Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems

Yuxin Chen (Princeton)  Emmanuel Candès (Stanford)

nonconvex optimization  (high-dimensional) statistics
Solving quadratic systems of equations

\[
A \quad x = \quad Ax \quad y = |Ax|^2
\]

Solve for \( x \in \mathbb{C}^n \) in \( m \) quadratic equations

\[
y_k \approx |\langle a_k, x \rangle|^2, \quad k = 1, \ldots, m
\]
Motivation: a missing phase problem in imaging science

Detectors record intensities of diffracted rays

- $x(t_1, t_2) \rightarrow$ Fourier transform $\hat{x}(f_1, f_2)$

intensity of electrical field: $|\hat{x}(f_1, f_2)|^2 = \left| \int x(t_1, t_2)e^{-i2\pi(f_1 t_1 + f_2 t_2)} dt_1 dt_2 \right|^2$

Phase retrieval: recover true signal $x(t_1, t_2)$ from intensity measurements
Motivation: learning neural nets with quadratic activation

\[ y = \sum_{i=1}^{r} \sigma(\mathbf{a}^\top \mathbf{x}_i)^2 = \sum_{i=1}^{r} (\mathbf{a}^\top \mathbf{x}_i)^2 \]

input features: \( \mathbf{a} \); weights: \( \mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_r] \)

\[ \sigma(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases} \]
Solving quadratic systems is NP-complete in general ...

“I can’t find an efficient algorithm, but neither can all these people.”

Fig credit: coding horror
Statistical models come to rescue

When data are generated by certain statistical / randomized models, problems are often much nicer than worst-case instances

e.g. \( \mathbf{a}_k \sim \mathcal{N}(0, I_n) \)
Convex relaxation

Lifting: introduce $X = xx^*$ to linearize constraints

\[ y_k = |a_k^* x|^2 = a_k^* (xx^*) a_k \implies y_k = a_k^* X a_k \]
Convex relaxation

Lifting: introduce $X = xx^*$ to linearize constraints

$$y_k = |a_k^* x|^2 = a_k^* (xx^*) a_k \quad \Rightarrow \quad y_k = a_k^* X a_k$$

find $X \succeq 0$

s.t. $y_k = a_k^* X a_k, \quad k = 1, \cdots, m$

$\text{rank}(X) = 1$
Convex relaxation

Lifting: introduce $X = xx^*$ to linearize constraints

$$y_k = |a_k^* x|^2 = a_k^* (xx^*) a_k \quad \Rightarrow \quad y_k = a_k^* X a_k$$

find $X \succeq 0$

s.t. $y_k = a_k^* X a_k, \quad k = 1, \ldots, m$

$\text{rank}(X) = 1$
Convex relaxation

Lifting: introduce $X = xx^*$ to linearize constraints

$$y_k = |a_k^* x|^2 = a_k^* (xx^*) a_k \implies y_k = a_k^* X a_k$$

find $X \succeq 0$

s.t. $y_k = a_k^* X a_k$, $k = 1, \cdots, m$

$\text{rank}(X) = 1$

Works well if $\{a_k\}$ are random
Convex relaxation

Lifting: introduce $X = xx^*$ to linearize constraints

$$y_k = |a_k^* x|^2 = a_k^*(xx^*)a_k \quad \implies \quad y_k = a_k^*Xa_k$$

find $X \succeq 0$

s.t. $y_k = a_k^*Xa_k, \quad k = 1, \ldots, m$

$\text{rank}(X) = 1$

Works well if $\{a_k\}$ are random, but huge increase in dimensions
Prior art (before our work)

\[ y = |Ax|^2, \ A \in \mathbb{R}^{m \times n} \]

- \( n \): \# unknowns
- \( m \): sample size (\# eqns)

Sample complexity:
- Infeasible
- \( n \): \# unknowns
- \text{Comput. cost}

This work: random quadratic systems are solvable in linear time!
Prior art (before our work)

\[ n: \# \text{unknowns}; \quad m: \text{sample size (\# eqns)}; \quad y = |Ax|^2, A \in \mathbb{R}^{m \times n} \]

sample complexity

comput. cost

infeasible

infeasible

This work: random quadratic systems are solvable in linear time!

✓ minimal sample size

✓ optimal statistical accuracy
Prior art (before our work)

\[ n: \text{# unknowns}; \quad m: \text{sample size (# eqns)}; \quad y = |Ax|^2, A \in \mathbb{R}^{m \times n} \]

This work: random quadratic systems are solvable in linear time!
Prior art (before our work)

\[ y = |Ax|^2, \ A \in \mathbb{R}^{m \times n} \]

\( n \): # unknowns; \( m \): sample size (# eqns);

\begin{align*}
\text{sample complexity} & \quad \text{comput. cost} \\
\text{infeasible} & \quad mn \log n \\
\text{infeasible} & \quad mn^2 \log n \\
\text{Wirtinger flow} & \quad \text{cvx relaxation}
\end{align*}

This work: random quadratic systems are solvable in linear time! ✓

✓ minimal sample size

✓ optimal statistical accuracy
Prior art (before our work)

\[ y = |Ax|^2, \quad A \in \mathbb{R}^{m \times n} \]

- \( n \): \# unknowns;
- \( m \): sample size (# eqns);

Sample complexity

- cvx relaxation
- Wirtinger flow
- alt-min (fresh samples at each iter)
- infeasible

Comput. cost

- \( mn \)
- \( mn^2 \)

This work: random quadratic systems are solvable in linear time!
A glimpse of our results

\[ y = |Ax|^2, \ A \in \mathbb{R}^{m \times n} \]

\[ n: \# \ unknowns; \quad m: \ sample \ size \ (\# \ eqns); \]

This work: random quadratic systems are solvable in \textit{linear time!}
A glimpse of our results

\[ y = |Ax|^2, \ A \in \mathbb{R}^{m \times n} \]

- \( n \): # unknowns;
- \( m \): sample size (# eqns);

This work: random quadratic systems are solvable in \textit{linear time}!

\( \checkmark \) minimal sample size

\( \checkmark \) optimal statistical accuracy
A first impulse: maximum likelihood estimate

\[
\text{minimize}_z \quad f(z) = \frac{1}{m} \sum_{k=1}^{m} f_k(z)
\]
A first impulse: maximum likelihood estimate

\[
\text{minimize}_z \quad f(z) = \frac{1}{m} \sum_{k=1}^{m} f_k(z)
\]

- Gaussian data: \( y_k \sim |a_k^* x|^2 + \mathcal{N}(0, \sigma^2) \)
  \[
  f_k(z) = (y_k - |a_k^* z|^2)^2
  \]
A first impulse: maximum likelihood estimate

\[ \text{minimize}_z \quad f(z) = \frac{1}{m} \sum_{k=1}^{m} f_k(z) \]

- Gaussian data: \( y_k \sim |a_k^* x|^2 + \mathcal{N}(0, \sigma^2) \)
  \[ f_k(z) = \left( y_k - |a_k^* z|^2 \right)^2 \]

- Poisson data: \( y_k \sim \text{Poisson}(|a_k^* x|^2) \)
  \[ f_k(z) = |a_k^* z|^2 - y_k \log |a_k^* z|^2 \]
A first impulse: maximum likelihood estimate

\[ \text{minimize}_z \quad f(z) = \frac{1}{m} \sum_{k=1}^{m} f_k(z) \]

- Gaussian data: \( y_k \sim |a_k^* x|^2 + \mathcal{N}(0, \sigma^2) \)
  \[ f_k(z) = (y_k - |a_k^* z|^2)^2 \]

- Poisson data: \( y_k \sim \text{Poisson}( |a_k^* x|^2 ) \)
  \[ f_k(z) = |a_k^* z|^2 - y_k \log |a_k^* z|^2 \]

**Problem:** \( f(\cdot) \) nonconvex, many local stationary points
A plausible nonconvex paradigm

\[
\text{minimize}_z \quad f(z) = \sum_{k=1}^{m} f_k(z)
\]

1. initialize within local basin sufficiently close to \( \mathbf{x} \)

(hopefully) nicer landscape
A plausible nonconvex paradigm

\[
\min_z f(z) = \sum_{k=1}^m f_k(z)
\]

1. Initialize within local basin sufficiently close to \(x\)
   \[\text{(hopefully) nicer landscape}\]

2. Iterative refinement
Wirtinger flow (Candès, Li, Soltanolkotabi ’14)

\[
\minimize_z \quad f(z) = \frac{1}{m} \sum_{k=1}^{m} \left[ (a_k^\top z)^2 - y_k \right]^2
\]

- **spectral initialization:** \( z^0 \leftarrow \) leading eigenvector of certain data matrix

- **(Wirtinger) gradient descent:**
  \[
  z^{t+1} = z^t - \mu_t \nabla f(z^t), \quad t = 0, 1, \ldots
  \]
Performance guarantees for WF

- suboptimal computational cost?
  - \( n \) times more expensive than linear-time algorithms

- suboptimal sample complexity?
Iterative refinement stage: search directions

Wirtinger flow: 
$$z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{k=1}^{m} \left( y_k - |a_k^T z^t|^2 \right) a_k a_k^T z^t$$

$$= \nabla f_k (z^t)$$
Iterative refinement stage: search directions

Wirtinger flow: \[ z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{k=1}^{m} \left( y_k - \left| a_k^\top z^t \right|^2 \right) a_k a_k^\top z^t \]

\[ = \nabla f_k(z^t) \]

Even in a local region around \( x \) (e.g. \( \{ z \mid \| z - x \|_2 \leq 0.1\| x \|_2 \} \)):

- \( f(\cdot) \) is NOT strongly convex unless \( m \gg n \)
- \( f(\cdot) \) has huge smoothness parameter
Iterative refinement stage: search directions

Wirtinger flow:  
\[ z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{k=1}^{m} \left( y_k - |a_k^\top z^t|^2 \right) a_k a_k^\top z^t \]

\[ = \nabla f_k(z^t) \]

Problem: descent direction has large variability
Our solution: variance reduction via proper trimming

More adaptive rule:

\[ z^{t+1} = z^t - \mu_t \frac{1}{m} \sum_{i=1}^{m} y_i - |a_i^T z^t|^2 a_i 1_{E_1(z^t) \cap E_2(z^t)} \]

where \( E_1(z) = \left\{ \alpha^b z \leq |a_i^T z| \leq \alpha^u z \right\} \); \( E_2(z) = \left\{ |y_i - |a_i^T z|^2 | \leq \frac{\alpha_h}{m} \frac{ \| y - A(z z^T) \|_1 |a_i^T z| }{ \| z \|_2 } \right\} \)
Our solution: variance reduction via proper trimming

More adaptive rule:

\[ z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{i=1}^{m} \frac{y_i - |a_i^T z^t|^2}{a_i^T z^t} a_i 1_{E_i^1(z^t) \cap E_i^2(z^t)} \]

where \( E_i^1(z) = \left\{ \alpha_{z}^{lb} \leq \frac{|a_i^T z|}{\|z\|_2} \leq \alpha_{z}^{ub} \right\} \); \( E_i^2(z) = \left\{ |y_i - |a_i^T z|^2| \leq \frac{\alpha_h m \|y - A(z z^T)\|_1 |a_i^T z|}{\|z\|_2} \right\} \)
Our solution: variance reduction via proper trimming

More adaptive rule:

$$z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{i=1}^{m} y_i - |a_i^T z^t|^2 \frac{a_i 1_{E_1(z^t)} \cap E_2(z^t)}{a_i^T z^t}$$

where $E_1(z) = \left\{ \alpha^l_z \leq \frac{|a_i^T z|}{\|z\|_2} \leq \alpha^u_z \right\}$; $E_2(z) = \left\{ |y_i - |a_i^T z|^2| \leq \frac{\alpha_h m \|y - A(z z^T)\|_1 |a_i^T z|}{\|z\|_2} \right\}$

Informally, $z^{t+1} = z^t - \frac{\mu}{m} \sum_{k \in \mathcal{T}} \nabla f_k(z^t)$

- $\mathcal{T}$ trims away excessively large gradient components
Our solution: variance reduction via proper trimming

More adaptive rule:

\[ \mathbf{z}^{t+1} = \mathbf{z}^t - \frac{\mu_t}{m} \sum_{i=1}^{m} \frac{y_i - |a_i^T \mathbf{z}^t|^2}{a_i^T \mathbf{z}^t} a_i \mathbf{1}_{\mathcal{E}_1^i(z^t)} \cap \mathcal{E}_2^i(z^t) \]

where \( \mathcal{E}_1^i(z) = \left\{ \alpha_{z}^b \leq \frac{|a_i^T z|}{\|z\|_2} \leq \alpha_{z}^u \right\} \); \( \mathcal{E}_2^i(z) = \left\{ |y_i - |a_i^T z|^2| \leq \frac{\alpha_{h}}{m} \frac{\|y - A(z z^T)\|_1}{\|z\|_2} |a_i^T z| \right\} \)

informally, \( \mathbf{z}^{t+1} = \mathbf{z}^t - \frac{\mu}{m} \sum_{k \in T} \nabla f_k(\mathbf{z}^t) \)

- \( T \) trims away excessively large grad components

Slight bias \quad + \quad much reduced variance
Larger step size $\mu_t$ is feasible

\[
A \mathbf{x} = |A\mathbf{x}|_2 \cdot |\cdot|_2 \text{ entrywise squared magnitude}
\]

minimize $\mathbf{X}$

s.t. $b_k = a_k \mathbf{x}$, $k = 1, \ldots, m$

$\mathbf{y} = (h \mathbf{x}, i + \mathbf{\pi}, k)$ with prob. $\frac{1}{2}$

initial guess $\mathbf{z}_0$

basin of attraction

With better-controlled descent directions, one proceeds far more aggressively.

without trimming: $\mu_t = O(1/n)$

with trimming: $\mu_t = O(1)$
Initialization stage

Spectral initialization (e.g. alt-min, WF): \( z^0 \leftarrow \) leading eigenvector of

\[
Y := \frac{1}{m} \sum_{k=1}^{m} y_k a_k a_k^* 
\]
Initialization stage

Spectral initialization (e.g. alt-min, WF): $z^0 \leftarrow \text{leading eigenvector of}$

\[
\mathbf{Y} := \frac{1}{m} \sum_{k=1}^{m} y_k \mathbf{a}_k \mathbf{a}_k^* 
\]

• Rationale: $\mathbb{E}[\mathbf{Y}] = \|\mathbf{x}\|_2^2 \mathbf{I} + 2\mathbf{x}\mathbf{x}^*$ under i.i.d. Gaussian design
Initialization stage

Spectral initialization (e.g. alt-min, WF): \( z^0 \leftarrow \) leading eigenvector of

\[
Y := \frac{1}{m} \sum_{k=1}^{m} y_k a_k a_k^* 
\]

- Rationale: \( \mathbb{E}[Y] = \|x\|_2^2 I + 2xx^* \) under i.i.d. Gaussian design

- Would succeed if \( Y \to \mathbb{E}[Y] \)
Improving initialization

\[ Y = \frac{1}{m} \sum_k \left( y_k a_k a_k^* \right) \quad \text{heavy-tailed} \quad \leadsto \quad \mathbb{E}[Y] \quad \text{unless } m \gg n \]
Improving initialization

\[ Y = \frac{1}{m} \sum_{k} y_k a_k a_k^* \quad \rightarrow \quad \mathbb{E}[Y] \quad \text{unless } m \gg n \]

Problem: large outliers

\[ y_k = |a_k^* x| \]

Solution: discard large samples and run PCA for

\[ \frac{1}{\|a_k\|^2} a_k^* Y a_k \]

\[ x^* Y x \]
**Improving initialization**

\[
Y = \frac{1}{m} \sum_{k} y_k a_k a_k^* \quad \rightarrow \quad \mathbb{E}[Y] \quad \text{unless } m \gg n
\]

**Problem**  
Large outliers \( y_k = |a_k^* x|^2 \) bear too much influence
Improving initialization

\[
Y = \frac{1}{m} \sum_k y_k a_k a_k^* \rightarrow \mathbb{E}[Y] \text{ unless } m \gg n
\]

Problem \ large outliers \( y_k = |a_k^* x|^2 \) bear too much influence

Solution \ discard large samples and run PCA for \( \frac{1}{m} \sum_k y_k a_k a_k^* 1\{|y_k| \leq \text{Avg}\{|y_l|\}\} \)
Summary of proposed algorithm

1. **Regularized spectral initialization**: \( z^0 \leftarrow \text{principal component of} \)

\[
\frac{1}{m} \sum_{k \in \mathcal{T}_0} y_k a_k a_k^* \]

2. **Regularized gradient descent**

\[
z^{t+1} = z^t - \frac{\mu_t}{m} \sum_{k \in \mathcal{T}_t} \nabla f_k(z) \]

**Adaptive and iteration-varying rules**: discard high-leverage data \( \{y_k : k \notin \mathcal{T}_t\} \)
Theoretical guarantees (noiseless data)

\[ A x A y = |A x|^2 \]

\[ \text{minimize } \|X \|_{\text{entrywise}}^2 \text{ s.t. } b^k = a^k \sum_{k=1}^m a_k, \cdots, m \]

\[ y_k = (h x_k, i) + \gamma_k, \text{ with prob. } \frac{1}{2} h x_k, i + \gamma_k, \text{ else } y \mapsto h x, i \]

initial guess \( z^0 \)

basin of attraction

**Theorem (Chen & Candès)** When \( a_k \) i.i.d. \( \sim \mathcal{N}(0, I_n) \) and \( m \gtrsim n \), with high probability our algorithm attains \( \varepsilon \) accuracy in \( O \left( \log \frac{1}{\varepsilon} \right) \) iterations

\( \varepsilon \)-accuracy

\[ \text{dimension-free linear convergence} \]
Computational complexity

\[ A := \{a_k^*\}_{1 \leq k \leq m} \]

- **Initialization:** leading eigenvector $\rightarrow$ a few applications of $A$ and $A^*$

\[ \sum_{k \in \mathcal{T}_0} y_k a_k a_k^* = A^* \text{ diag}\{y_k \cdot 1_{k \in \mathcal{T}_0}\} A \]
Computational complexity

\[ A := \{ a_k^* \}_{1 \leq k \leq m} \]

- **Initialization:** leading eigenvector → a few applications of \( A \) and \( A^* \)

\[ \sum_{k \in T_0} y_k a_k a_k^* = A^* \text{ diag}\{y_k \cdot 1_{k \in T_0}\} \ A \]

- **Iterations:** one application of \( A \) and \( A^* \) per iteration

\[
\begin{align*}
    z^{t+1} &= z^t - \frac{\mu_t}{m} \nabla f_{\text{tr}}(z^t) \\
    \nabla f_{\text{tr}}(z^t) &= A^* \nu \\
    \nu &= 2 \left| A z^t \right|^2 - y \cdot 1_T
\end{align*}
\]
Computational complexity

\[ A := \{a_k^*\}_{1 \leq k \leq m} \]

- **Initialization:** leading eigenvector → a few applications of \( A \) and \( A^* \)

\[ \sum_{k \in \mathcal{T}_0} y_k a_k a_k^* = A^* \text{ diag}\{y_k \cdot 1_{k \in \mathcal{T}_0}\} A \]

- **Iterations:** one application of \( A \) and \( A^* \) per iteration

\[ z^{t+1} = z^t - \frac{\mu_t}{m} \nabla f_{tr}(z^t) \quad \text{and} \quad \nabla f_{tr}(z^t) = A^* \nu \]

\[ \nu = 2 \frac{|A z^t|^2 - y}{A z^t} \cdot 1_{\mathcal{T}} \]

**Approximate runtime:** several tens of applications of \( A \) and \( A^* \)
Numerical performance

- CG: solve $y = Ax$
- Our algorithm: solve $y = |Ax|^2$

![Graph showing relative error (log scale) vs. iteration for both CG and proposed algorithm. The proposed algorithm outperforms the least squares (CG) method with a lower relative error across iterations.]
Numerical performance

- CG: solve $y = Ax$
- Our algorithm: solve $y = |Ax|^2$

For random quadratic systems ($m = 8n$)
comput. cost of our algo. $\approx 4 \times$ comput. cost of least squares
Empirical performance \((m = 12n)\)

Ground truth \(x \in \mathbb{R}^{409600}\)
Empirical performance ($m = 12n$)

Spectral initialization
Empirical performance \((m = 12n)\)

Spectral initialization

Proposed: regularized spectral initialization
Empirical performance ($m = 12n$)

After regularized spectral initialization
Empirical performance \( (m = 12n) \)

After regularized spectral initialization

After 50 proposed iterations
Stability under noisy data

Comparison with genie-aided MLE (with phase info. revealed)

\[ y_k \sim \text{Poisson}( |a_k^* x|^2 ) \quad \text{and} \quad \varepsilon_k = \text{sign}(a_k^* x) \quad \text{(revealed by a genie)} \]
Stability under noisy data

Comparison with genie-aided MLE (with phase info. revealed)

\[
y_k \sim \text{Poisson}(|a_k^* x|^2) \quad \text{and} \quad \varepsilon_k = \text{sign}(a_k^* x) \quad \text{(revealed by a genie)}
\]
Stability under noisy data

Comparison with genie-aided MLE (with phase info. revealed)

\[ y_k \sim \text{Poisson}(|a_k^*x|^2) \quad \text{and} \quad \varepsilon_k = \text{sign}(a_k^*x) \quad \text{(revealed by a genie)} \]

Theorem (Chen & Candès) Our algorithm achieves optimal statistical accuracy!
Deal with complicated dependencies across iterations

Several prior approaches: require fresh samples at each iteration
Deal with complicated dependencies across iterations

Several prior approaches: require fresh samples at each iteration

This approach: reuse all samples in all iterations
A small sample of more recent works

- **other optimal algorithms**
  - reshaped WF (Zhang et al.), truncated AF (Wang et al.), median-TWF (Zhang et al.)
  - alt-min w/o resampling (Waldspurger)
  - composite optimization (Duchi et al., Charisopoulos et al.)
  - approximate message passing (Ma et al.)
  - block coordinate descent (Barmherzig et al.)
  - PhaseMax (Goldstein et al., Bahmani et al., Salehi et al., Dhifallah et al., Hand et al.)
- **stochastic algorithms** (Kolte et al., Zhang et al., Lu et al., Tan et al., Jeong et al.)
- **improved WF theory**: iteration complexity $\rightarrow O(\log n \log \frac{1}{\varepsilon})$ (Ma et al.)
- **improved initialization** (Lu et al., Wang et al., Mondelli et al.)
- **random initialization** (Chen et al.)
- **structured quadratic systems** (Cai et al., Soltanolkotabi, Wang et al., Yang et al., Qu et al.)
- **geometric analysis** (Sun et al., Davis et al.)
- **low-rank generalization** (White et al., Li et al., Vaswani et al.)
Concluding remarks

Achieves optimal bias-variance tradeoff by adaptively discarding high-leverage data

<table>
<thead>
<tr>
<th></th>
<th>comput. cost</th>
<th>sample size</th>
<th>statistical accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvx relaxation</td>
<td>🔴</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>our non-cvx algo.</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
</tbody>
</table>
Concluding remarks

Achieves optimal bias-variance tradeoff by adaptively discarding high-leverage data

<table>
<thead>
<tr>
<th></th>
<th>comput. cost</th>
<th>sample size</th>
<th>statistical accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvx relaxation</td>
<td></td>
<td>![Green Thumbs Up]</td>
<td>![Green Thumbs Up]</td>
</tr>
<tr>
<td>our non-cvx algo.</td>
<td>![Red Thumbs Down]</td>
<td>![Green Thumbs Up]</td>
<td>![Green Thumbs Up]</td>
</tr>
</tbody>
</table>

nonconvex optimization  
(high-dimensional) statistics