DOI:10.19789/j.1004-9398.2018.04.004

一个求解绝对值方程组稀疏解问题的不动点算法

王 鹏 任 天*
(天津大学数学学院,天津 300354)

摘 要

本文提出基于 prox 算子不动点算法(fixed-point algorithm)求解 NP 难的绝对值方程组 $Ax - |x| = b$ 的最稀疏解. 该算法首先将问题松弛为 ℓ_1 范数最小化问题,利用外罚函数法进一步松弛为一个无约束优化问题,其次求解近似后的无约束优化问题.

关键词: l_1 范数最小化,稀疏解,prox 算子,绝对值方程组.
中图分类号: O221.2

0 引 言

绝对值方程组是一类特殊的非线性、非光滑方程组,它与互补问题之间有着紧密联系,互补问题可以说是绝对值方程的主要来源之一. 互补问题属于运筹学和计算数学的交叉领域,在力学、工程、经济、交通等诸多领域都有着广泛的应用. Mangasarian [1] 指出了在一定条件下,绝对值方程组与线性互补问题是等价的,这是非常重要的一个结果. 另外,绝对值方程组还涵盖了线性规划、二次规划、双矩阵对策等许多方面的问题,在金融、工程等许多领域有着广泛的应用. 因此,绝对值方程组问题是具有重要实际应用背景、非常值得研究的问题,近年来国内外学者对绝对值方程组的求解及其它的性质有着浓厚的兴趣,得到了很多好的研究成果[1-4].

"稀疏"是近年来优化领域比较热的话题,越来越多的国内外学者来研究各种问题的稀疏解. 在现有的研究成果中,更多的是关于线性方程组稀疏解的研究[5]. Chen 和 Huang 在文献[6]中提出了基于 prox 算子的不动点算法,并给出收敛性分析,在实际应用中证明有很好的运算效果,已成为学者们关注的热点.

本文利用 prox 算子不动点算法寻求绝对值方程组的稀疏解,绝对值方程组的形式如下表示

$$Ax - |x| = b, \tag{1}$$

其中 $A \in R^{n\times n}, b \in R^n, x \in R^n$, $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^{\mathrm{T}} \in R^n$. 在压缩感知领域中求解绝对值方程组的稀疏解通常考虑求解如下的约束优化问题 $\min\limits_{x\in R^n} \|x\|_0 \text{ s.t. } Ax - |x| = b$,其中 $\|x\|_0$ 称为 x 的 ℓ_0 范数,表示向量 x 的非零分量的个数. 然后将其松弛为 ℓ_1 范数最小化问题,即

$$\min_{x\in R^n}\{\|x\|_1 : Ax - |x| = b\}. \tag{2}$$

然后利用外罚函数法,进一步将其松弛为如下的无约束问题

$$\min_{x\in R^n} \|x\|_1 + \frac{1}{2\mu}\|Ax - |x| - b\|_2^2 \tag{3}$$

其中 $\mu > 0$ 是外罚因子. 本文讨论利用 prox 算子的不动点算法解决上述优化问题(3).

1 基于 prox 算子的不动点算法

1.1 prox 算子

对于问题(3),令 $f(x) = \|x\|_1, h(x) = \frac{1}{2\mu}\|Ax - |x| - b\|_2^2$. Combettes 和 Wajs 在文献[7]中提出近端向前分裂(PFBS)的算法如下:

$$x_{k+1} = \mathrm{prox}_{\gamma f}(x_k - \gamma \nabla h(x_k))$$

其中 $\gamma > 0$ 是步长参数. 算子 prox_f 的定义如下

$$\mathrm{prox}_f : \chi \to \chi$$

$$x \longmapsto \operatorname{argmin}_{y\in\chi} f(y) + \frac{1}{2}\|x - y\|_2^2$$

收稿日期:2017-06-20
* 通信作者

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

我们知道,这种分离算法最早是为解决偏微分方程而提出研究[8-9]的,proximity 算子的概念最早是由 Morean[10] 作为一种投影算子的推广而引用的.

需要注意的是: 本文中 $h(x)$ 不一定是凸的,在这里我们采取 $h(x)$ 的其中一个次梯度,即 $\nabla h(x^k)$ 是 $h(x)$ 在 $x = x^k$ 的一个次梯度. 求解如下

$$\begin{aligned} h(x) &= \frac{1}{2\mu}\| Ax - |x| - b\|_2^2 \\ &= \frac{1}{2\mu}(Ax - |x| - b)'(Ax - |x| - b) \\ &= \frac{1}{2\mu}(x'A'Ax - x'A'|x| - x'A'b - |x|'Ax + \\ &\quad \mu|x|'|x| + |x|'b - b'Ax + \\ &\quad b'|x| + \| b\|_2^2) \\ &= \frac{1}{2\mu}(x'A'Ax - x'A'b - b'Ax) + \\ &\quad \frac{1}{2\mu}(-x'A'|x| - |x|'Ax + |x|'|x| + \\ &\quad |x|'b + b'|x|) + \frac{1}{2\mu}\| b\|_2^2. \end{aligned} \tag{4}$$

对于可微函数而言,次梯度等于梯度. 下面,求解(4)中那些不可微项的次梯度. 注意到: $|x_i|$ 的次微分 $\partial|x_i| = \begin{cases} \{1\}, & x_i > 0 \\ [-1,1], & x_i = 0 \\ \{-1\}, & x_i < 0 \end{cases}$,显然符号函数 $\mathrm{sign}(x_i)$ 是 $|x_i|$ 的一个次导数. 通过该次导数,容易得到 $\mathrm{subgrad}(|x|'b) = \mathrm{diag}(\mathrm{sign}(x))b = D(x)b$,其中,$\mathrm{sign}(x) = (\mathrm{sign}(x_1), \mathrm{sign}(x_2), \cdots \mathrm{sign}(x_n))'$

接下来,利用符号函数 $\mathrm{sign}(x_i)$ 是 $|x_i|$ 的一个次导数,求 $|x|'Ax$ 的次梯度.

$$\begin{aligned} &|x|'Ax \\ &= [|x_1| \quad |x_2| \quad \cdots \quad |x_n|] \\ &\quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} |x_i| a_{i1}x_1 + \\ &\quad \sum_{i=1}^{n} |x_i| a_{i2}x_2 + \cdots + \sum_{i=1}^{n} |x_i| a_{in}x_n, \end{aligned}$$

则 $|x|'Ax$ 的一个次梯度为

$$\begin{aligned} \frac{\partial |x|'Ax}{\partial x} &= \begin{bmatrix} 2a_{11}|x_1| + |x_2|a_{21} + \cdots + |x_n|a_{n1} \\ |x_1|a_{12} + 2|x_2|a_{22} + \cdots + |x_n|a_{n2} \\ \vdots \\ |x_1|a_{1n} + |x_2|a_{2n} + \cdots + 2a_{nn}|x_n| \end{bmatrix} + \begin{bmatrix} \mathrm{sign}(x_1)a_{12}x_2 + \mathrm{sign}(x_1)a_{13}x_3 + \cdots + \mathrm{sign}(x_1)a_{1n}x_n \\ \mathrm{sign}(x_2)a_{21}x_1 + \mathrm{sign}(x_2)a_{23}x_3 + \cdots + \mathrm{sign}(x_2)a_{2n}x_n \\ \vdots \\ \mathrm{sign}(x_n)a_{n1}x_1 + \mathrm{sign}(x_n)a_{n2}x_2 + \cdots + \mathrm{sign}(x_n)a_{nn-1}x_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}|x_1| + |x_2|a_{21} + \cdots + |x_n|a_{n1} \\ a_{12}|x_1| + |x_2|a_{22} + \cdots + |x_n|a_{n2} \\ \vdots \\ a_{1n}|x_1| + |x_2|a_{2n} + \cdots + |x_n|a_{nn} \end{bmatrix} + \begin{bmatrix} \mathrm{sign}(x_1)a_{11}x_1 + \mathrm{sign}(x_1)a_{12}x_2 + \cdots + \mathrm{sign}(x_1)a_{1n}x_n \\ \mathrm{sign}(x_2)a_{21}x_1 + \mathrm{sign}(x_2)a_{22}x_2 + \cdots + \mathrm{sign}(x_2)a_{2n}x_n \\ \vdots \\ \mathrm{sign}(x_n)a_{n1}x_1 + \mathrm{sign}(x_n)a_{n2}x_2 + \cdots + \mathrm{sign}(x_n)a_{nn}x_n \end{bmatrix} \\ &= A'|x| + DAx. \end{aligned}$$

由以上分析我们得到 $h(x)$ 在 $x = x^k$ 处的一个次梯度 $\nabla h(x^k)$ 的表达式为:

$$\begin{aligned} \nabla h(x^k) = \frac{1}{\mu}(&A'Ax^k - A'b + x^k - \\ &(D^kAx^k + A'|x^k|) + D^kb), \end{aligned} \tag{5}$$

其中 $D^k = D(x^k)$.

1.2　不动点算法

类似于 Chambolle[11] 提出的基于 ROF 去噪模型的不动点算法,Micchelli 等提出一种叫做 FP^{2}O[12] 的方法求解 proximity 算子,本文提出利用 FP^{2}O 的方法求解 $x_{k+1} = \mathrm{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k))$

首先定义算子

$$\begin{aligned} &\mathrm{A}v: x_k - \gamma\nabla h(x_k) + (1-\lambda)v, \\ &\mathrm{H}: (\mathrm{I} - \mathrm{prox}_{\frac{\gamma}{\lambda}f}) \circ \mathrm{A}. \end{aligned} \tag{6}$$

由文献[12]可以得到如下定理:

定理 1　$\mathrm{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k)) = x_k - \gamma\nabla h(x_k) - \lambda v$ 等价于 $\mathrm{A}v: x_k - \gamma\nabla h(x_k) + (1-\lambda)v$,$\mathrm{H}: (\mathrm{I} - \mathrm{prox}_{\frac{\gamma}{\lambda}f}) \circ \mathrm{A}$,向量 v 是算子 H 的不动点.

证明: 若对于向量 v,$v \in \partial\left(\frac{\gamma}{\lambda}f\right)(\mathrm{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k)))$,使得算子 $\mathrm{prox}_{\gamma f}$ 满足 $\mathrm{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k)) = x_k - \gamma\nabla h(x_k) - \lambda v$,则可推出 $v \in \partial\left(\frac{\gamma}{\lambda}f\right)(x_k - \gamma\nabla h(x_k) - \lambda v)$,由此可得 $x_k -$

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

$\gamma\nabla h(x_k) - \lambda v = \text{prox}_{\frac{\gamma}{\lambda}f}(x_k - \gamma\nabla h(x_k) - \lambda v + v)$，即 $x_k - \gamma\nabla h(x_k) - \lambda v = \text{prox}_{\frac{\gamma}{\lambda}f}(x_k - \gamma\nabla h(x_k) + (1-\lambda)v)$，所以

$$\begin{aligned} v &= x_k - \gamma\nabla h(x_k) + (1-\lambda)v - \\ &\quad \text{prox}_{\frac{\gamma}{\lambda}f}(x_k - \gamma\nabla h(x_k) + (1-\lambda)v) \\ &= (I - \text{prox}_{\frac{\gamma}{\lambda}f})(x_k - \gamma\nabla h(x_k) + (1-\lambda)v). \end{aligned}$$

很显然，v 是算子 H 的不动点.

由以上定理以及定义的(6)式可得

$$\text{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k)) = x_k - \gamma\nabla h(x_k) - \lambda v. \tag{7}$$

最后得到迭代模式:

$$\begin{cases} v_{k+1} = (\mathrm{I} - \text{prox}_{\frac{\gamma}{\lambda}f})(x_k - \gamma\nabla h(x_k) + (1-\lambda)v_k) \\ x_{k+1} = x_k - \gamma\nabla h(x_k) - \lambda v_k. \end{cases} \tag{8}$$

即可以推出本文提出的基于 proximity 算子的不动点算法.

算法: 基于 proximity 算子的不动点算法

第一步: 取 $x_0 \in R^n, v_0 \in R^n, 0 < \lambda < 1, \gamma > 0$

第二步: 对于 $k = 0,1,2,\cdots$

$$x_{k+\frac{1}{2}} = x_k - \gamma\nabla h(x_k)$$

$$v_{k+1} = (\mathrm{I} - \text{prox}_{\frac{\gamma}{\lambda}f})(x_{k+\frac{1}{2}} + (1-\lambda)v_k)$$

$$x_{k+1} = x_{k+\frac{1}{2}} - \lambda v_{k+1}$$

结束

2 数值实验

随机生成 k 稀疏向量 x，这里的 x 是非零分量个数小于或等于 k，非零分量的均值为零，方差为 1 的 i. i. d 高斯随机变量. 令 $b = Ax - |x|$ 生成向量 b，随即生成一个问题. 其中 A 是 Walsh-Hadamard 变换矩阵. 实验中，取罚参数 $\mu = 10^{-3}$，参数 $\lambda = 0.9$，选取的初始点为 $x^0 = (1,\cdots,1)' \in R^n$，$v^0 = (1,\cdots,1)' \in R^n$，利用文中提出的迭代模式(8)进行迭代，规定最大迭代次数 Imax 的值为 100 次. 令 $\varepsilon = 10^{-6}$，当迭代次数达到设定的最大迭代次数或迭代之后的稀疏解满足 $\|x^{k+1} - x^k\|_2 \leqslant \varepsilon$，则终止迭代. 如果我们找到解 x^k 满足 $\|x^k\|_0 \leqslant k$(这里 $\|x^k\|_0$ 表示 x^k 中满足 $|x_i^k| \geqslant 10^{-5}$ 的分量个数)，并且 $\|x^k - x\|_2 \leqslant 10^{-5}$，就认为文中提出的不动点算法能够成功恢复 x.

稀疏度 $k = 25$，问题规模为 $n = 512, n = 1\,024, n = 2\,048, n = 4\,096, n = 8\,192$ 时，分别生成 100 个问题，计算恢复率，实验结果如表 1 所示.

由表可知，选择不同的参数 γ，恢复的成功率都几乎达到 100%. 数据结果表明此方法能够找到绝对值方程组的稀疏解，并且对于较大规模的问题也能有很好的恢复效果，随着问题规模的增大要求参数 γ 的取值反而要随之减小才能达到好的恢复效果.

对于稀疏度 $k = 25$，问题的规模 $n = 512$ 进行计算. 改变参数 γ 的值，每次生成 100 个问题，研究成功率的变化. 实验结果如表 2 所示.

表 1 不同的规模计算的恢复率

γ	2.3×10^{-6}	1.5×10^{-6}	9.0×10^{-7}	4.5×10^{-7}	6.5×10^{-8}	3.5×10^{-8}	2.0×10^{-8}
$n = 512$	100%	100%	88%	45%	0	0	0
$n = 1\,024$	0	100%	100%	100%	0	0	0
$n = 2\,048$	0	0	100%	100%	1%	0	0
$n = 4\,096$	0	0	0	100%	100%	100%	0
$n = 8\,192$	0	0	0	0	100%	100%	100%

表 2 不同参数的 γ 值计算的恢复率

γ	3.5×10^{-6}	3.1×10^{-6}	2.3×10^{-6}	1.3×10^{-6}	1.0×10^{-6}	6.0×10^{-7}	5.0×10^{-7}
恢复率	45%	91%	100%	100%	96%	66%	36%

由表 2 可以看出，当 $3.1\times10^{-6} \leqslant \gamma \leqslant 1.0\times10^{-6}$ 时，成功率较高，几乎能接近 100%，而当 $\gamma < 3.1\times10^{-6}$ 或 $\gamma > 1.0\times10^{-6}$ 时，运算效果较差;

再对于稀疏度 $k = 25$，问题的规模为 $n = 1\,024$ 进行计算. 同样每次生成 100 个问题，当 γ 的值不同时，恢复的成功率如下表表 3 所示.

表 3 不同参数的 γ 值计算的恢复率

γ	1.79×10^{-6}	1.78×10^{-6}	1.7×10^{-6}	1.0×10^{-6}	2.0×10^{-7}	1.6×10^{-7}	1.3×10^{-7}
恢复率	22%	99%	100%	100%	100%	100%	2%

由表 3 可知，当 $1.78\times10^{-6} \leqslant \gamma \leqslant 2.0\times10^{-7}$，成功率较高，几乎能达到 100%，而当 $\gamma < 1.7\times10^{-6}$

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

或 $\gamma > 1.0 \times 10^{-7}$ 时,运算效果较差.

对于其他规模的问题,改变参数 γ 的值,得到了类似的实验结果.

当问题的规模 $n = 512$,稀疏度 $k = 5,10,15,20,25$ 时,分别生成 100 个问题,令参数 γ 取不同的值,$\gamma = 6.0 \times 10^{-7}$,$\gamma = 5.0 \times 10^{-7}$,$\gamma = 4.0 \times 10^{-7}$,得如下实验结果.

表 4　不同稀疏度计算的恢复率

k	5	10	15	20	25
$\gamma = 6.0 \times 10^{-7}$	100%	100%	100%	98%	67%
$\gamma = 5.0 \times 10^{-7}$	100%	100%	99%	98%	50%
$\gamma = 4.0 \times 10^{-7}$	100%	100%	97%	95%	20%

当问题的规模 $n = 512$,稀疏度 $k = 5,10,15,20,25,30$ 时,同样分别生成 100 个问题,参数 γ 取不同的值,恢复的成功率如表 5 所示.

表 5　不同稀疏度计算的恢复率

k	5	10	15	20	25	30
$\gamma = 1.78 \times 10^{-6}$	100%	99%	100%	100%	99%	97%
$\gamma = 1.79 \times 10^{-6}$	99%	98%	78%	45%	22%	5%

当 $n = 2\,048$ 和 $n = 4\,096$ 时,实验结果也有同样的规律,数据结果表明稀疏度 k 对成功率是有一定影响的.

3　结　论

基于 prox 算子的不动点算法求解绝对值方程组的稀疏解,结果显示能够得到很好恢复效果. 针对参数值的变化以及稀疏度的变化的数值实验结果表明参数 γ 对恢复的成功率有明显的影响,尤其对于规模较大的问题,参数 γ 的值发生极小的变化对恢复率都有很大的影响,说明参数 γ 的值对实验的影响很大,并且稀疏度对恢复效果也有一定的影响,向量越稀疏,恢复的成功率越高. 实验中,改变初始点为零向量或随机向量,同样有很好的恢复效果,说明初始点的选择对实验结果并无影响. 本研究的结果为求解绝对值方程组的稀疏解提供了有效的算法.

参　考　文　献

[1] Mangasarian O L. Absolute value programming [J]. Comput Optim Appl, 2007, 36(1): 43 - 53.

[2] Caccetta L, Qu B, Zhou G L. A globally and quadratically convergent method for absolute value equations [J]. Comput Optim Appl, 2011, 48(1): 45 - 58.

[3] Mangasarian O L. Absolute value equation solution via dual complementarity [J]. Optim Lett, 2013, 7(4): 625 - 630.

[4] Mangasarian O L. Absolute value equation solution via linear programming [J]. Optim Theory Appl, 2014, 161(3): 870 - 876.

[5] Zhang M, Huang Z H, Li Y F. The sparsest solution to the system of absolute value equations [J]. J Oper Res Soc China, 2015, 3(1): 31 - 35.

[6] Chen P, Huang J, Zhang X. A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration [J]. Inverse Problems, 2013, 29(2): 11 - 25.

[7] Combettes P L, Wajs V R. Signal recovery by proximal forward-backward splitting [J]. Multiscale Model Simul, 2005, 4(4): 33 - 45.

[8] Lions P L, Mercier B. Splitting algorithms for the sum of two nonlinear operator [J]. SIAM J Numer Anal, 1979, 16(6): 964 - 979.

[9] Pssty G B. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space [J]. J Math Anal Appl, 1979, 72(2): 383 - 390.

[10] Moreau J J. Fonctions convexes duales et points proximaux dans un espace hilbertien [J]. Comptes Rendus Hebdomadaires Des Séances De Lacadémie Des Sciences, 1962, 255: 2897 - 2899.

[11] Chambolle A. An algorithm for total variation minimization and applications [J]. J Math Imaging Vis, 2004, 20(1 - 2): 89 - 97.

[12] TMicchelli C A, Shen L, Xu Y. Proximity algorithms for image models: denoising [J]. Inverse Problems, 2011, 27(4): 9 - 45.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

A Fixed-point Algorithm for the Sparest Solution Problem of Absolute Value Equations

Wang Peng Ren Tian

(School of Mathematical Science, Tianjin University, Tianjin 300354)

Abstract

In this paper, the prox operator based on the fixed point algorithm was proposed in order to solve the sparsest solution of the NP-hard absolute value equations $Ax - |x| = b$. The algorithm first relaxed the question to a ℓ_1-norm minimization problem, it was further relaxed to an unconstrained optimization problem by exterior penalty function method, next the fixed-point algorithm of the prox operator was used to solve the approximate optimization problem.

Key words: ℓ_1-minimization, sparse solution, the prox operator, absolute value equations.

(上接第 3 页)

Designing and Programming of Carrying Robot Based on Arduino Technology (Ⅰ)

——Arduino Mainboard and Application of Infrared Sensor

Li Zhenyu

(School of Mathematical Sciences, Capital Normal University, Beijing 100048)

Abstract

Education robot competition of China is a national competition, which is organized by the China ai association for the participation of students from colleges and universities. This paper makes a technical analysis of the basic principle of Arduino main board and the application of infrared sensor. As the front end input block of carrying robot, the infrared sensors form the tracking and the distance measurement of moving object.

Key words: carrying robot, Arduino, infrared sensor, program design.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net