Iteratively Reweighted Least Squares by Diagonal Regularization

Session 11: Machine Learning and Intelligent Systems
Paper-ID: 1570900706

Bamrung Tausiesakul[1] and Krissada Asavaskulkiet[2]

[1]Department of Electrical Engineering
Faculty of Engineering
Srinakharinwirot University
Nakhon Nayok, Thailand

[2]Department of Electrical Engineering
Faculty of Engineering
Mahidol University
Nakhon Pathom, Thailand

Thursday 29th June, 2023

20th International Joint Conference on Computer Science and Software Engineering (JCSSE 2023), Phitsanulok, Thailand

1. A linear system with noisy output
2. Weighted least squares
3. Iteratively reweighted least squares (IRLS)
4. IRLS with diagonal regularization
5. Numerical examples
6. Conclusion

- Several operations in science and engineering need to get back
  - a desired signal $\boldsymbol{x} \in \mathbb{R}^{N \times 1}$

$$\boldsymbol{x} = \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{1} \left.\vphantom{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}\right\} N \tag{1}$$

  - from a set of observed data or measured data $\boldsymbol{b} \in \mathbb{R}^{M \times 1}$

$$\boldsymbol{b} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}}_{1} \left.\vphantom{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}}\right\} M \tag{2}$$

- based on a modeling matrix or measurement matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$,

$$\boldsymbol{A} = \underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N} \end{bmatrix}}_{N} \Bigg\} M \quad (3)$$

  which either
  - depends on the model or
  - can be chosen beforehand,

where

- $M \in \mathbb{N}^{1 \times 1}$ and
- $N \in \mathbb{N}^{1 \times 1}$

are the lengths of

- real-valued output data $\boldsymbol{b}$ and
- real-valued input data $\boldsymbol{x}$,

respectively.

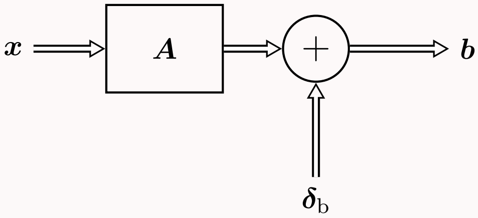

Figure 1: A linear system with noisy output.

- In Fig. 1, the linear system can be written as

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\delta}_\mathrm{b} = \boldsymbol{b}, \tag{4}$$

or, if the perturbation $\boldsymbol{\delta}_\mathrm{b}$ is negligible, approximately as

$$\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}. \tag{5}$$

- A weighted least squares (WLS) optimization problem can be formulated as

$$\hat{\boldsymbol{x}}_{\mathsf{WLS}} = \arg\min_{\boldsymbol{x}} \sum_{n=1}^{N} w_n x_n^2 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{6}$$

- The Lagrangian function of (6) can be written as

$$f(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{n=1}^{N} w_n x_n^2 + \boldsymbol{\lambda}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}), \tag{7}$$

where

  - $\boldsymbol{\lambda} \in \mathbb{R}^{N \times 1}$ is the Lagrange multiplier vector, which is also unknown.

- The closed-form solution of (6) by using the Lagrange multiplier method is given by

$$\hat{\boldsymbol{x}}_{\mathsf{WLS}} = \boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{b}, \tag{8}$$

where

- $\boldsymbol{w} \in \mathbb{R}^{N\times 1}$ is the weighting vector, denoted by

$$\boldsymbol{w} = \underbrace{\left.\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}\right\} N}_{1}, \tag{9}$$

and

- $(\cdot)^{-1}$ is the inverse of a square matrix $\cdot$.

- An iterative computation was introduced earlier as follows.

Algorithm 1 Iteratively Reweighted Least Squares (IRLS) [1, 2]

Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $p \in (0,1]$, $K \in \mathbb{Z}_{+}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_p \in \mathbb{R}^{N\times 1}$

$\hat{\boldsymbol{x}}[0] \leftarrow \mathbf{1}$
$\hat{\boldsymbol{w}}[0] \leftarrow \mathbf{1}$
$\hat{\epsilon}_r[0] \leftarrow 1$
$i \leftarrow 0$
while $\hat{\epsilon}_r[i] \neq 0$ **do**
  $i \leftarrow i + 1$
  $\hat{\epsilon}_r[i] \leftarrow \min\left(\hat{\epsilon}_r[i-1], \frac{1}{N} L_{K+1}(\hat{\boldsymbol{x}}[i-1])\right)$
  $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{D}\left(\hat{\boldsymbol{w}}[i-1]\right)\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{D}\left(\hat{\boldsymbol{w}}[i-1]\right)\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{b}$
  $\hat{\boldsymbol{w}}[i] \leftarrow \left(\hat{\boldsymbol{x}}^2[i] + \hat{\epsilon}_r^2[i]\mathbf{1}\right)^{1-\frac{1}{2}p}$
end while
return $\hat{\boldsymbol{x}}[i]$

- Next we let the weight w_n be $|\hat{x}_n[i]|^{p-2}$, i.e.

$$w_n[i] = |\hat{x}_n[i]|^{p-2}, \tag{10}$$

where

  - i is the index of iteration, p is the exponent of an ℓ_p norm and
  - $\hat{x}_n[i]$ is the n-th element of $\hat{\boldsymbol{x}}[i] \in \mathbb{R}^{N\times 1}$, i.e.

$$\hat{\boldsymbol{x}}[i] = \begin{bmatrix} \hat{x}_1[i] & \hat{x}_2[i] & \cdots & \hat{x}_N[i] \end{bmatrix}^{\mathsf{T}}. \tag{11}$$

- An alternative optimization for the i-th iteration can be written in the form of

$$\hat{\boldsymbol{x}}[i] = \arg\min_{\boldsymbol{u}} \sum_{n=1}^{N} |\hat{x}_n[i-1]|^{p-2} u_n^2 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}. \tag{12}$$

- Following from the same way as (8), one would arrive at

$$\hat{\boldsymbol{x}}[i] = \boldsymbol{D}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{D}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{b}, \quad (13)$$

where

- the definite reciprocal operator $(\cdot)^{\mp}$ is defined as

$$z^{\mp} = \begin{cases} \frac{1}{z}, & z \neq 0, \\ 0, & z = 0, \end{cases} \quad (14)$$

  - for $z \in \mathbb{C}^{1\times 1}$.

- We propose an algorithm based on (13) as follows.

Algorithm 2 IRLS with diagonal regularization

Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $p \in (0,1]$, $N_{\max} \in \mathbb{Z}_{+}^{1\times 1}$, $\epsilon_{\min} \in \mathbb{R}_{+}^{1\times 1}$

Output: $\hat{\boldsymbol{x}}_p \in \mathbb{R}^{N\times 1}$

$\hat{\boldsymbol{x}}[0] \leftarrow \mathbf{1}$

$i \leftarrow 0$

$\epsilon_{\hat{x}} \leftarrow \epsilon_{\min} + 1$

while $\epsilon_{\hat{x}} > \epsilon_{\min} \wedge i \leq N_{\max}$ **do**

$i \leftarrow i + 1$

$\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{D}^{\mp}(|\boldsymbol{x}|^{p-2})\boldsymbol{A}^{\mathsf{T}} \left(\boldsymbol{A}\boldsymbol{D}^{\mp}(|\boldsymbol{x}|^{p-2})\boldsymbol{A}^{\mathsf{T}}\right)^{-1} \boldsymbol{b}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[i-1]}$

$\epsilon_{\hat{x}} \leftarrow \frac{\|\hat{\boldsymbol{x}}[i]-\hat{\boldsymbol{x}}[i-1]\|_2}{\|\hat{\boldsymbol{x}}[i-1]\|_2}$

end while

return $\hat{\boldsymbol{x}}[i]$

- Since $\hat{x}_n[i]$ has a high probability to be nonzero during the iteration, the definite inverse for the matrix $\boldsymbol{D}(|\boldsymbol{x}|^{p-2})$ can be replaced by

$$\begin{aligned}\boldsymbol{D}^{\mp}(|\boldsymbol{x}|^{p-2}) &= \boldsymbol{D}^{-1}(|\boldsymbol{x}|^{p-2}) \\ &= \boldsymbol{D}(|\boldsymbol{x}|^{2-p}).\end{aligned} \tag{15}$$

- Since the matrix $\boldsymbol{A}\boldsymbol{D}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathsf{T}}$ can be ill-conditioned, we approximate it by

$$\begin{aligned}&\boldsymbol{A}\boldsymbol{D}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathsf{T}} \\ &\approx \lim_{\varepsilon\to 0}\left(\boldsymbol{A}\boldsymbol{D}^{-1}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathsf{T}} + \varepsilon\boldsymbol{I}\right)^{-1},\end{aligned} \tag{16}$$

where

  - $\varepsilon \in \mathbb{R}_{+}^{1\times 1}$ is a small positive constant close to zero.

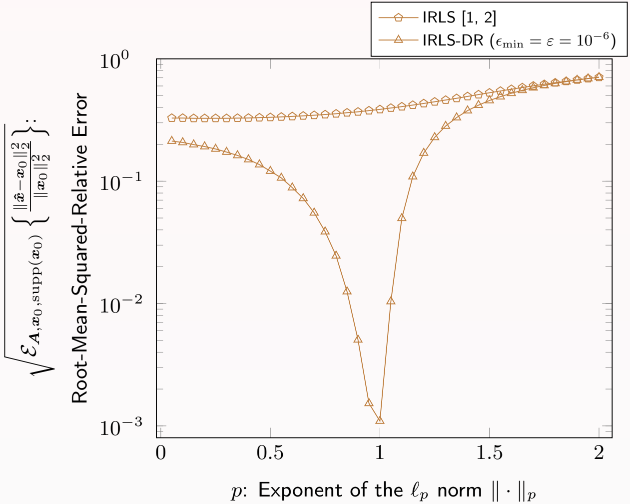

Figure 2: RMSRE as a function of p for $K = 32$, $M = 128$, and $N = 256$ from $N_{\mathrm{R}} = 100{,}000$ independent runs.

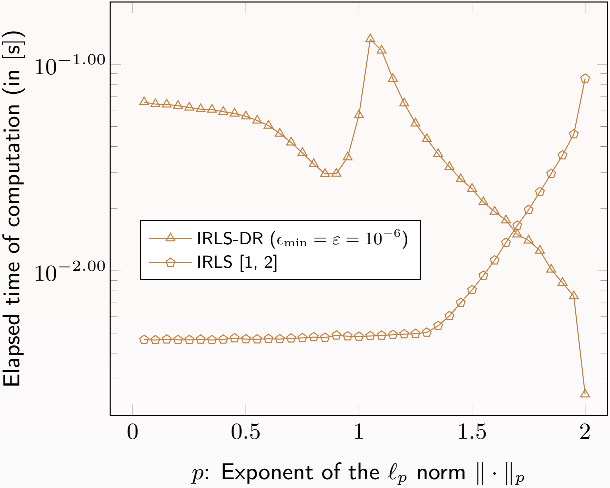

Figure 3: Elapsed time of computation as a function of p for $K = 32$, $M = 128$, and $N = 256$ from $N_{\mathrm{R}} = 100{,}000$ independent runs.

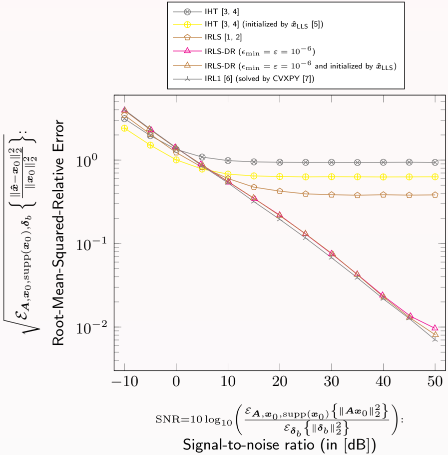

Figure 4: RMSRE as a function of SNR for $K = 32$, $M = 128$, $N = 256$, $p = 0.9$, and $\epsilon_{\min} = \varepsilon = 10^{-6}$ from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = 32$, $N_{\text{supp}(\boldsymbol{x}_0)} = 32$, $N_{\boldsymbol{\delta}_b} = 100$, and $N_{\text{R}} = N_{\boldsymbol{A}} N_{\text{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b} = 102{,}400$ independent runs.

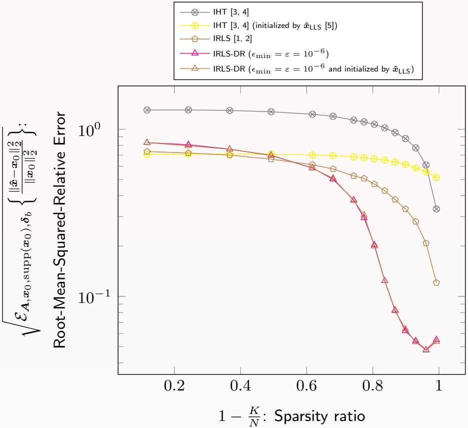

Figure 5: RMSRE as a function of sparsity ratio for $M = 128$, $N = 256$, $p = 0.9$, $\mathrm{SNR} \approx 30$[dB], and $\epsilon_{\min} = \varepsilon = 10^{-6}$ from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = \left\lceil \frac{10^3}{K} \right\rceil$, $N_{\mathrm{supp}(\boldsymbol{x}_0)} = K$, $N_{\boldsymbol{\delta}_b} = 100$, and $N_{\mathrm{R}} = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b}$ independent runs.

- We derive a closed-form solution of the IRLS optimization.
- We resolve an ill condition of a required matrix inverse by adding the diagonal regularization.
- Numerical results illustrate that
  - the error given by the new proposed IRLS method is obviously lower than that by the conventional IRLS algorithm.

[1] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. 2008 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2008)*, Las Vegas, NV, Mar./Apr. 2008, pp. 3869–3872.

[2] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Comm. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.

[3] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 629–654, Dec. 2008.

[4] ——, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[5] B. Tausiesakul, "Iterative hard thresholding using least squares initialization," in *Proc. 2022 IEEE Int. Conf. Cummun. Syst. Netw. Technol. (CSNT 2022)*, Indore, India, Apr. 2022, pp. 612–615.

[6] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.

[7] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 2909–2913, Jan. 2016.

Thank you for your attention