Sketchy Decisions

Joel A. Tropp

Computing + Mathematical Sciences
California Institute of Technology
jtropp@cms.caltech.edu

Collaborators: Volkan Cevher (EPFL), Roarke Horstmeyer (Duke), Quoc Tran-Dinh (UNC), Madeleine Udell (Cornell), Alp Yurtsever (EPFL)

Research supported by ONR, AFOSR, NSF, DARPA, ERC, SNF, Sloan, and Moore.
Fourier Ptychography
Microscopy: Field of View / Resolution

Source: Adapted from Zhang et al. 2013.
Fourier Ptychography: Field of View + Resolution

Fourier Ptychography: Malaria Example

Fourier Ptychography: Schematic

Source: Adapted from Horstmeyer & Yang 2014.
Fourier Ptychography: Reconstruction

- Acquire a family of noisy measurements:

\[ b_i = |\langle a_i, \psi \rangle|^2 + \xi_i \quad \text{for } i = 1, \ldots, d \]

- \( a_i \in \mathbb{C}^n \) are known measurement vectors that model FP system
- \( \psi \in \mathbb{C}^n \) is the unknown sample transmission function
- \( \xi_i \in \mathbb{R} \) is unknown noise

- Reconstruction via optimization:

\[
\min_{x \in \mathbb{C}^n} \sum_{i=1}^{d} \text{loss}( |\langle a_i, x \rangle|^2; b_i )
\]

- Assume \( \text{loss}(\cdot; b) \) is a convex function

**Malaria example:** \( n = 25, 600 \) and \( d = 185, 600 \)

Fourier Ptychography: Convex Reconstruction

- **Observe:** $|\langle a, x \rangle|^2 = a^* (xx^*) a = a^* X a$ where $X$ is rank-one, psd

- Lift to matrix optimization problem:

  $$\min_{X \in \mathbb{C}^{n \times n}} \sum_{i=1}^{d} \text{loss}(a_i^* X a_i; b_i) \quad \text{subject to} \quad \text{rank}(X) = 1; \quad X \text{ psd}$$

- Replace rank constraint with trace constraint to obtain convex problem:

  $$\min_{X \in \mathbb{C}^{n \times n}} \sum_{i=1}^{d} \text{loss}(a_i^* X a_i; b_i) \quad \text{subject to} \quad \text{trace}(X) = \alpha; \quad X \text{ psd}$$

- Return maximum eigenvector $x_*$ of a solution $X_*$

- **Malaria example:** Matrix $X$ has $n^2 = 6.55 \cdot 10^8$ real dof

Sources: AIM Frames Workshop 2008; Edidin et al. 2009; Chai et al. 2011; Candès et al. 2013; Horstmeyer et al. 2015.
Convexity: Why Bother?

Wirtinger Flow (not convex)  Burer–Monteiro (sort of convex)  ??? (convex)

images of $x$ phase gradient

**Challenge:** How to solve the convex ptychography problem at scale?

**Sources:** Burer & Monteiro 2003; Candès et al. 2014; Horstmeyer et al. 2015; Yurtsever et al. 2017.
Optimization with Optimal Storage
Can we develop algorithms that reliably solve an optimization problem using storage that does not exceed the size of the problem data or the size of the solution?
Convex Low-Rank Matrix Optimization

\[
\minimize_{X \in \mathbb{H}_n} f(A X) \quad \text{subject to} \quad \text{trace}(X) = \alpha; \quad X \text{ psd}
\]

Details:

- \( A : \mathbb{H}_n \to \mathbb{R}^d \) is a real-linear map on \( n \times n \) Hermitian matrices
- \( f : \mathbb{R}^d \to \mathbb{R} \) is convex and differentiable
- In many applications,
  - \( A \) extracts \( d \) linear measurements of \( n \times n \) matrix
  - \( f = \text{loss}(\cdot; b) \) for data \( b \in \mathbb{R}^d \)
  - \( d \ll n^2 \)
  - \( \alpha \) modulates rank of solution
- Models problems in signal processing, statistics, and machine learning
  (e.g., convex ptychography)
Optimal Storage

What kind of storage bounds can we hope for?

- **Assume** black-box implementation of operations with linear map:
  \[
  u \mapsto \mathcal{A}(uu^*) \quad \quad (u, z) \mapsto (\mathcal{A}^* z) u
  \]
  \[
  \mathbb{C}^n \to \mathbb{R}^d \quad \quad \mathbb{C}^n \times \mathbb{R}^d \to \mathbb{C}^n
  \]

- Need \( \Theta(n + d) \) storage for output of black-box operations

- Need \( \Theta(rn) \) storage for rank-\( r \) approximate solution of model problem

**Definition.** An algorithm for the model problem has optimal storage if its working storage is \( \Theta(d + rn) \) rather than \( \Theta(n^2) \).

**Source:** Yurtsever et al. 2017; Cevher et al. 2017.
So Many Algorithms...

1990s: **Interior-point methods**
- Storage cost $\Theta(n^4)$ for Hessian

2000s: **Convex first-order methods**
- (Accelerated) proximal gradient, spectral bundle methods, and others
- Store matrix variable $\Theta(n^2)$

2008–Present: **Storage-efficient convex first-order methods**
- Conditional gradient method (CGM) and extensions
- Store matrix in low-rank form $O(tn)$; no storage guarantees

2009–Present: **Nonconvex heuristics**
- Burer–Monteiro factorization idea + various nonlinear programming methods
- Store low-rank matrix factors $\Theta( rn)$
- For guaranteed solution, need unrealistic + unverifiable statistical assumptions

Sources: **Interior-point**: Nemirovski & Nesterov 1994; ... **First-order**: Rockafellar 1976; Helmberg & Rendl 1997; Auslender & Teboulle 2006; ... **CGM**: Frank & Wolfe 1956; Levitin & Poljak 1967; Jaggi 2013; ... **Heuristics**: Burer & Monteiro 2003; Keshavan et al. 2009; Jain et al. 2012; Candès et al. 2014; Bhojanapalli et al. 2015; Boumal et al. 2016; ....
The Challenge

- Some algorithms provably solve the model problem...
- Some algorithms have optimal storage guarantees...

Is there an algorithm that provably computes a low-rank approximation to a solution of the model problem + has optimal storage guarantees?
Conditional Gradient Method
Geometry of CGM

\[ H = \arg \max_{Y \in \mathcal{D}} \langle Y, -\nabla g(X) \rangle \]

\[ X_+ = (1 - \eta)X + \eta H \]

\[ \{ Y : g(Y) \leq g(X) \} \]

\[ \min_{X \in \mathcal{D}} g(X) \]

\[ \mathcal{D} = \{ Y \text{ psd} : \text{trace}(Y) = 1 \} \]
CGM for the Model Problem

Input: Problem data; suboptimality $\varepsilon$
Output: Approximate solution $X_{\text{cgm}}$

1. function CGM
2. $X \leftarrow 0$  \hspace{1cm} $\triangleright$ Initialize variable
3. for $t \leftarrow 0, 1, 2, 3, \ldots$ do
4. \hspace{1cm} $u \leftarrow \text{MinEigVec}(\mathcal{A}^*(\nabla f(\mathcal{A}X)))$  \hspace{1cm} $\triangleright$ Lanczos!
5. \hspace{1cm} $H \leftarrow -\alpha uu^*$  \hspace{1cm} $\triangleright$ Form update direction
6. \hspace{1cm} if $\langle X - H, \mathcal{A}^*(\nabla f(\mathcal{A}X)) \rangle \leq \varepsilon$
7. \hspace{2.5cm} then break for  \hspace{1cm} $\triangleright$ Stop when $\varepsilon$-suboptimal
8. \hspace{1cm} $\eta \leftarrow 2/(t + 2)$  \hspace{1cm} $\triangleright$ Update learning rate
9. \hspace{1cm} $X \leftarrow (1 - \eta)X + \eta H$  \hspace{1cm} $\triangleright$ Update variable
10. return $X$

Comment: In notation of last slide, $g = f \circ \mathcal{A}$. The gradient $\nabla g = \mathcal{A}^* \circ \nabla f \circ \mathcal{A}$.

Evolution of $\varepsilon$-Rank of CGM Iterates

Comments: Malaria data, quadratic loss, $\alpha = 1,400$. 
SketchyCGM
Crisis:

- CGM needs many iterations to converge to a near-low-rank solution
- The $\varepsilon$-rank of the CGM iterates can increase without bound
- CGM requires high + unpredictable storage
- Typically involves dynamic memory allocation

Opportunity:

- Modify CGM to work with optimal storage!
- Drive the CGM iteration with small “dual” variable $z = AX$
- Maintain small randomized sketch of primal matrix variable $X$
- After iteration terminates, reconstruct matrix variable $X$ from sketch

CGM, Redux

Input: Problem data; suboptimality $\varepsilon$
Output: Approximate solution $X_{\text{cgm}}$

function CGM

\[ X \leftarrow 0_{n \times n} \]

for $t \leftarrow 0, 1, 2, 3, \ldots$ do

\[ u \leftarrow \text{MinEigVec}(A^*(\nabla f(A X))) \]

\[ H \leftarrow -\alpha uu^* \]

if $\langle A (X - H), \nabla f(A X) \rangle \leq \varepsilon$

then break for

\[ \eta \leftarrow 2/(t + 2) \]

\[ X \leftarrow (1 - \eta) X + \eta H \]

return $X$

Idea: Apply $A$ to all expressions involving $X$ and $H$...
A Dual Formulation of CGM

Input: Problem data; suboptimality $\varepsilon$
Output: Approximate dual solution $AX_{cgm}$

1. function DUALCGM
2. $AX \leftarrow A0_{n \times n}$
3. for $t \leftarrow 0, 1, 2, 3, \ldots$ do
4. $u \leftarrow \text{MinEigVec}(A^*(\nabla f(AX)))$
5. $AH \leftarrow A(-\alpha uu^*)$
6. if $\langle A(X - H), \nabla f(AX) \rangle \leq \varepsilon$
7. then break for
8. $\eta \leftarrow 2/(t + 2)$
9. $AX \leftarrow (1 - \eta)AX + \eta AH$
10. return $AX$

Idea: Change variables $z = AX$ and $h = AH$...
A Dual Formulation of CGM

**Input:** Problem data; suboptimality $\varepsilon$

**Output:** Approximate dual solution $z_{cgm}$

```plaintext
function DUALCGM
    $z \leftarrow 0_d$
    for $t \leftarrow 0, 1, 2, 3, \ldots$ do
        $u \leftarrow \text{MinEigVec}(A^*(\nabla f(z)))$
        $h \leftarrow A(-\alpha uu^*)$
        if $\langle z - h, \nabla f(z) \rangle \leq \varepsilon$
            then break for
        $\eta \leftarrow 2/(t + 2)$
        $z \leftarrow (1 - \eta)z + \eta h$
    return $z$
```
A Dual Formulation of CGM

Input: Problem data; suboptimality $\varepsilon$
Output: Approximate dual solution $z_{\text{cgm}}$

function DUALCGM

1. $z \leftarrow 0_d$
2. for $t \leftarrow 0, 1, 2, 3, \ldots$ do
3. \hspace{1em} $u \leftarrow \text{MinEigVec}(A^*(\nabla f(z)))$
4. \hspace{1em} $h \leftarrow A(-\alpha uu^*)$
5. \hspace{1em} if $\langle z - h, \nabla f(z) \rangle \leq \varepsilon$
6. \hspace{2em} then break for
7. \hspace{1em} $\eta \leftarrow 2/(t + 2)$
8. \hspace{1em} $z \leftarrow (1 - \eta)z + \eta h$
9. return $z$

 Benefit: Only uses storage $\Theta(n + d)$!

Problem: Where do we get $X_{\text{cgm}}$?
Sketching the Decision Variable

- **Idea:** Maintain small sketch of primal variable \( X \! \)
- Fix target rank \( r \) of solution
- Draw Gaussian dimension reduction map

\[
\Omega \in \mathbb{C}^{n \times k} \quad \text{where} \quad k = 2r
\]

- Sketch takes the form

\[
Y = X \Omega \in \mathbb{C}^{n \times k}
\]

- Can perform linear update \( X \leftarrow (1 - \eta)X + \eta H \) by operating on sketch
- Can compute provably good rank-\( r \) approximation \( \hat{X} \) from sketch
- Only needs additional storage \( \Theta(rn) \! \)

SketchyCGM for the Model Problem

**Input:** Problem data; suboptimality $\varepsilon$; target rank $r$

**Output:** Rank-$r$ approximate solution $\hat{X} = V \Lambda V^*$ in factored form

```plaintext
function SketchyCGM
  Sketch.Init(n, r)  ▷ Initialize sketch to zero
  z ← 0
  for $t ← 0, 1, 2, 3, \ldots$ do
    $u ← \text{MinEigVec}(A^*(\nabla f(z)))$
    $h ← A(-\alpha uu^*)$
    if $\langle z - h, \nabla f(z) \rangle ≤ \varepsilon$ then break for
    $\eta ← 2/(t + 2)$
    $z ← (1 - \eta)z + \eta h$
    Sketch.CGMUpdate($-\sqrt{\alpha}u, \eta$) ▷ Update sketch of $X$
  (V, \Lambda) ← Sketch.Reconstruct() ▷ Approx. eigendecomposition of $X$
  return (V, \Lambda)
```

Methods for SKETCH Object

1. **Function Sketch.Init(n, r)**
   
   \( k \leftarrow 2r \)
   
   \( \Omega \leftarrow \text{randn}(\mathbb{C}, n, k) \)
   
   \( Y \leftarrow \text{zeros}(n, k) \)

   - Rank-\( r \) approx of \( n \times n \) psd matrix

2. **Function Sketch.CGMUpdate(s, \( \theta \))**
   
   \( Y \leftarrow (1 - \theta)Y + \theta s(s^*\Omega) \)

   - Average \( ss^* \) into sketch

3. **Function Sketch.Reconstruct()**
   
   \( C \leftarrow \text{chol}(\Omega^*Y) \)
   
   \( Z \leftarrow Y/C \)
   
   \((U, \Sigma, \sim) \leftarrow \text{svds}(Z, r)\)
   
   return \((U, \Sigma^2)\)

   - Cholesky decomposition
   - Solve least-squares problems
   - Compute \( r \)-truncated SVD
   - Return eigenvalue factorization

Theorem 1 (YUTC 2016). *SKETCHYCGM has the following properties:*

- *SKETCHYCGM has optimal storage guarantee* $\Theta(d + r n)$

- *SKETCHYCGM produces an $\varepsilon$-suboptimal objective value after* $O(\varepsilon^{-1})$ *iterations*

- *Suppose CGM produces iterates* $X_t$ *that converge to* $X_{\text{cgm}}$. *Then SKETCHYCGM produces rank-$r$ iterates* $\hat{X}_t$ *that satisfy*

$$\limsup_{t \to \infty} \mathbb{E} \left\| \hat{X}_t - X_{\text{cgm}} \right\|_{S_1} \leq \text{const} \cdot \left\| X_{\text{cgm}} - [X_{\text{cgm}}]_r \right\|_{S_1}$$

*In particular, if* $\text{rank}(X_{\text{cgm}}) \leq r$, *then* $\mathbb{E} \left\| \hat{X}_t - X_{\text{cgm}} \right\|_{S_1} \to 0$

*Source:* “Everything you always wanted in an algorithm. And less.”
https://www.youtube.com/watch?v=0agZEMEpiVI.
Performance of Sketchy CGM
Fourier Ptychography, Redux

29 illuminations; 80 × 80 pixels each; \( d = 1.86 \cdot 10^5 \) measurements
image size \( n = 160 \times 160 \) pixels; matrix size \( n^2 = 6.55 \cdot 10^8 \)
SKETCHYCGM storage (rank \( r = 1 \)): \( 6.53 \cdot 10^5 \)
quadatric loss

Fourier Ptychography: Malaria Phase Gradients

\[ \Delta_x \]

\[ \Delta_y \]

Wirtinger Flow  Burer–Monteiro  SKETCHYCGM
Convex Low-Rank Matrix Completion

Suppose $X_{\natural} \in \mathbb{R}^{m \times n}$ is a rank-$r$ matrix.

Observe a subset of entries + noise:

$$b_{ij} = (X_{\natural})_{ij} + \xi_{ij} \quad \text{for } (i, j) \in E$$

Matrix completion via convex programming:

$$\minimize_{X \in \mathbb{R}^{m \times n}} \sum_{(i,j) \in E} \text{loss}(x_{ij}; b_{ij}) \quad \text{subject to } \|X\|_{S_1} \leq \alpha$$

Matrix $X_{\natural}$ has about $r(m + n)$ degrees of freedom.

Convex method often effective when $\#E = \Theta(r(m + n))$.

But decision variable $X$ has $mn$ degrees of freedom!

SKETCHYCGM works with storage $\Theta(\#E + r(m + n))$.

MovieLens 10M

$m = 71,567$ users, $n = 10,681$ movies, $d = 10^7$ ratings, dim. $mn = 7.64 \cdot 10^8$

Denouement
Beyond...

- Other low-rank matrix optimization problems!
- More effective algorithms!
- Many applications!
- Beyond low-rank matrix optimization!
A Large-Scale Problem

29 illuminations; 250 × 250 pixels each; $d = 1.81 \cdot 10^6$ measurements
image size $n = 501 \times 501$ pixels; matrix size $n^2 = 6.25 \cdot 10^{10}$
solution via SKETCHYUPD

Sources: Cevher et al. 2017.
The MAXCUT SDP

MAXCUT SDP; sparse graph with 10,000 nodes (G67); via SKETCHYUPD

Sources: Goemans & Williamson 1995; Boumal 2015; Cevher et al. 2017.

Joel A. Tropp (Caltech), Sketchy Decisions, Data Sciences Program, NUS, 29 May 2017
To learn more...

E-mail: jtropp@cms.caltech.edu

Web: http://users.cms.caltech.edu/~jtropp

Papers:

- More to come!