Proximal Iteratively Reweighted Algorithm with Multiple Splitting for Nonconvex Sparsity Optimization

Canyi Lu[1], Yunchao Wei[2], Zhouchen Lin[3], Shuicheng Yan[1]

[1]National University of Singapore, [2]Beijing Jiaotong University, [3]Peking University

Problem

This paper aims to solve the following general problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}) = \lambda f(\mathbf{g}(\mathbf{x})) + h(\mathbf{x}), \tag{1}$$

where $\lambda > 0$, and f, $\mathbf{g}$ and h satisfy the following conditions:

C1 $f(\mathbf{y})$ is nonnegative, concave and increasing.
Examples: nonconvex surrogates of ℓ_0-norm, e.g., ℓ_p-norm, logarithm function.

C2 $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^d$ is a nonnegative multi-dimensional function, such that

$$\min_{\mathbf{x}\in\mathbb{R}^n} \lambda\langle \mathbf{w}, \mathbf{g}(\mathbf{x})\rangle + \frac{1}{2}||\mathbf{x}-\mathbf{b}||_2^2,$$

is convex and can be cheaply solved for any given nonnegative $\mathbf{w} \in \mathbb{R}^d$.
Examples: $|\mathbf{x}|$ (absolute value of $\mathbf{x}$ element-wise).

C3 $h(\mathbf{x})$ is continuously differentiable with Lipschitz continuous gradient

$$||\nabla h(\mathbf{x}) - \nabla h(\mathbf{y})|| \le L(h)||\mathbf{x}-\mathbf{y}|| \quad \text{for any} \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,$$

$L(h) > 0$ is called the Lipschitz constant of ∇h.
Examples: squared loss and logistic loss.

C4 $\lambda f(\mathbf{g}(\mathbf{x})) + h(\mathbf{x}) \to \infty$ iff $||\mathbf{x}||_2 \to \infty$;

Proximal Iteratively REweighted (PIRE) Algorithm

Method: when updating $\mathbf{x}^{k+1}$, linearize f at $\mathbf{g}(\mathbf{x}^k)$ and h at $\mathbf{x}^k$, simultaneously.
Motivation: two key inequalities: (2) and (4).

- Since f is concave, $-f$ is convex. Linearize $-f$ at $\mathbf{g}(\mathbf{x}^k)$, we have

$$f(\mathbf{g}(\mathbf{x})) \le f(\mathbf{g}(\mathbf{x}^k)) + \langle \mathbf{w}^k, \mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^k)\rangle, \tag{2}$$

where $-\mathbf{w}^k$ is the subgradient of $-f$ at $\mathbf{g}(\mathbf{x}^k)$, i.e.,

$$\mathbf{w}^k \in -\partial\left(-f(\mathbf{g}(\mathbf{x}^k))\right). \tag{3}$$

- Since $\nabla h(\mathbf{x})$ is Lipschitz continuous, we have

$$h(\mathbf{x}) \le h(\mathbf{x}^k) + \langle \nabla h(\mathbf{x}^k), \mathbf{x}-\mathbf{x}^k\rangle + \frac{\mu}{2}||\mathbf{x}-\mathbf{x}^k||_2^2, \ \ \forall \mu \ge L(h). \tag{4}$$

The right hand side of (2) and (4) are used as surrogate functions of $f(\mathbf{g}(\mathbf{x}))$ and $h(\mathbf{x})$, respectively. Combining (2) and (4), we update $\mathbf{x}$ by

$$\begin{aligned}\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \lambda\langle \mathbf{w}^k, \mathbf{g}(\mathbf{x})\rangle + \langle \nabla h(\mathbf{x}^k), \mathbf{x}-\mathbf{x}^k\rangle + \frac{\mu}{2}||\mathbf{x}-\mathbf{x}^k||_2^2 \\ &= \arg\min_{\mathbf{x}} \lambda\langle \mathbf{w}^k, \mathbf{g}(\mathbf{x})\rangle + \frac{\mu}{2}\left\|\mathbf{x} - \left(\mathbf{x}^k - \frac{1}{\mu}\nabla h(\mathbf{x}^k)\right)\right\|_2^2,\end{aligned} \tag{5}$$

Proximal Iteratively REweighted (PIRE) algorithm: Alternately updating $\mathbf{w}$ by (3) and $\mathbf{x}$ by (5) to solve (1).

Convergence Analysis of PIRE

Theorem *The sequence $\{\mathbf{x}^k\}$ generated by PIRE satisfies the following properties:*

- *The objective function value $F(\mathbf{x}^k)$ is monotonically decreasing. Indeed,* $F(\mathbf{x}^k) - F(\mathbf{x}^{k+1}) \ge \left(\mu - \frac{L(h)}{2}\right)||\mathbf{x}^k - \mathbf{x}^{k+1}||^2 \ge 0$, *when* $\mu > \frac{L(h)}{2}$;
- $\lim_{k\to\infty}(\mathbf{x}^k - \mathbf{x}^{k+1}) = \mathbf{0}$;
- *The sequence $\{\mathbf{x}^k\}$ is bounded;*
- *Any accumulation point $\mathbf{x}^*$ of $\{\mathbf{x}^k\}$ is a stationary point to problem (1);*

Proximal Iteratively Reweighted Algorithm with Multiple Splitting

We further consider problem (1) with multi-variables

$$\min_{\mathbf{x}_1,\cdots,\mathbf{x}_S} F(\mathbf{x}) = \lambda\sum_{s=1}^S f_s(\mathbf{g}_s(\mathbf{x}_s)) + h(\mathbf{x}_1,\cdots,\mathbf{x}_S), \tag{6}$$

where f_s and $\mathbf{g}_s$ holds the same assumptions as f and $\mathbf{g}$ in problem (1), respectively.
Based on different assumptions of $h(\mathbf{x}_1,\cdots,\mathbf{x}_S)$, we have two splitting versions of the PIRE algorithm.

Proximal Iteratively Reweighted Algorithm with Parallel Splitting

Assume that $\nabla h(\mathbf{x}_1,\cdots,\mathbf{x}_S)$ is Lipschitz continuous (condition (**C3**) holds).
PIRE is naturally parallelizable. Both $\mathbf{x}_s$ and $\mathbf{w}_s$ are updated in parallel by

$$\mathbf{x}_s^{k+1} = \arg\min_{\mathbf{x}_s} \lambda\langle \mathbf{w}_s^k, \mathbf{g}_s(\mathbf{x}_s)\rangle + \frac{\mu}{2}\left\|\mathbf{x}_s - \left(\mathbf{x}_s^k - \frac{1}{\mu}\nabla_s h\left(\mathbf{x}_1^k,\cdots,\mathbf{x}_S^k\right)\right)\right\|^2, \tag{7}$$

where $\mu > L(h)/2$, and

$$\mathbf{w}_s^k \in -\partial\left(-f_s(\mathbf{g}_s(\mathbf{x}_s^k))\right), \ s = 1,\cdots,S. \tag{8}$$

If $h(\mathbf{x}_1,\cdots,\mathbf{x}_S) = \frac{1}{2}\left\|\sum_{s=1}^S \mathbf{A}_s\mathbf{x}_s - \mathbf{b}\right\|_2^2$, we can update $\mathbf{x}_s^{k+1}$ by

$$\mathbf{x}_s^{k+1} = \arg\min_{\mathbf{x}_s} \lambda\langle \mathbf{w}_s^k, \mathbf{g}_s(\mathbf{x}_s)\rangle + \frac{\mu_s}{2}\left\|\mathbf{x}_s - \left(\mathbf{x}_s^k - \frac{1}{\mu_s}\mathbf{A}_s^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})\right)\right\|_2^2, \tag{9}$$

where $\mu_s > L_s(h)/2$ and $L_s(h) = ||\mathbf{A}_s||_2^2$ is the Lipschitz constant of $\nabla_s h(\mathbf{x}_1,\cdots,\mathbf{x}_S)$.
Updating $\mathbf{x}_s^{k+1}$ by (9) is faster than (7) since the required μ_s is usually much smaller than μ.
Proximal Iteratively REweighted algorithm with Parallel Splitting (PIRE-PS): updating $\mathbf{w}_s$ by (8) and $\mathbf{x}_s$ by (7) or (9) in parallel to solve (6).

Proximal Iteratively Reweighted Algorithm with Alternative Updating

Assume that $\nabla_s h(\mathbf{x}_1,\cdots,\mathbf{x}_S)$, $s = 1,\cdots,S$, are Lipschitz continuous.
Then $\mathbf{x}_s$, $s = 1,\cdots,S$, are updated in the alternative way

$$\mathbf{x}_s^{k+1} = \arg\min_{\mathbf{x}_s} \lambda\langle \mathbf{w}_s^k, \mathbf{g}_s(\mathbf{x}_s)\rangle + \frac{\mu_s}{2}\left\|\mathbf{x}_s - \left(\mathbf{x}_s^k - \frac{1}{\mu_s}\nabla_s h(\mathbf{x}_1^{k+1},\cdots,\mathbf{x}_{s-1}^{k+1},\mathbf{x}_s^k,\cdots,\mathbf{x}_S^k)\right)\right\|^2, \tag{10}$$

where $\mu_s > L_s(h)/2$ and $\mathbf{w}_s^k$ is defined in (8).
Proximal Iteratively REweighted algorithm with Alternative Updating (PIRE-AU): updating $\mathbf{w}_s$ by (8) and $\mathbf{x}_s$ by (10) in the alternative way to solve (6).

Some Remarks

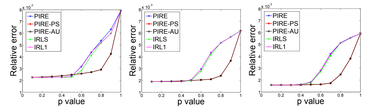
- Similar convergence results of PIRE hold for PIRE-PS and PIRE-AU, i.e., the objectionve function value is monotonically decreasing, and any accumulation point is a stationary point.
- PIRE-PS can be implemented in parallel, while PIRE-AU cannot.
- PIRE-AU may converge faster than PIRE-PS due to smaller Lipschitz constants.
- If the squared loss function is used, PIRE-PS uses the same small Lipschitz constants as PIRE-AU.

Experiment: ℓ_p-Minimization

Test on the ℓ_p-minimization problem

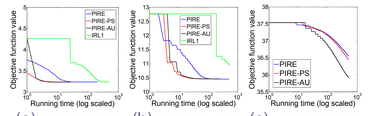
$$\min_{\mathbf{x}\in\mathbb{R}^{n\times t}} \lambda||\mathbf{X}||_p^p + \frac{1}{2}||\mathbf{A}\mathbf{X} - \mathbf{B}||_F^2,$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{B} \in \mathbb{R}^{m\times t}$, and $||\mathbf{X}||_p^p = \sum_{ij}|X_{ij}|^p$.
Compared methods: Iteratively Reweighted L1 (IRL1) and Iteratively Reweighted Least Squares (IRLS).



(a) $m = 200$ (b) $m = 300$ (c) $m = 400$

Figure: Recovery error v.s. different p values with different size of $\mathbf{A} \in \mathbb{R}^{m\times 1000}$.



(a) (1000,3000,500) (b) (1000,5000,1000) (c) (1000,10000,1000)

Figure: Running time v.s. objective function value on three synthesis data sets with different sizes of $\mathbf{A}$ and $\mathbf{B}$: (m, n, t).

Table: Comparison results for the ℓ_p minimization.

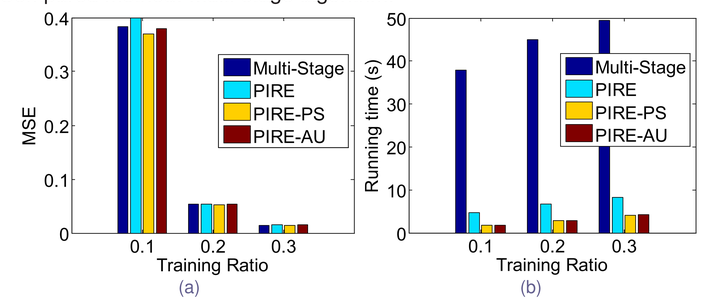
Size (m,n,t)	Methods	Iter.	Time (second).	Obj. ($\times 10^{-2}$)	Recovery error ($\times 10^{-3}$)
(100,500,50)	PIRE	116	0.70	**5.238**	2.529
	PIRE-PS	58	**0.48**	5.239	2.632
	PIRE-AU	**56**	0.63	5.239	2.632
	IRLS	168	81.82	5.506	**2.393**
	IRL1	**56**	3.43	5.239	2.546
(200,800,100)	PIRE	119	1.48	16.923	2.246
	PIRE-PS	37	**0.82**	**16.919**	2.192
	PIRE-AU	**36**	0.88	**16.919**	2.192
	IRLS	169	474.19	17.784	**2.142**
	IRL1	81	13.53	16.924	2.248
(300,1000,200)	PIRE	151	4.63	42.840	2.118
	PIRE-PS	29	1.38	**42.815**	1.978
	PIRE-AU	**28**	**1.34**	**42.815**	**1.977**
	IRLS	171	1298.70	44.937	2.015
	IRL1	79	35.59	42.844	2.124
(500,1500,200)	PIRE	159	8.88	64.769	2.010
	PIRE-PS	26	2.27	**64.718**	**1.814**
	PIRE-AU	**25**	**2.20**	**64.718**	**1.814**
	IRLS	171	3451.79	67.996	1.923
	IRL1	89	80.89	64.772	2.013
(800,2000,200)	PIRE	140	14.99	87.616	1.894
	PIRE-PS	33	5.15	**87.533**	**1.648**
	PIRE-AU	**32**	**4.97**	**87.533**	**1.648**
	IRLS	177	7211.2	91.251	1.851
	IRL1	112	173.26	87.617	1.895

Experiment: Multi-Task Feature Learning

Given m learning tasks $\{(\mathbf{X}_1,\mathbf{y}_1),\cdots,(\mathbf{X}_m,\mathbf{y}_m)\}$, where $\mathbf{X}_i \in \mathbb{R}^{n_i\times d}$ is the data matrix of the i-th task with each row a sample, $\mathbf{y}_i \in \mathbb{R}^{n_i}$ is the label of the i-th task, n_i is the number of samples for the i-th task, and d is the dimension.
Our goal is to find a matrix $\mathbf{Z} = [\mathbf{z}_1,\cdots,\mathbf{z}_m] \in \mathbb{R}^{d\times m}$ such that $\mathbf{y}_i \approx \mathbf{X}_i\mathbf{z}_i$.
Test on the capped-ℓ_1 norm minimization

$$\min_{\mathbf{Z}} \lambda\sum_{j=1}^d \min(||\mathbf{z}^j||_1, \theta) + h(\mathbf{Z}),$$

where $h(\mathbf{Z}) = \sum_{i=1}^m ||\mathbf{X}_i\mathbf{z}_i - \mathbf{y}_i||_2^2/mn_i$, $\theta > 0$, and $\mathbf{z}^j$ is the j-th row of $\mathbf{Z}$.
Compared method: Multi-stage algorithm.



(a) (b)

Figure: Comparison of (a) mean squared error (MSE) and (b) running time on the Isolet data set for multi-task feature learning.

Department of Electrical and Computer Engineering, National University of Singapore. Email: canyilu@nus.edu.sg

Matlab code available at our homepage: https://sites.google.com/site/canyilu/