{W}_i = (\varepsilon^2 + |x_i|^2)^{-3/4}$, 当 $\varepsilon \rightarrow 0$ 时, 有

$$\|\overline{W}x\|_2^2 = \sum_{i=1}^n |\overline{W}_i x_i|^2 \longrightarrow \sum_{i=1}^n |x_i|^{1/2} = \|x\|_{1/2}^{1/2},$$

即求解

$$\begin{aligned} \min \quad & \|\overline{W}x\|_2^2 \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0, \end{aligned} \quad (3.13)$$

这种方法也被称作迭代调整权值最小二乘 (IRLS) [23, 32].

4 数值实验与分析

计算机环境: 中央处理器: Intel Core i5; 内存: 8GB; 系统类型: 64 位操作系统.

实验平台: MATLAB R2016b.

对于数独问题, 在保证唯一解的基础上, 提示数越少难度越高, 目前通过计算机穷举方法证明了数独有唯一解的最小提示数目为 17, 本文的实验数据是 47,621 个提示数为 17 的数独 (实验数据已上传至 <https://pan.baidu.com/s/1yVE3cPulVeTXG0S7cWwXrw>).

对于数独问题, 我们用前文中提到的 3 种方式分别求解并对比分析. (1) 将 ℓ_0 极小化问题松弛为 ℓ_1 极小化问题, 然后用 BP 算法、ADMM 算法和 PPA 算法求解. (2) 用双曲正切函数来近似 ℓ_0 范数, 并用序列二次规划 (sequential quadratic programming, SQP) 方法求解近似后的模型. (3) 将 ℓ_0 极小化问题松弛为 $\ell_{1/2}$ 极小化问题, 再用 $\ell_{1/2}$ -SLP 算法求解. 算例中, 参数 $\beta = \frac{1}{2}$, $\varepsilon^0 = \frac{1}{2}$, MaxIter = 10, TOL = 10^{-4} , 求解线性规划子问题时, 调用 Matlab 优化工具箱中的 linprog 函数.

我们将用 3 种方式分别求解相同的 100、500、1,000 和 10,000 个数独, 并从运行的成功率和时间进行对比, 见表 2. 我们发现 BP、PPA 和 ADMM 在成功率上都是相差无几, 但 BP 最快, PPA 最慢. SL0 算法虽然在成功率上与凸松弛 ℓ_1 模型相差无几, 但是在时间上已经不在一个数量级, 所以后续时间对比时将不考虑 SL0. 本文采用的 $\ell_{1/2}$ -SLP 算法在时间上略慢于 BP 算法, 但成功率远超 BP 算法, 即使在求解 10,000 个数独的时候成功率也达到了 95.72%. 为了直观地看出性能的差距, 取五组数据, 每组 1,000 个, 然后对 4 种算法进行了比较, 见图 1. 在成功率上, 除 $\ell_{1/2}$ -SLP 算法外, 其他 3 种算法成功率均一致, 从时间和成功率综合来看 $\ell_{1/2}$ -SLP 算法效果最好. 用 BP、ADMM、PPA、SL0(tanh) 和 $\ell_{1/2}$ -SLP 算法将 47,621 个数独全部求解一遍, 从时间和成功率方面验证了 $\ell_{1/2}$ -SLP 算法的高效性, 见表 3. $\ell_{1/2}$ -SLP 算法在求解全部 47,621 个提示数为 17 的数独时成功率达到了 95.69%, 比其他算法高了约 10%, 而且速度也很快.

表 2 算法运行效率对比

实验次数	100		500		1,000		10,000	
	效率	成功数 总时间 (秒)	成功数 总时间 (秒)	成功数 总时间 (秒)	成功数 总时间 (秒)	成功数 总时间 (秒)	成功数 总时间 (秒)	成功数 总时间 (秒)
BP		86 4.58	443 23.55	864 45.45	8,505 461.21			
ADMM		86 6.51	443 32.17	864 64.96	8,505 667.83			
PPA		86 7.85	443 38.36	864 76.35	8,505 755.10			
SL0(tanh)		86 87.90	443 439.20	864 887.80	8,505 9,940.60			
$\ell_{1/2}$ -SLP		98 5.47	486 27.78	959 59.45	9,572 594.34			

注: SL0(tanh) 算法效率的总时间与别的算法差一个数量级.

选取一个数独问题, 见图 2(a), 其正确解为图 2(b). 经过数值实验可以发现只有 $\ell_{1/2}$ 正则化能够解出正确解. 图 3 为 BP 算法、PPA 算法、ADMM 算法、 $\ell_{1/2}$ -SLP 算法、SL0(Gauss) 算法和 SL0(tanh) 算法得到的解, 除了 $\ell_{1/2}$ -SLP 算法外其余 5 种算法都没有得到正确解. 对于其他算法能求解的数独问题, $\ell_{1/2}$ -SLP 算法都能够正确快速地求解.

图 1 (网络版彩图) 成功率和时间对比. (a) 五组 1,000 个数独成功率对比; (b) 五组 1,000 个数独总时间对比 (秒)

表 3 47,621 个数独求解效率对比

效率	成功次数	成功率	总时间 (秒)	平均时间 (秒)
BP	40,459	84.96%	2,195.83	0.0461
ADMM	40,459	84.96%	3,672.90	0.0771
PPA	40,459	84.96%	3,861.01	0.0812
SL0(tanh)	40,459	84.96%	37,730.50	0.7923
$\ell_{1/2}$ -SLP	45,567	95.69%	2,855.68	0.0600

注: SL0(tanh) 算法效率的总时间与别的算法差一个数量级.

图 2 数独问题. (a) 初始问题; (b) 正确解

3	4	1	8	6	2	7	9	5
6	2	8	7	8	4	1	4	3
7	1	7	4	1	4	8	6	2
4	5	8	2	9	1	9	5	7
8	9	2	4	7	6	2	3	1
5	1	3	4	3	8	9	2	7
9	8	9	1	4	2	2	7	1
1	7	5	9	2	7	3	8	4
2	7	4	8	1	7	3	1	9

(a)

3	4	1	8	6	2	7	9	5
5	2	8	7	2	4	1	4	3
7	1	7	4	1	9	8	6	2
4	5	8	2	9	1	9	5	7
8	9	2	4	7	6	2	3	1
5	1	3	4	3	8	9	2	7
9	8	9	1	4	2	2	7	1
1	7	5	9	2	7	4	8	4
2	7	4	8	1	7	3	1	9

(b)

3	4	1	8	6	2	7	9	5
6	2	8	7	2	5	1	4	3
7	5	9	3	1	9	8	6	2
4	3	7	2	9	1	6	5	8
8	9	2	5	7	6	2	3	1
5	1	6	6	3	8	9	2	7
9	8	3	1	4	5	2	7	6
1	7	5	9	2	7	3	8	4
2	7	4	8	8	7	5	1	9

(c)

3	4	1	8	6	2	7	9	5
6	2	8	7	5	9	1	4	3
7	5	9	3	1	4	8	6	2
4	3	7	2	9	1	6	5	8
8	9	2	5	7	6	4	3	1
5	1	6	4	3	8	9	2	7
9	8	3	1	4	5	2	7	6
1	6	5	9	2	7	3	8	4
2	7	4	6	8	3	5	1	9

(d)

3	4	1	8	6	2	7	9	5
5	2	8	7	8	4	1	4	3
7	1	7	5	1	4	8	6	2
4	5	6	2	9	1	9	5	7
8	9	2	4	7	6	2	3	1
4	1	3	9	3	8	6	2	7
9	8	9	1	4	5	2	7	1
1	7	5	9	2	7	4	8	4
2	7	4	8	1	7	3	1	9

(e)

3	4	1	8	6	2	7	9	5
9	2	8	7	8	4	1	6	3
3	1	7	3	1	9	8	4	2
7	5	6	2	3	1	9	5	4
8	9	2	4	7	6	2	3	1
5	1	3	4	9	8	6	2	7
9	3	9	1	4	2	2	7	1
1	6	5	3	2	7	4	8	9
2	7	4	6	1	3	3	1	9

(f)

图 3 不同算法数独的解. (a) BP; (b) PPA; (c) ADMM; (d) $\ell_{1/2}$ -SLP; (e) SL0(Gauss); (f) SL0(tanh)

5 结论

将数独这样一个整数规划问题, 转化为 ℓ_0 极小化问题这样一个稀疏优化模型, 并松弛为其他模型, 这种方法是求解数独问题一种快速且有效的方法. 本文证明对于数独这样的特殊问题, ℓ_0 极小化模型和 $\ell_{1/2}$ 极小化模型等价, 并用 $\ell_{1/2}$ -SLP 方法求解 $\ell_{1/2}$ 模型, 对求解数独问题的速度和正确率有了很大的提升.

参考文献

- 1 Dai W, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans Inform Theory*, 2009, 55: 2230–2249
- 2 Jiao L C, Yang S Y, Liu F, et al. Development and prospect of compressive sensing (in Chinese). *Acta Electron Sin*, 2011, 39: 1651–1662 [焦李成, 杨淑媛, 刘芳, 等. 压缩感知回顾与展望. *电子学报*, 2011, 39: 1651–1662]
- 3 Wen Z W, Yin W T, Liu X, et al. Introduction to compressive sensing and sparse optimization (in Chinese). *Oper Res Trans*, 2012, 16: 49–64 [文再文, 印卧涛, 刘歆, 等. 压缩感知和稀疏优化简介. *运筹学学报*, 2012, 16: 49–64]
- 4 Tibshirani R. Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B Stat Methodol*, 1996, 58: 267–288
- 5 Zhou Z H, Wang J. *Machine Learning and Its Applications 2009* (in Chinese). Beijing: Tsinghua University Press, 2009 [周志华, 王珏. *机器学习及其应用 2009*. 北京: 清华大学出版社, 2009]
- 6 Mallat S G, Zhang Z F. Matching pursuits with time-frequency dictionaries. *IEEE Trans Signal Process*, 1993, 41: 3397–3415
- 7 Needell D, Vershynin R. Greedy signal recovery and uncertainty principles. In: *Proceedings of the International Society for Optical Engineering*, vol. 6814. San Jose: SPIE, 2008, 68140J
- 8 Tropp J A, Gilbert A C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans Inform Theory*, 2007, 53: 4655–4666
- 9 Blumensath T, Davies M E. Iterative hard thresholding for compressed sensing. *Appl Comput Harmon Anal*, 2009, 27: 265–274
- 10 Zhao R Z, Lin W J, Li H, et al. Reconstruction algorithm for compressive sensing based on smoothed ℓ_0 norm and revised Newton method (in Chinese). *J Comput Aided Design Comput Graph*, 2012, 24: 478–484 [赵瑞珍, 林婉娟, 李浩, 等. 基于光滑 ℓ_0 范数和修正牛顿法的压缩感知重建算法. *计算机辅助设计与图形学学报*, 2012, 24: 478–484]
- 11 Mohimani H, Babaie-Zadeh M, Jutten C. Fast sparse representation based on smoothed ℓ_0 norm. In: *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation*. London: Springer-Verlag, 2007, 389–396
- 12 Candès E J, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inform Theory*, 2006, 52: 489–509
- 13 Candès E J. The restricted isometry property and its implications for compressed sensing. *C R Math Acad Sci Paris*, 2008, 346: 589–592
- 14 Candès E J, Tao T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans Inform Theory*, 2006, 52: 5406–5425
- 15 Daubechies I, Fornasier M, Loris I. Accelerated projected gradient method for linear inverse problems with sparsity constraints. *J Fourier Anal Appl*, 2008, 14: 764–792
- 16 Lin C J. Projected gradient methods for nonnegative matrix factorization. *Neural Comput*, 2007, 19: 2756–2779
- 17 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci*, 2009, 2: 183–202
- 18 Yang J F, Zhang Y. Alternating direction algorithms for ℓ_1 -problems in compressive sensing. *SIAM J Sci Comput*, 2011, 33: 250–278
- 19 Yin W T, Osher S, Goldfarb D, et al. Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing. *SIAM J Imaging Sci*, 2008, 1: 143–168
- 20 Gorodnitsky I F, Rao B D. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Trans Signal Process*, 1997, 45: 600–616
- 21 Rao B D, Kreutz-Delgado K. An affine scaling methodology for best basis selection. *IEEE Trans Signal Process*, 1999, 47: 187–200
- 22 Wang T J, Wan X L, Guan X J, et al. An affine scaling steepest descent algorithm for target localization in wireless

- sensor networks. IEEE Access, 2018, 6: 47575–47586
- 23 Chartrand R, Yin W T. Iteratively reweighted algorithms for compressive sensing. In: Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. Las Vegas: IEEE, 2008, 3869–3872
- 24 Xu Z B, Chang X Y, Xu F M, et al. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. IEEE Trans Neural Netw Learn Syst, 2012, 23: 1013–1027
- 25 Babu P, Pelckmans K, Stoica P, et al. Linear systems, sparse solutions, and Sudoku. IEEE Signal Process Lett, 2010, 17: 40–42
- 26 Zhang Y D, Wang S H, Huo Y K, et al. A novel Sudoku solving method based on sparse optimization (in Chinese). J Nanjing Univ Inf Sci Technol Nat Sci, 2011, 3: 23–27 [张煜东, 王水花, 霍元恺, 等. 一种基于稀疏优化的数独求解新方法. 南京信息工程大学学报, 2011, 3: 23–27]
- 27 Gribonval R, Nielsen M. Sparse representations in unions of bases. IEEE Trans Inform Theory, 2003, 49: 3320–3325
- 28 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn, 2011, 3: 1–122
- 29 He B S, Yuan X M, Zhang W X. A customized proximal point algorithm for convex minimization with linear constraints. Comput Optim Appl, 2013, 56: 559–572
- 30 Xu Z B. Data modeling: Visual psychology approach and $\ell_{1/2}$ regularization theory. In: Proceedings of the International Congress of Mathematicians. Hyderabad: World Scientific, 2010, 3151–3184
- 31 Xu Z B, Zhang H, Wang Y, et al. $L_{1/2}$ regularization. Sci China Inf Sci, 2010, 53: 1159–1169
- 32 Lu Z S. Iterative reweighted minimization methods for ℓ_p regularized unconstrained nonlinear programming. Math Program, 2014, 147: 277–307

Application of sparse optimization in Sudoku

Yongxin Chen, Xingju Cai & Bo Jiang

Abstract Sudoku is a difficult integer linear programming problem. The restriction of integer constraint is eliminated by real number coding and the integer linear programming model can be converted to the ℓ_0 norm minimization model. The existing algorithms are mostly used to solve the relaxed ℓ_1 norm minimization model, but by the ℓ_1 model some Sudoku problems cannot be solved. We prove that the ℓ_q ($0 < q < 1$) norm minimization model is equivalent to the ℓ_0 norm minimization model for a particular Sudoku problem and the $\ell_{1/2}$ model is solved by $\ell_{1/2}$ -SLP (sequential linear programming) algorithm. In numerical experiments, $\ell_{1/2}$ -SLP algorithm can solve more Sudoku problems. The efficiency of the algorithm is verified in terms of time and success rate.

Keywords real number coding, sparse optimization, ℓ_0 norm minimization model, ℓ_q ($0 < q < 1$) norm minimization model, $\ell_{1/2}$ -SLP algorithm

MSC(2020) 90C25, 90C26, 90C90

doi: 10.1360/SSM-2020-0006