Sketchy Decisions: Convex Low-Rank Matrix Optimization with Optimal Storage

Madeleine Udell

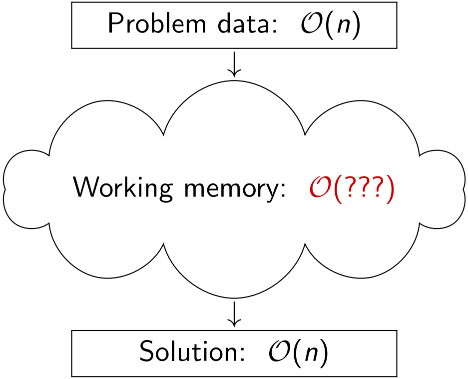
Operations Research and Information Engineering
Cornell University

Based on joint work with
Alp Yurtsever (EPFL), Volkan Cevher (EPFL),
and Joel Tropp (Caltech)

Simons Institute, October 4 2017

Desiderata

Suppose that the solution to a convex optimization problem has a **compact representation**.



Can we develop algorithms that provably solve the problem using **storage** bounded by the size of the **problem data** and the size of the **solution**?

Model problem: low rank matrix optimization

consider a convex problem with decision variable $X \in \mathbb{R}^{m \times n}$

compact matrix optimization problem:

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array} \qquad \text{(CMOP)}$$

- $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^d$
- $f : \mathbb{R}^d \to \mathbb{R}$ convex and smooth
- $\|X\|_{S_1}$ is Schatten-1 norm: sum of singular values

Model problem: low rank matrix optimization

consider a convex problem with decision variable $X \in \mathbb{R}^{m \times n}$

compact matrix optimization problem:

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array} \qquad \text{(CMOP)}$$

- $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^d$
- $f : \mathbb{R}^d \to \mathbb{R}$ convex and smooth
- $\|X\|_{S_1}$ is Schatten-1 norm: sum of singular values

assume

- **compact specification**: problem data use $\mathcal{O}(n)$ storage
- **compact solution**: $\operatorname{rank} X_\star = r$ constant

Model problem: low rank matrix optimization

consider a convex problem with decision variable $X \in \mathbb{R}^{m \times n}$

compact matrix optimization problem:

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array} \qquad \text{(CMOP)}$$

- $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^d$
- $f : \mathbb{R}^d \to \mathbb{R}$ convex and smooth
- $\|X\|_{S_1}$ is Schatten-1 norm: sum of singular values

assume

- **compact specification**: problem data use $\mathcal{O}(n)$ storage
- **compact solution**: rank $X_\star = r$ constant

Note: Same ideas work for $X \succeq 0$

Are desiderata achievable?

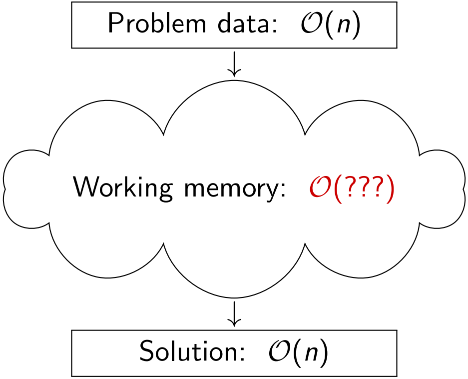
$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array}$$

CMOP, using any first order method:

$$\boxed{\text{Problem data: } \mathcal{O}(n)}$$
$$\downarrow$$
$$\boxed{\text{Working memory: } \mathcal{O}(n^2)}$$
$$\downarrow$$
$$\boxed{\text{Solution: } \mathcal{O}(n)}$$

Are desiderata achievable?

CMOP, using **???**:



Application: matrix completion

find X matching M on observed entries

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\Omega}(X_{ij} - M_{ij})^2 \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array}$$

- $m =$ rows, $n =$ columns of matrix to complete
- $d = |\Omega|$ number of observations
- $\mathcal{A}$ selects observed entries X_{ij}, $(i, j) \in \Omega$
- $f(\mathcal{A}X) = \|\mathcal{A}X - \mathcal{A}M\|^2$

compact if $d = \mathcal{O}(n)$ observations and $\text{rank}(X^\star)$ constant

Application: matrix completion

find X matching M on observed entries

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\Omega}(X_{ij} - M_{ij})^2 \\ \text{subject to} & \|X\|_{S_1} \le \alpha \end{array}$$

- $m =$ rows, $n =$ columns of matrix to complete
- $d = |\Omega|$ number of observations
- $\mathcal{A}$ selects observed entries X_{ij}, $(i, j) \in \Omega$
- $f(\mathcal{A}X) = \|\mathcal{A}X - \mathcal{A}M\|^2$

compact if $d = \mathcal{O}(n)$ observations and $\text{rank}(X^\star)$ constant
why is there a good approximation to M with constant rank?

Nice latent variable models

Suppose matrix $A \in \mathbb{R}^{m \times n}$ generated by a latent variable model:

- $\alpha_i \sim \mathcal{A}$ iid, $i = 1, \ldots, m$
- $\beta_j \sim \mathcal{B}$ iid, $j = 1, \ldots, n$
- $A_{ij} = g(\alpha_i, \beta_j)$

We say latent variable model is **nice** if

- distributions $\mathcal{A}$ and $\mathcal{B}$ have bounded support
- g is piecewise analytic and on each piece: for some $M \in \mathbb{R}$,

$$\|D^\mu g(\alpha, \beta)\| \leq C M^{|\mu|} \|g\|.$$

($\|g\| = \sup_{x \in \mathbf{dom}\, g} g(x)$ is sup norm.)

Examples: $g(\alpha, \beta) = \text{poly}(\alpha, \beta)$ or $g(\alpha, \beta) = \exp(\text{poly}(\alpha, \beta))$

Rank of nice latent variable models?

Question: How does rank of ϵ-approximation to $A \in \mathbb{R}^{m \times n}$ change with m and n?

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & \|X - M\|_\infty \leq \epsilon \end{array}$$

Rank of nice latent variable models?

Question: How does rank of ϵ-approximation to $A \in \mathbb{R}^{m \times n}$ change with m and n?

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & \|X - M\|_\infty \leq \epsilon \end{array}$$

Answer: rank grows as $\mathcal{O}(\log(m+n)/\epsilon^2)$

Rank of nice latent variable models?

Question: How does rank of ϵ-approximation to $A \in \mathbb{R}^{m \times n}$ change with m and n?

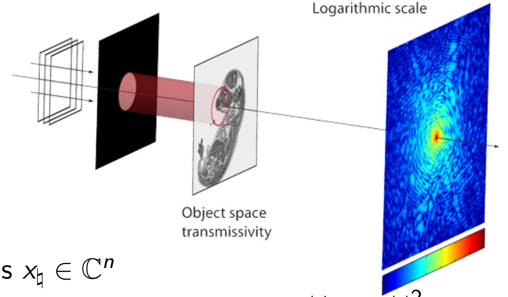
$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & \|X - M\|_\infty \leq \epsilon \end{array}$$

Answer: rank grows as $\mathcal{O}(\log(m+n)/\epsilon^2)$

Theorem (Udell and Townsend, 2017)

Nice latent variable models are of log rank.

Application: Phase retrieval



- image with n pixels $x_\natural \in \mathbb{C}^n$
- acquire noisy nonlinear measurements $b_i = |\langle a_i, x_\natural \rangle|^2 + \omega_i$
- relax: if $X = x_\natural x_\natural^*$, then

$$|\langle a_i, x_\natural \rangle|^2 = x_\natural a_i^* a_i x_\natural^* = \mathrm{tr}(a_i^* a_i x_\natural^* x_\natural) = \mathrm{tr}(a_i^* a_i X)$$

- recover image by solving

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X; b) \\ \text{subject to} & \mathrm{tr}\, X \le \alpha \\ & X \succeq 0. \end{array}$$

compact if $d = \mathcal{O}(n)$ observations and $\mathrm{rank}(X^\star)$ constant

Why compact?

why a compact specification?

- data is expensive
- collect constant data per column (=user or sample)
- if solution is compact, compact specification should suffice

why a compact solution?

- the world is simple and structured
- nice latent variable models are of log rank
- given d observations, there is a solution with rank $\mathcal{O}(\sqrt{d})$
  (Barvinok 1995, Pataki 1998)

Optimal Storage

What kind of storage bounds can we hope for?

- Assume black-box implementation of

$$\mathcal{A}(uv^*) \qquad u^*(\mathcal{A}^*z) \qquad (\mathcal{A}^*z)v$$

  where $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, and $z \in \mathbb{R}^d$
- Need $\Omega(m+n+d)$ storage to apply linear map
- Need $\Theta(r(m+n))$ storage for a rank-r approximate solution

Definition. An algorithm for the model problem has **optimal storage** if its working storage is

$$\Theta(d + r(m+n)).$$

Goal: optimal storage

We can specify the problem using $\mathcal{O}(n) \ll mn$ units of storage.

Can we solve the problem using only $\mathcal{O}(n)$ units of storage?

Goal: optimal storage

We can specify the problem using $\mathcal{O}(n) \ll mn$ units of storage.

Can we solve the problem using only $\mathcal{O}(n)$ units of storage?

If we write down X, we've already failed.

A brief biased history of matrix optimization

- 1990s: **Interior-point methods**
  - Storage cost $\Theta((m+n)^4)$ for Hessian
- 2000s: **Convex first-order methods (FOM)**
  - (Accelerated) proximal gradient and others
  - Store matrix variable $\Theta(mn)$

(**Interior-point:** Nemirovski & Nesterov 1994; . . . ; **First-order:** Rockafellar 1976; Auslender & Teboulle 2006; . . .)

A brief biased history of matrix optimization

- 2008–Present: **Storage-efficient convex FOM**
  - Conditional gradient method (CGM) and extensions
  - Store matrix in low-rank form $\mathcal{O}(t(m+n))$ after t iterations
  - Requires storage $\Theta(mn)$ for $t \geq \min(m, n)$
  - Variants: prune factorization, or seek rank-reducing steps

- 2003–Present: **Nonconvex heuristics**
  - Burer–Monteiro factorization idea + various opt algorithms
  - Store low-rank matrix factors $\Theta(r(m+n))$
  - For guaranteed solution, need unrealistic + unverifiable statistical assumptions

(**CGM:** Frank & Wolfe 1956; Levitin & Poljak 1967; Hazan 2008; Clarkson 2010; Jaggi 2013; . . . ; **CGM + pruning:** Rao Shah Wright 2015; Freund Grigas Mazumder 2017; . . . ; **Heuristics:** Burer & Monteiro 2003; Keshavan et al. 2009; Jain et al. 2012; Bhojanapalli et al. 2015; Candès et al. 2014; Boumal et al. 2015; . . .)

The dilemma

- convex methods: slow memory hogs with guarantees
- nonconvex methods: fast, lightweight, but brittle

The dilemma

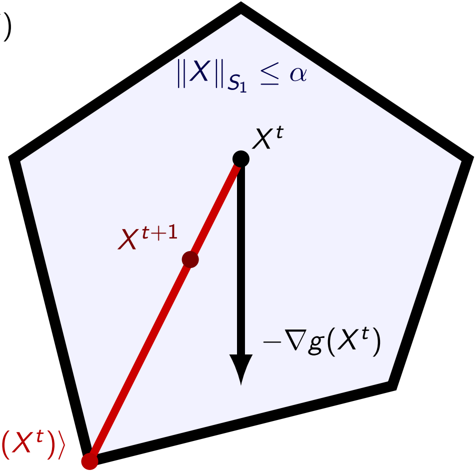
- convex methods: slow memory hogs with guarantees
- nonconvex methods: fast, lightweight, but brittle

low memory or guaranteed convergence

. . . but not both?

Conditional Gradient Method

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) = g(X) \\ \text{subject to} & \|X\|_{S_1} \leq \alpha \end{array}$$



$$H^t = \underset{\|X\|_{S_1} \leq \alpha}{\text{argmax}} \ \langle X, -\nabla g(X^t) \rangle$$

Conditional Gradient Method

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X) \\ \text{subject to} & \|X\|_{S_1} \le \alpha \end{array}$$

CGM. set $X^0 = 0$. for $t = 0, 1, \ldots$

- compute $G^t = \mathcal{A}^* \nabla f(\mathcal{A}X^t)$
- set search direction

$$H^t = \underset{\|X\|_{S_1} \le \alpha}{\operatorname{argmax}} \langle X, -G^t \rangle$$

- set stepsize $\eta^t = 2/(t+2)$
- update $X^{t+1} = (1 - \eta^t) X^t + \eta^t H^t$

Conditional gradient method (CGM)

features:

- relies on efficient **linear optimization oracle** to compute

$$H^t = \underset{\|X\|_{S_1} \leq \alpha}{\operatorname{argmax}} \langle X, -G^t \rangle$$

- bound on suboptimality follows from subgradient inequality

$$\begin{aligned} f(\mathcal{A}X^t) - f(\mathcal{A}X^\star) &\leq \langle X^t - X^\star, G^t \rangle \\ &\leq \langle X^t - X^\star, \mathcal{A}^* \nabla f(\mathcal{A}X^t) \rangle \\ &\leq \langle \mathcal{A}X^t - \mathcal{A}X^\star, \nabla f(\mathcal{A}X^t) \rangle \\ &\leq \langle \mathcal{A}X^t - \mathcal{A}H^t, \nabla f(\mathcal{A}X^t) \rangle \end{aligned}$$

  to provide stopping condition
- faster variants: linesearch, away steps, ...

Linear optimization oracle for MOP

compute search direction

$$\underset{\|X\|_{S_1}\leq\alpha}{\operatorname{argmax}}\langle X, -G\rangle$$

Linear optimization oracle for MOP

compute search direction

$$\underset{\|X\|_{S_1} \leq \alpha}{\operatorname{argmax}} \langle X, -G \rangle$$

- solution given by maximum singular vector of $-G$:

$$-G = \sum_{i=1}^{n} \sigma_i u_i v_i^* \quad \Longrightarrow \quad X = \alpha u_1 v_1^*$$

- use Lanczos method: only need to apply G and G^*

Conditional gradient descent

Algorithm 1 CGM for the model problem (CMOP)

Input: Problem data for (CMOP); suboptimality ε
Output: Solution $X_\star$

```
function CGM
    X ← 0
    for t ← 0, 1, ... do
        (u, v) ← MaxSingVec(−𝒜*(∇f(𝒜X)))
        H ← −α uv*
        if ⟨𝒜X − 𝒜H, ∇f(𝒜X)⟩ ≤ ε then break for
        η ← 2/(t + 2)
        X ← (1 − η)X + ηH
    return X
```

Two crucial ideas

To solve the problem using optimal storage:

- Use the low-dimensional "dual" variable

$$z_t = \mathcal{A}X_t \in \mathbb{R}^d$$

  to drive the iteration.
- Recover solution from small (randomized) sketch.

Two crucial ideas

To solve the problem using optimal storage:

- Use the low-dimensional "dual" variable

$$z_t = \mathcal{A}X_t \in \mathbb{R}^d$$

  to drive the iteration.
- Recover solution from small (randomized) sketch.

Never write down X until it has converged to low rank.

Conditional gradient descent

Algorithm 2 CGM for the model problem (CMOP)

Input: Problem data for (CMOP); suboptimality ε
Output: Solution $X_\star$

function CGM
  $X \leftarrow 0$
  for $t \leftarrow 0, 1, \ldots$ **do**
    $(u, v) \leftarrow \texttt{MaxSingVec}(-\mathcal{A}^*(\nabla f(\mathcal{A}X)))$
    $H \leftarrow -\alpha\, uv^*$
    if $\langle \mathcal{A}X - \mathcal{A}H, \nabla f(\mathcal{A}X)\rangle \leq \varepsilon$ **then break for**
    $\eta \leftarrow 2/(t+2)$
    $X \leftarrow (1-\eta)X + \eta H$
  return X

Conditional gradient descent

Introduce "dual variable" $z = \mathcal{A}X \in \mathbb{R}^d$; eliminate X.

Algorithm 3 Dual CGM for the model problem (CMOP)

Input: Problem data for (CMOP); suboptimality ε
Output: Solution $X_\star$

```
function dualCGM
    z ← 0
    for t ← 0, 1, ... do
        (u, v) ← MaxSingVec(−A*(∇f(z)))
        h ← A(−α u v*)
        if ⟨z − h, ∇f(z)⟩ ≤ ε then break for
        η ← 2/(t + 2)
        z ← (1 − η)z + ηh
```

Conditional gradient descent

Introduce "dual variable" $z = \mathcal{A}X \in \mathbb{R}^d$; eliminate X.

Algorithm 4 Dual CGM for the model problem (CMOP)

Input: Problem data for (CMOP); suboptimality ε
Output: Solution $X_\star$

function $\textsc{dualCGM}$
    $z \leftarrow 0$
    for $t \leftarrow 0, 1, \ldots$ **do**
        $(u, v) \leftarrow \texttt{MaxSingVec}(-\mathcal{A}^*(\nabla f(z)))$
        $h \leftarrow \mathcal{A}(-\alpha u v^*)$
        if $\langle z - h,\, \nabla f(z)\rangle \leq \varepsilon$ **then break for**
        $\eta \leftarrow 2/(t+2)$
        $z \leftarrow (1-\eta)z + \eta h$

we've solved the problem... but where's the solution?

Two crucial ideas

1. Use the low-dimensional "dual" variable

$$z_t = \mathcal{A}X_t \in \mathbb{R}^d$$

   to drive the iteration.
2. Recover solution from small (randomized) sketch.

How to catch a low rank matrix

if $\hat{X}$ has the same rank as $X^\star$,
and $\hat{X}$ acts like $X^\star$ (on its range and co-range),
then $\hat{X}$ is $X^\star$

How to catch a low rank matrix

if $\hat{X}$ has the same rank as $X^\star$,
and $\hat{X}$ acts like $X^\star$ (on its range and co-range),
then $\hat{X}$ is $X^\star$

- see a series of additive updates
- remember how the matrix acts
- reconstruct a low rank matrix that acts like $X^\star$

Single-pass randomized sketch

- Draw and fix two independent standard normal matrices

$$\Omega \in \mathbb{R}^{n \times k} \quad \text{and} \quad \Psi \in \mathbb{R}^{\ell \times m}$$

with $k = 2r + 1$, $\ell = 4r + 2$.

Single-pass randomized sketch

- Draw and fix two independent standard normal matrices

$$\Omega \in \mathbb{R}^{n\times k} \quad \text{and} \quad \Psi \in \mathbb{R}^{\ell\times m}$$

  with $k = 2r + 1$, $\ell = 4r + 2$.
- The sketch consists of two matrices that capture the range and co-range of X:

$$Y = X\Omega \in \mathbb{R}^{n\times k} \quad \text{and} \quad W = \Psi X \in \mathbb{R}^{\ell\times m}$$

Single-pass randomized sketch

- Draw and fix two independent standard normal matrices

$$\Omega \in \mathbb{R}^{n\times k} \quad \text{and} \quad \Psi \in \mathbb{R}^{\ell\times m}$$

  with $k = 2r + 1$, $\ell = 4r + 2$.
- The sketch consists of two matrices that capture the range and co-range of X:

$$Y = X\Omega \in \mathbb{R}^{n\times k} \quad \text{and} \quad W = \Psi X \in \mathbb{R}^{\ell\times m}$$

- Rank-1 updates to X can be performed on sketch:

$$X' = \beta_1 X + \beta_2 uv^*$$
$$\Downarrow$$
$$Y' = \beta_1 Y + \beta_2 uv^*\Omega \quad \text{and} \quad W' = \beta_1 W + \beta_2 \Psi uv^*$$

Single-pass randomized sketch

- Draw and fix two independent standard normal matrices

$$\Omega \in \mathbb{R}^{n \times k} \quad \text{and} \quad \Psi \in \mathbb{R}^{\ell \times m}$$

  with $k = 2r + 1$, $\ell = 4r + 2$.
- The sketch consists of two matrices that capture the range and co-range of X:

$$Y = X\Omega \in \mathbb{R}^{n \times k} \quad \text{and} \quad W = \Psi X \in \mathbb{R}^{\ell \times m}$$

- Rank-1 updates to X can be performed on sketch:

$$X' = \beta_1 X + \beta_2 uv^*$$

$$\Downarrow$$

$$Y' = \beta_1 Y + \beta_2 uv^* \Omega \quad \text{and} \quad W' = \beta_1 W + \beta_2 \Psi uv^*$$

- Both the storage cost for the sketch and the arithmetic cost of an update are $\mathcal{O}(r(m+n))$.

Recovery from sketch

To recover rank-r approximation $\hat{X}$ from the sketch, compute

1. $Y = QR$ (tall-skinny QR)
2. $B = (\Psi Q)^\dagger W$ (small QR + backsub)
3. $\hat{X} = Q[B]_r$ (tall-skinny SVD)

Recovery from sketch

To recover rank-r approximation $\hat{X}$ from the sketch, compute

1. $Y = QR$ (tall-skinny QR)
2. $B = (\Psi Q)^{\dagger} W$ (small QR + backsub)
3. $\hat{X} = Q[B]_r$ (tall-skinny SVD)

Theorem (Reconstruction (Tropp Yurtsever U Cevher, 2016))

Fix a target rank r. Let X be a matrix, and let (Y, W) be a sketch of X. The reconstruction procedure above yields a rank-r matrix $\hat{X}$ with

$$\mathbb{E}\,\|X - \hat{X}\|_{\mathrm{F}} \leq 2\,\|X - [X]_r\|_{\mathrm{F}}\,.$$

Similar bounds hold with high probability.

Previous work (Clarkson Woodruff 2009) algebraically but not numerically equivalent.

Recovery from sketch: intuition

recall

$$Y = X\Omega \in \mathbb{R}^{n \times k} \quad \text{and} \quad W = \Psi X \in \mathbb{R}^{\ell \times m}$$

- if Q is an orthonormal basis for $\mathcal{R}(X)$, then

$$X = QQ^* X$$

- if $QR = X\Omega$, then Q is (approximately) a basis for $\mathcal{R}(X)$
- and if $W = \Psi X$, we can estimate

$$\begin{aligned} W &= \Psi X \\ &\approx \Psi Q Q^* X \\ (\Psi Q)^\dagger W &\approx Q^* X \end{aligned}$$

- hence we may reconstruct X as

$$X \approx QQ^* X \approx Q(\Psi Q)^\dagger W$$

SketchyCGM

Algorithm 5 SketchyCGM for the model problem (CMOP)

Input: Problem data; suboptimality ε; target rank r
Output: Rank-r approximate solution $\hat{X} = U\Sigma V^*$

function $\textsc{SketchyCGM}$
  $\textsc{Sketch.Init}(m, n, r)$
  $z \leftarrow 0$
  for $t \leftarrow 0, 1, \ldots$ **do**
    $(u, v) \leftarrow \texttt{MaxSingVec}(-\mathcal{A}^*(\nabla f(z)))$
    $h \leftarrow \mathcal{A}(-\alpha uv^*)$
    if $\langle z - h, \nabla f(z)\rangle \leq \varepsilon$ **then break for**
    $\eta \leftarrow 2/(t + 2)$
    $z \leftarrow (1 - \eta)z + \eta h$
    $\textsc{Sketch.CGMUpdate}(-\alpha u, v, \eta)$
  $(U, \Sigma, V) \leftarrow \textsc{Sketch.Reconstruct}(\,)$
  return (U, Σ, V)

Guarantees

Suppose

- $X_{\mathrm{cgm}}^{(t)}$ is tth CGM iterate
- $\lfloor X_{\mathrm{cgm}}^{(t)} \rfloor_r$ is best rank r approximation to CGM solution
- $\hat{X}^{(t)}$ is SketchyCGM reconstruction after t iterations

Theorem (Convergence to CGM solution)

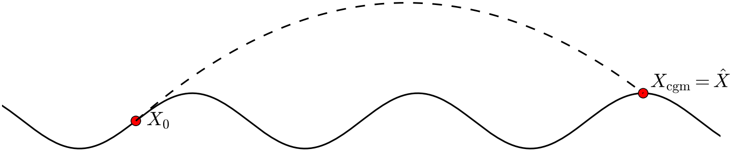
After t iterations, the SketchyCGM reconstruction satisfies

$$\mathbb{E}\,\|\hat{X}^{(t)} - X_{\mathrm{cgm}}^{(t)}\|_{\mathrm{F}} \leq 2\,\|\lfloor X_{\mathrm{cgm}}^{(t)} \rfloor_r - X_{\mathrm{cgm}}^{(t)}\|_{\mathrm{F}}\,.$$

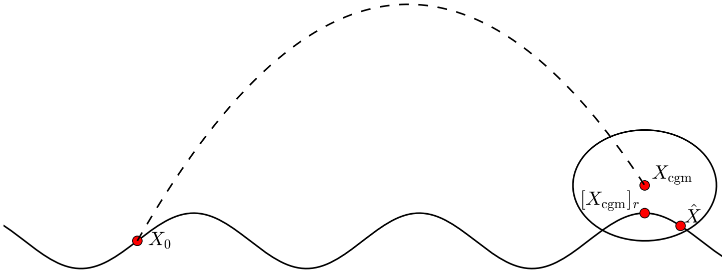
If in addition $X^\star = \lim_{t\to\infty} X_{\mathrm{cgm}}^{(t)}$ has rank r, then RHS $\to 0$!

(Tropp Yurtsever U Cevher, 2016)

Convergence when $\text{rank}(X_{\mathbf{cgm}}) \leq r$



Convergence when $\mathrm{rank}(X_{\mathbf{cgm}}) > r$



Guarantees (II)

Theorem (Convergence rate)

Fix $\kappa > 0$ and $\nu \geq 1$. Suppose the (unique) solution $X_\star$ of (CMOP) *has* $\operatorname{rank}(X_\star) \leq r$ *and*

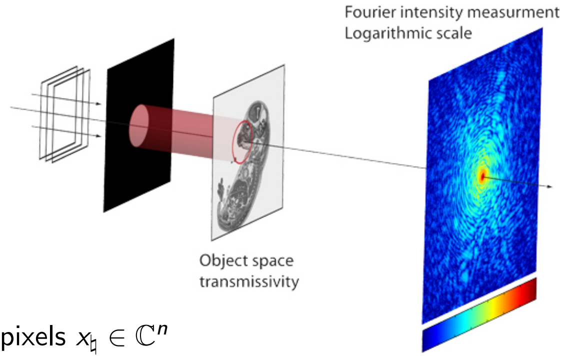
$$f(\mathcal{A}X) - f(\mathcal{A}X_\star) \geq \kappa \, \|X - X_\star\|_{\mathrm{F}}^{\nu} \quad \text{for all} \quad \|X\|_{S_1} \leq \alpha. \qquad (1)$$

Then we have the error bound

$$\mathbb{E}\, \|\hat{X}_t - X_\star\|_{\mathrm{F}} \leq 6 \left(\frac{2\kappa^{-1} C}{t+2} \right)^{1/\nu} \quad \text{for } t = 0, 1, 2, \ldots$$

where C is the curvature constant (Eqn. (3), Jaggi 2013) of the problem (CMOP).

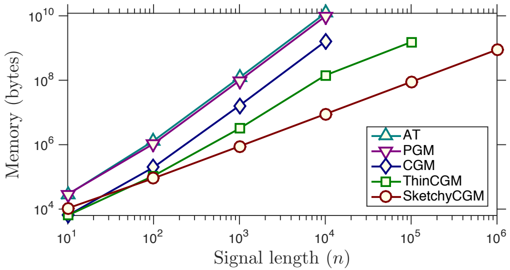
Application: Phase retrieval



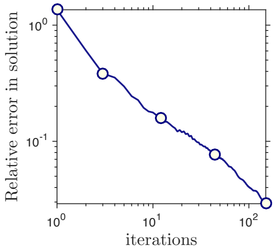
- image with n pixels $x_\natural \in \mathbb{C}^n$
- acquire noisy measurements $b_i = |\langle a_i, x_\natural \rangle|^2 + \omega_i$
- recover image by solving

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}X; b) \\ \text{subject to} & \operatorname{tr} X \leq \alpha \\ & X \succeq 0. \end{array}$$

SketchyCGM is scalable



(A) Memory usage for five algorithms



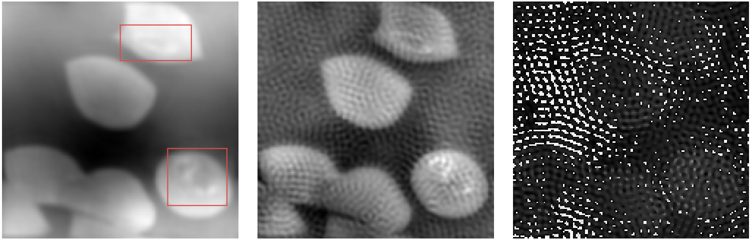
(B) Convergence for $n = 8 \cdot 10^6$.

PGM	=	proximal gradient (via TFOCS (Becker Candès Grant, 2011))
AT	=	accelerated PGM (Auslander Teboulle, 2006) (via TFOCS),
CGM	=	conditional gradient method (Jaggi, 2013)
ThinCGM	=	CGM with thin SVD updates (Yurtsever Hsieh Cevher, 2015)
SketchyCGM	=	ours, using $r = 1$

SketchyCGM is reliable

Fourier ptychography:

- imaging blood cells with $\mathcal{A}$ = subsampled FFT
- $n = 25,600$, $d = 185,600$
- $\text{rank}(X_\star) \approx 5$ (empirically)



(A) SketchyCGM (B) Burer–Monteiro (C) Wirtinger Flow

- brightness indicates phase of pixel (thickness of sample)
- red boxes mark malaria parasites in blood cells

Conclusion

SketchyCGM offers a proof-of-concept **convex method** with **optimal storage** for low rank matrix optimization using two new ideas:

- Drive the algorithm using a smaller (dual) variable.
- Sketch and recover the decision variable.

References:

- J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Randomized single-view algorithms for low-rank matrix reconstruction. SIMAX (to appear).
- A. Yurtsever, M. Udell, J. A. Tropp, and V. Cevher. Sketchy Decisions: Convex Optimization with Optimal Storage. AISTATS 2017.