Contents

1 Wirtinger Flow

The whole thing about *Wirtinger Flow* variants started with the seminal work of Candes and Soltanolkotabi[1]. The most important improvements chronologically were done by Candes and Chen[2], Kolte and Özgür[3], and Zhang et al.[4]. For a quite extensive survey on *Wirtinger Flow* variants please refer to Liu et al.[5]. Chandra et al.[6] gathered quite number of *Phase Retrieval* methods including a couple of *Wirtinger Flow* variants in the MATLAB® problem solving environment in a uniform manner.
We quickly go over the problem formulation, difficulties, algorithms, and at the of the chapter we give some numerical experiments we are going to refer to in the subsequent chapters.

1.1 Problem Formulation

Consider the ray $\boldsymbol{x} \in \mathbb{C}^{n\times 1}$ is emitted onto the object of interest and the diffracted rays are measured as $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ and is connected to the original ray by $\boldsymbol{y} = \varphi(\boldsymbol{A}\boldsymbol{x})$, where $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ and φ the usual element-wise absolute value(or the squared absolute value) from $\mathbb{C}^{m\times 1}$ to $\mathbb{R}^{m\times 1}$.
Candes and Soltanolkotabi[1] considered φ to be squared element-wise absolute value and the loss function to be quadratic. The summary for all the variants in terms of formulation is in table1.1

1.2 Difficulties

The loss function is non-convex. Set $n = 1$, $m = 2$, $\boldsymbol{x}_1 = \begin{pmatrix}1+i\end{pmatrix}^{1\times 1}$, $\boldsymbol{x}_2 = \begin{pmatrix}-1-i\end{pmatrix}^{1\times 1}$, $\boldsymbol{A} = \begin{pmatrix}1\\ i\end{pmatrix}^{2\times 1}$, $\boldsymbol{y} = \begin{pmatrix}1\\ 2\end{pmatrix}^{2\times 1}$, and $\lambda = 1/2$ to build a counterexample. Non-convexity is bad news for optimization as it can be seen vividly in [7] and [8]. To make the matter worse the loss function is not holomorphic(it can be easily seen that Cauchy-Riemann equations[9] do not hold) and therefore complex differentiability is out of the question[9].

Wirtinger Flow Variant	φ	loss functions
Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Truncated Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Incrementally Truncated Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Reshaped Wirtinger Flow	$\lvert\boldsymbol{z}\rvert$	quadratic
Incrementally Reshaped Flow	$\lvert\boldsymbol{z}\rvert$	quadratic

Table 1.1: φ and the loss function used in [1], [2], [3], [4]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Parameters: Lower and upper thresholds α_l, α_u for truncation in initialization, step size μ;
Initialization: Let $\boldsymbol{z}^{(0)} = \lambda_0 \tilde{\boldsymbol{z}}$, where $\lambda_0 = \frac{mn}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1} \cdot \left(\frac{1}{m}\sum_{i=1}^m y_i\right)$ and $\tilde{\boldsymbol{z}}$ is the leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{i=1}^m y_i \boldsymbol{a}_i \boldsymbol{a}_i^* \mathbf{1}_{\{\alpha_l \lambda_0 < y_i < \alpha_u \lambda_0\}}. \tag{1.1}$$

Update loop: for $t = 0 : T-1$ do

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \frac{\mu}{m}\sum_{i=1}^m \left(\boldsymbol{a}_i^* \boldsymbol{z}^{(t)} - y_i \cdot \frac{\boldsymbol{a}_i^* \boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^* \boldsymbol{z}^{(t)}|} \right) \boldsymbol{a}_i. \tag{1.2}$$

Output $\boldsymbol{z}^{(T)}$.

Algorithm 1: Reshaped *Wirtinger Flow* suggested by [4]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Initialization: Same as in RWF (Algorithm 1);
Parameters: Lower and upper thresholds α_l, α_u for truncation in initialization, step size μ;
Update loop: for $t = 0 : T-1$ do
Choose i_t uniformly at random from $\{1, 2, \ldots, m\}$, and let

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \left(\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)} - y_{i_t} \cdot \frac{\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)}|} \right) \boldsymbol{a}_{i_t}, \tag{1.3}$$

Output $\boldsymbol{z}^{(T)}$.

Algorithm 2: Incremental Reshaped *Wirtinger Flow* (IRWF) suggested by [4]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Initialization: Same as in RWF (Algorithm 1);
Update loop: for $t = 0 : T-1$ do
Choose Γ_t uniformly at random from the subsets of $\{1, 2, \ldots, m\}$ with cardinality k, and let

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \cdot \boldsymbol{A}_{\Gamma_t}^* \left(\boldsymbol{A}_{\Gamma_t} \boldsymbol{z}^{(t)} - \boldsymbol{y}_{\Gamma_t} \odot \mathrm{Ph}(\boldsymbol{A}_{\Gamma_t} \boldsymbol{z}^{(t)}) \right), \tag{1.4}$$

where $\boldsymbol{A}_{\Gamma_t}$ is a matrix stacking $\boldsymbol{a}_i^*$ for $i \in \Gamma_t$ as its rows, $\boldsymbol{y}_{\Gamma_t}$ is a vector stacking y_i for $i \in \Gamma_t$ as its elements, $\odot$ denotes element-wise product, and $\mathrm{Ph}(\boldsymbol{z})$ denotes a phase vector of $\boldsymbol{z}$.
Output $\boldsymbol{z}^{(T)}$.

Algorithm 3: Minibatch Incremetnal Reshaped *Wirtinger Flow* (minibatch IRWF) suggested by [4]

Bibliography

[1] E. J. Candes, X. Li, and M. Soltanolkotabi. *Phase Retrieval via Wirtinger Flow: Theory and Algorithms*. Apr. 2015.

[2] Y. Chen and E. J. Candes. *Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems*. 2016. arXiv: `1505.05114 [cs.IT]`.

[3] R. Kolte and A. Özgür. *Phase Retrieval via Incremental Truncated Wirtinger Flow*. 2016. arXiv: `1606.03196 [cs.IT]`.

[4] H. Zhang, Y. Zhou, Y. Liang, and Y. Chi. *Reshaped Wirtinger Flow and Incremental Algorithm for Solving Quadratic System of Equations*. 2016. arXiv: `1605.07719 [stat.ML]`.

[5] J.-W. Liu, Z.-J. Cao, J. Liu, X.-L. Luo, W.-M. Li, N. Ito, and L.-C. Guo. "Phase Retrieval via Wirtinger Flow Algorithm and Its Variants." In: *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*. 2019, pp. 1–9.

[6] R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, and T. Goldstein. *PhasePack: A Phase Retrieval Library*. 2017. arXiv: `1711.10175 [math.OC]`.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd ed. Springer New York, 2006.

[9] L. V. Ahlfors. *Complex Analysis. An introduction to the theory of analytic functions of one complex variable*. 2nd ed. McGraw-Hill Book Company, 1966.