Storage optimal semidefinite programming

Volkan Cevher
https://lions.epfl.ch

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Switzerland

École Polytechnique, Paris

[June 2019]

Semi-definite programming relaxations

linear programming (LP) $\quad \min_x \{c^* x : A(x) = b, x \geq 0\}$

semi-definite programming (SDP) $\quad \min_x \{\mathrm{tr}(cx) : A(x) = b, x \succeq 0, x^* = x\}$

LP $\subseteq$ QP $\subseteq$ QCQP $\subseteq$ SOCP $\subseteq$ SDP

Semi-definite programming relaxations

linear programming (LP) $\quad \min_x \{c^*x : A(x) = b, x \geq 0\}$

semi-definite programming (SDP) $\quad \min_x \{\mathrm{tr}(cx) : A(x) = b, x \succeq 0, x^* = x\}$

LP $\subseteq$ QP $\subseteq$ QCQP $\subseteq$ SOCP $\subseteq$ SDP

◦ Relaxations for combinatorial optimization and other difficult problems

▷ e.g., max-cut, clustering, quadratic assignment, power-flow,...

If unique games conjecture is true, SDP relaxation is the best we can do

▷ e.g., robustness of neural networks, GAN denoising

Example: Finding maximum-weight cut of a graph

◦ **Goal:** Given an undirected graph $G = (V, E)$ with a set of weights $c : E \to \mathbb{R}_+$

$$\min_{x \in \mathbb{Z}^p} \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_i x_j) : x_i \in \{-1, +1\} \right\} \qquad \text{(Weighted max-cut)}$$

Example: Finding maximum-weight cut of a graph

- **Goal:** Given an undirected graph $G = (V, E)$ with a set of weights $c : E \to \mathbb{R}_+$

$$\min_{x \in \mathbb{Z}^p} \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_i x_j) : x_i \in \{-1, +1\} \right\} \qquad \text{(Weighted max-cut)}$$

- **The SDP approach:** Lift & relax

  ▷ lift as a matrix optimization problem

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij}) : \mathrm{diag}(x) = 1,\ x \succeq 0,\ x^* = x,\ \mathrm{rank}(x) = 1 \right\}$$

  ▷ relax the non-convex rank constraint

$$\min_{x \in \mathbb{R}^{p \times p}} \Bigg\{ \underbrace{\frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij})}_{\mathrm{tr}(cx)} : \underbrace{\mathrm{diag}(x) = 1}_{A(x)=b},\ x \succeq 0,\ x^* = x \Bigg\} \qquad \text{(Max-cut SDP)}$$

Example: Finding maximum-weight cut of a graph

- **Goal:** Given an undirected graph $G = (V, E)$ with a set of weights $c : E \to \mathbb{R}_+$

$$\min_{x \in \mathbb{Z}^p} \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_i x_j) : x_i \in \{-1, +1\} \right\} \qquad \text{(Weighted max-cut)}$$

- **The SDP approach:** Lift & relax

  ▷ lift as a matrix optimization problem

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij}) : \mathrm{diag}(x) = 1,\ x \succeq 0,\ x^* = x,\ \mathrm{rank}(x) = 1 \right\}$$

  ▷ relax the non-convex rank constraint

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \underbrace{\frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij})}_{\mathrm{tr}(cx)} : \underbrace{\mathrm{diag}(x) = 1}_{A(x) = b},\ x \succeq 0,\ x^* = x \right\} \qquad \text{(Max-cut SDP)}$$

- Always delivers solutions 0.87856 times the optimal value after randomized rounding

Example: Clustering with minimal sum-of-squares

- **Goal:** Given data points $s_1, s_2, \ldots, s_p \in \mathbb{R}^q$, assign them into k disjoint clusters.

  ▷ Minimize the sum of squared distances of all points to their cluster centers

$$\min_z \left\{ \sum_{j=1}^{k} \sum_{i=1}^{p} z_{ij} \left\| s_i - w_j(z) \right\|^2 : \sum_{j=1}^{k} z_{ij} = 1, \sum_{i=1}^{p} z_{ij} \geq 1, z_{ij} \in \{0,1\} \right\} \tag{MinSumClu.}$$

where $z \in \{0,1\}^{p \times k}$ is the assignment matrix with $z_{ij} = \begin{cases} 1 & \text{if } s_i \in j\text{th cluster} \\ 0 & \text{otherwise} \end{cases}$

where $w_1, \ldots, w_k$ are cluster centers with $w_j(z) = \left(\sum_{i=1}^{p} z_{ij} s_i\right) \left(\sum_{i=1}^{p} z_{ij}\right)^{-1}$

Example: Clustering with minimal sum-of-squares

◦ **Goal:** Given data points $s_1, s_2, \ldots, s_p \in \mathbb{R}^q$, assign them into k disjoint clusters.

▷ Minimize the sum of squared distances of all points to their cluster centers

$$\min_z \left\{ \sum_{j=1}^{k} \sum_{i=1}^{p} z_{ij} \|s_i - w_j(z)\|^2 : \sum_{j=1}^{k} z_{ij} = 1, \sum_{i=1}^{p} z_{ij} \geq 1, z_{ij} \in \{0,1\} \right\} \tag{MinSumClu.}$$

where $z \in \{0,1\}^{p\times k}$ is the assignment matrix with $z_{ij} = \begin{cases} 1 & \text{if } s_i \in j\text{th cluster} \\ 0 & \text{otherwise} \end{cases}$

where $w_1, \ldots, w_k$ are cluster centers with $w_j(z) = \left(\sum_{i=1}^{p} z_{ij} s_i\right)\left(\sum_{i=1}^{p} z_{ij}\right)^{-1}$

◦ **The SDP approach:** Lift & relax (details omitted)

$$\min_{x\in\mathbb{R}^{p\times p}} \left\{ \mathrm{tr}(cx) : x \geq 0,\ x1 = 1,\ x \succeq 0,\ x^* = x,\ \mathrm{tr}(x) = k \right\} \qquad \text{(Clustering SDP)}$$

where $x = z(z^*z)^{-1}z^*$ and $c_{ij} = \|s_i - s_j\|^2$

Example: Clustering with minimal sum-of-squares

◦ **Goal:** Given data points $s_1, s_2, \ldots, s_p \in \mathbb{R}^q$, assign them into k disjoint clusters.

▷ Minimize the sum of squared distances of all points to their cluster centers

$$\min_{z} \left\{ \sum_{j=1}^{k} \sum_{i=1}^{p} z_{ij} \left\| s_i - w_j(z) \right\|^2 : \sum_{j=1}^{k} z_{ij} = 1, \sum_{i=1}^{p} z_{ij} \geq 1, z_{ij} \in \{0,1\} \right\} \tag{MinSumClu.}$$

where $z \in \{0,1\}^{p \times k}$ is the assignment matrix with $z_{ij} = \begin{cases} 1 & \text{if } s_i \in j\text{th cluster} \\ 0 & \text{otherwise} \end{cases}$

where $w_1, \ldots, w_k$ are cluster centers with $w_j(z) = \left(\sum_{i=1}^{p} z_{ij} s_i\right) \left(\sum_{i=1}^{p} z_{ij}\right)^{-1}$

◦ **The SDP approach:** Lift & relax (details omitted)

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \mathrm{tr}(cx) : x \geq 0,\ x1 = 1,\ x \succeq 0,\ x^* = x,\ \mathrm{tr}(x) = k \right\} \tag{Clustering SDP}$$

where $x = z(z^* z)^{-1} z^*$ and $c_{ij} = \|s_i - s_j\|^2$

◦ Improved guarantees over LP relaxations

J.Peng and Y.Wei, Approximating K-means-type clustering via semidefinite programming, 2005

Example: Neural networks

◦ **Goal:** Approximate the ℓ_∞-Lipschitz constant L_f of 1-layer ReLU network

$$f(z) := v^T \sigma(Wz + m)$$

▷ applications to robustness against adversarial examples, generalization...

Example: Neural networks

◦ **Goal:** Approximate the ℓ_∞-Lipschitz constant L_f of 1-layer ReLU network

$$f(z) := v^T \sigma(Wz + m)$$

▷ applications to robustness against adversarial examples, generalization...

◦ **The SDP approach:** Lift & relax (details omitted)

$$L_f \leq \bar{L}_f := -\frac{1}{4} \min_{x \in \mathbb{R}^{p \times p}} \{\mathsf{tr}(cx) : x \succeq 0, \mathrm{diag}(x) = 1\}$$

$$c := -\begin{bmatrix} 0 & 0 & \mathbf{1}^T W^T \mathsf{Diag}(v) \\ 0 & 0 & W^T \mathsf{Diag}(v) \\ \mathsf{Diag}(v)^T W \mathbf{1} & \mathsf{Diag}(v)^T W & 0 \end{bmatrix}$$

Example: Neural networks

◦ **Goal:** Approximate the ℓ_∞-Lipschitz constant L_f of 1-layer ReLU network

$$f(z) := v^T \sigma(Wz + m)$$

▷ applications to robustness against adversarial examples, generalization...

◦ **The SDP approach:** Lift & relax (details omitted)

$$L_f \leq \bar{L}_f := -\frac{1}{4} \min_{x \in \mathbb{R}^{p \times p}} \{\mathrm{tr}(cx) : x \succeq 0, \mathrm{diag}(x) = 1\}$$

$$c := -\begin{bmatrix} 0 & 0 & \mathbf{1}^T W^T \mathrm{Diag}(v) \\ 0 & 0 & W^T \mathrm{Diag}(v) \\ \mathrm{Diag}(v)^T W \mathbf{1} & \mathrm{Diag}(v)^T W & 0 \end{bmatrix}$$

◦ An open research area

Ragunathan et al. SDP relaxations for certifying robustness agains adversarial examples. ICLR2017

Structures in the SDP relaxations

$$\boxed{\min_x \{\mathsf{tr}(cx) : Ax = b, x \succeq 0, x^* = x\}}$$

- The decision variable has $\mathcal{O}(p^2)$-degrees of freedom
  - ▷ it's yuge!
  - ▷ need $\Theta(p^2)$ storage

Structures in the SDP relaxations

$$\boxed{\min_x \{\mathrm{tr}(cx) : Ax = b, x \succeq 0, x^* = x\}}$$

- The decision variable has $\mathcal{O}(p^2)$-degrees of freedom
  - it's yuge!
  - need $\Theta(p^2)$ storage

- Optimal solutions ($x^\star$) typically or approximately have $\mathcal{O}(rp)$-degrees of freedom
  - r: rank & $r \ll p$: low-rank
  - need $\Theta(rp)$ storage for a rank-r approximate solution

Structures in the SDP relaxations

$$\boxed{\min_x \{\mathrm{tr}(cx) : Ax = b, x \succeq 0, x^* = x\}}$$

- The decision variable has $\mathcal{O}(p^2)$-degrees of freedom
  - ▷ it's yuge!
  - ▷ need $\Theta(p^2)$ storage
- Optimal solutions ($x^\star$) typically or approximately have $\mathcal{O}(rp)$-degrees of freedom
  - ▷ r: rank & $r \ll p$: low-rank
  - ▷ need $\Theta(rp)$ storage for a rank-r approximate solution
- Example SDP's typically have $n = \tilde{\mathcal{O}}(p)$ affine constraints (ignore SoS)
  - ▷ affine constraints implement

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

    where $u \in \mathbb{R}^p$ and $v \in \mathbb{R}^p$ and $z \in \mathbb{R}^n$
  - ▷ need $\Omega(n + p)$ storage for computations with linear map

Structures in the SDP relaxations

Need $\underline{\Theta(n + rp)}$ storage to specify the problem and its solution

$$\boxed{\min_x \{\mathrm{tr}(cx) : Ax = b, x \succeq 0, x^* = x\}}$$

- The decision variable has $\mathcal{O}(p^2)$-degrees of freedom
  - ▷ it's yuge!
  - ▷ need $\Theta(p^2)$ storage $\leftarrow$ **this is a major problem**
- Optimal solutions ($x^\star$) typically or approximately have $\mathcal{O}(rp)$-degrees of freedom
  - ▷ r: rank & $r \ll p$: low-rank
  - ▷ need $\Theta(rp)$ storage for a rank-r approximate solution
- Example SDP's typically have $n = \tilde{\mathcal{O}}(p)$ affine constraints (ignore SoS)
  - ▷ affine constraints implement

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

where $u \in \mathbb{R}^p$ and $v \in \mathbb{R}^p$ and $z \in \mathbb{R}^n$

  - ▷ need $\Omega(n + p)$ storage for computations with linear map

Storage eliminates standard convex approaches for SDP

- First & second order methods (AGD, PD, NM, ...)
  - ▷ store matrix variable $\Theta(p^2)$
  - ▷ Hessian is worse
  - ▷ $\mathcal{O}\left(p^{3.5}\log(p/\epsilon)\right)$ total complexity (NM)

Non-convex approach: Burer-Monteiro factorization

◦ SDP template:

$$\min_{x\in\mathbb{R}^{p\times p}} \left\{ \mathsf{tr}(cx) : Ax = b, x \succeq 0, x^* = x, \mathsf{tr}(x) = \rho \right\}$$

◦ Burer-Monteiro splitting

$$\min_{u\in\mathbb{R}^{p\times r}} \left\{ \mathsf{tr}(cuu^*) : Auu^* = b, u \in \mathcal{U} := \{u : \|u\|_F \leq \sqrt{\rho}\} \right\}$$

▷ Nonlinear and non-convex problem $(Lu = Auu^* = b,\ \mathsf{tr}(cuu^*))$

▷ Local minima vs. saddle points issues

▷ Local minima vs. global minimum: $r = \Omega(\sqrt{p})$, due to Pataki and Barvinok

Barvinok, Problems of distance geometry and convex properties of quadratic maps, Disc Comp Geo, 1995.
Pataki, On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues, Math Oper Res, 1998.

State-of-the-art

- Substitute $uu^* \to x$ in the convex augmented Lagrangian

$$\mathcal{L}_\beta(u,y) = \mathrm{tr}(cuu^*) + \langle y, Auu^* - b\rangle + \frac{1}{2\beta}\|Auu^* - b\|^2.$$

- Burer-Monteiro's heuristic: $\begin{cases} u^+ = \arg\min_u \mathcal{L}_\beta(u,y) \\ \text{Update } y^+ \text{ or } \beta^+ \text{ according to feasibility progress} \end{cases}$
  - ▷ No inexact analysis for solving subproblems
  - ▷ Subproblem complexities e.g., $\begin{cases} \text{APGM (Ghadimi \& Lan, 2016): } \mathcal{O}(\frac{1}{\epsilon}) \\ \text{Trust region (Cartis et al., 2012):} \mathcal{O}(\frac{1}{\epsilon^3}) \end{cases}$
- Manifold optimization (ManOpt):
  - ▷ Smooth manifold assumption: Requires projectable sets
  - ▷ $\mathcal{O}\left(p^{10}/\epsilon^3\right)$ total complexity—$\mathcal{O}\left(p^6\right)$ flops per iteration
- Others like ALBUM (Bolte-Sabach-Teboulle) with caveats

Burer, Monteiro. Local minima and convergence in low-rank semidefinite programming. Math Prog, 2005.
Boumal, Mishra, Absil, Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds, JMLR, 2014.
Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming, Math Prog, 2016.
Cartis, C., Gould, N. I., and Toint, P. L. Complexity bounds for second-order optimality in unconstrained optimization, JoC, 2012.

Inexact augmented Lagrangian framework

◦ Our idea: Solve primal subproblems with stricter tolerance, i.e., $\epsilon \to 0$

$$
\text{iALM:} \begin{cases} \text{Obtain} & u^+ \text{ such that} \\ & \text{dist}(-\nabla_u \mathcal{L}_\beta(u^+, y), \partial g(u^+)) \leq \epsilon_f \text{ , or} \qquad \text{[1st order stationarity]} \\ & \lambda_{\text{min}}(\nabla_{uu} \mathcal{L}_\beta(u^+, y)) \geq -\epsilon_s \qquad \text{[2nd order stationarity]} \\ y^+ = & y + \sigma \left(L(u^+) - b \right) \\ \text{Pick} & \beta^+ < \beta \text{ and } \epsilon^+ = \beta^+ \\ \text{Update} & \sigma^+ = \sigma_0 \min \left(\frac{1}{||L(u) - b|| k \log^2(k+1)}, 1 \right) \quad \Longrightarrow \quad \text{Bounded dual} \end{cases}
$$

▷ $L(u) = Auu^*$ & $g(u) = \text{tr}(cuu^*)$

◦ Our result: FOS with $\mathcal{O}\left(\frac{1}{\epsilon^3}\right)$ & SOS $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^5}\right)$ total complexity

Cartis, C., Gould, N. I., and Toint, P. L. "Optimality of orders one to three and beyond: Characterization and evaluation complexity in constrained nonconvex optimization," Journal of Complexity, 2018.

Key assumption for the algorithm

- A novel constraint qualification: A new non-convex Slater's condition
  - ▷ Minimum angle between $T_{\mathcal{U}}(u)$ and null space of L
  - ▷ To bound feasibility gap by gradient mapping.
  - ▷ Related to the classical Mangasarian-Fromovitz constraint qualification
- We verify the condition for the following problems:
  - ▷ Clustering
  - ▷ Generalized eigenvalue
  - ▷ Basis pursuit

D. P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods. Belmont MA: Athena Scientific, originally published by academic press, inc., in 1982 edition, 1996.

Numerical experiment: Clustering

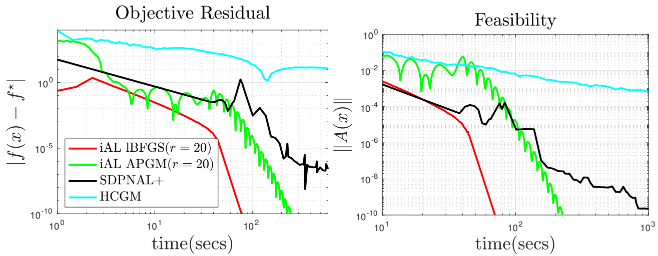
- Model free k-means clustering SDP:

$$\min \left\{ \mathrm{tr}(cx) : \; x1 = 1, \; x \geq 0, \; x \succeq 0, \; x^* = x, \; \mathrm{tr}(x) = \rho \right\},$$

- Nonconvex formulation:

$$\min \left\{ \mathrm{tr}(cuu^*) : \; uu^*1 = 1, \; \underbrace{u \geq 0, \; \|u\|_F \leq \sqrt{\rho}}_{\mathcal{U}} \right\},$$

- Preprocessing & setup & rounding as in (Mixon et. al., 2017)



D.Mixon, S.Villar and R.Ward, Clustering subgaussian mixtures by semidefinite programming, 2017

DARN with GANs - Numerical Results (MNIST)

◦ De-adversarial-noise with generative adversarial networks:

$$\begin{array}{ll} \underset{w,z}{\text{minimize}} & \|w - (w_0 + \eta)\|_\star \\ \text{subject to} & w = G(z), \end{array}$$

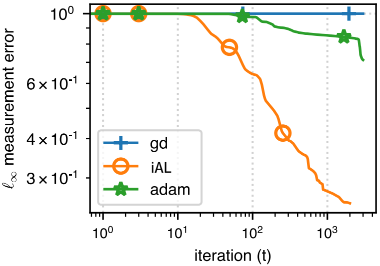


Figure: ℓ_∞ error per iteration

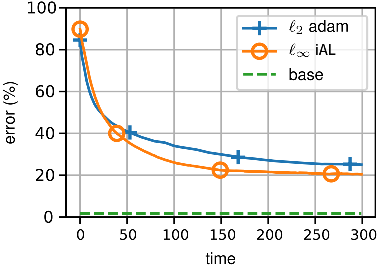


Figure: misclassification error per iteration

Numerical experiment: Basis Pursuit

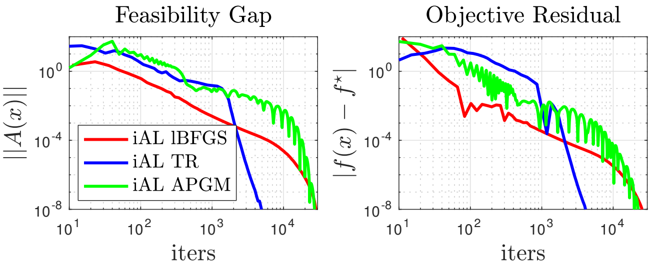
- Convex formulation:

$$\min\left\{\|x\|_1 :\ Ax = b\right\}$$

- Non-convex formulation:

$$\text{change of variables}\begin{cases} x & := x^+ - x^- \\ x^+ & := u_1^{\circ 2},\quad x^- := u_2^{\circ 2} \text{ and } u := [u_1^\top, u_2^\top]^\top \\ \bar{A} & := [A, -A] \end{cases}$$

$$\min\left\{\|u\|_2^2 :\ \bar{A}u^{\circ 2} = b\right\}$$



- Potential with more structured norms *(e.g., latent group lasso norm)*

Linearized augmented Lagrangian method

◦ Our idea: Alternate primal and dual gradient steps in u and y

$$\text{LALM: } \begin{cases} u^+ = \mathcal{P}_{\mathcal{U}}\left(u - \gamma \nabla_u \mathcal{L}_\beta(u, y)\right) \\ y^+ = y + \sigma\left(L(u^+) - b\right) \\ \text{Pick } \gamma^+ < \gamma,\ \beta^+ < \beta \text{ and } \sigma^+ < \sigma \end{cases}$$

$$\text{Update rule: } \begin{cases} \beta^+ = \beta\sqrt{\frac{(k-1)\log^2(k)}{k\log^2(k+1)}} \\ \sigma^+ = \sigma_0 \min\left(\frac{1}{\beta^+} - \frac{1}{\beta}, \frac{1}{||L(u)-b||k\log^2(k+1)}\right) \end{cases}$$

$$\text{Convergence: } \begin{cases} \min_u \|G_\gamma(u)\|^2 = \left\|\frac{u^+-u}{\gamma}\right\|^2 = \mathcal{O}\left(\frac{1}{k^{1/2}}\right) \\ \min_u \|Lu - b\| = \mathcal{O}\left(\frac{1}{k^{1/2}}\right) \end{cases}$$

◦ Best theoretical rates in the literature

◦ Alternating direction method-of-multipliers extension

Linearized alternating direction method of multiples

$$\text{LADMM:} \begin{cases} u^+ = \mathcal{P}_{\mathcal{U}}\left(u - \gamma \nabla_u \mathcal{L}_\beta(u, v, y)\right) \\ v^+ = \mathcal{P}_{\mathcal{V}}\left(v - \iota \nabla_v \mathcal{L}_\beta(u^+, v, y)\right) \\ y^+ = y + \sigma\left(A(u^+) + B(v^+) - b\right) \\ \text{Pick } \gamma^+ < \gamma,\ \iota^+ < \iota,\ \beta^+ < \beta \text{ and } \sigma^+ < \sigma \end{cases}$$

$$\text{Update rule:} \begin{cases} \beta^+ = \beta\sqrt{\frac{(k-1)\log^2(k)}{k\log^2(k+1)}} \\ \sigma^+ = \sigma_0 \min\left(\frac{1}{\beta^+} - \frac{1}{\beta}, \frac{1}{||A(u)+B(v)-b||k\log^2(k+1)}\right) \end{cases}$$

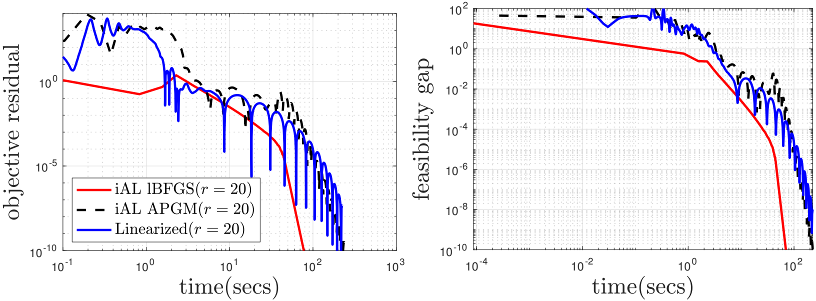
$$\text{Convergence:} \begin{cases} \min_u \|G_\gamma(u)\|^2 = \left\|\frac{u^+-u}{\gamma}\right\|^2 = \mathcal{O}\left(\frac{1}{k^{1/2}}\right) \\ \min_v \|H_\iota(u)\|^2 = \left\|\frac{v^+-v}{\iota}\right\|^2 = \mathcal{O}\left(\frac{1}{k^{1/2}}\right) \\ \min_u \|A(u) + B(v) - b\| = \mathcal{O}\left(\frac{1}{k^{1/2}}\right) \end{cases}$$

◦ Same rates as the linearized augmented Lagrangian

Numerical experiment: Clustering

$$\min \left\{ \mathrm{tr}(cuu^*) : \; uu^*1 = 1, \; \underbrace{u \geq 0, \; \|u\|_F \leq \sqrt{\rho}}_{\mathcal{U}} \right\},$$

◦ Preprocessing & setup & rounding as in (Mixon et. al., 2017)



Summary

- Potentially huge benefits from the non-convex approach
- Stationary points of the augmented Lagragian vs. original problem
- Major issue: Next!

Burer-Monteiro factorization is not storage optimal

(Waldspuger & Walters, Theorem 2)

◦ Suppose that the feasible set of SDP contains a rank-1 matrix.

◦ Suppose that the factorization rank satisfies

$$\binom{r+1}{2} + r \leq n.$$

◦ Then,* there is a set of cost matrices c with positive Lebesque measure for which

▷ Original SDP has a unique rank-1 minimum.

▷ Factorized SDP has a unique rank-1 global minimum.

▷ Factorized SDP has at least one suboptimal local minimum.

Solution rank is one, optimal storage is $\Theta(n+p)$,
but the Factorized SDP requires $\Omega(p\sqrt{n})$ storage only for decision variable.

*under a mild technical condition.

Waldspuger & Walters. Rank optimality for the Burer-Monteiro factorization, 2018.

Storage optimality

◦ 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?

`Definition.`

An algorithm for the model problem has optimal storage

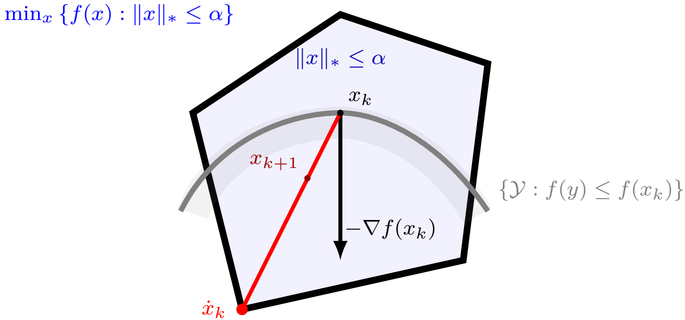
if its working storage is $\Theta(n + rp)$.

Storage-optimal & scalable SDP solutions

- Sketchy decisions [AISTATS 2017]
  - new convex rank-1 streaming models:
    - Universal primal-dual method [NIPS 2015]
    - Universality through learning rate adaptation [NIPS 2018]
    - Homotopy-CGM: A penalty approach [ICML 2018]
    - Stochastic-HCGM: Extension to stochastic setting [LoYFC, under review]
    - CGAL: An augmented Lagrangian framework [YFC, under review]
  - space optimal sketch objects:
    - Bilateral [SIMAX 2017]
    - Nyström [NIPS 2017]
    - Trilateral [TYUC, under review]

† Volkan [C]evher | Olivier [F]ercoq | Kfir [L]evy | Francesco [Lo]catello | Joel [T]ropp | Madeleine [U]dell | Alp [Y]urtsever

The conditional gradient method (CGM)



For $k = 0$ **to** $k_{\max}$:

$$\dot{x}_k = \arg\max_{y:\|y\|_* \leq \alpha} \langle -\nabla f(x_k), y \rangle$$

$$x_{k+1} = (1 - \eta_k)x_k + \eta_k \dot{x}_k$$

End for

$$\Longrightarrow$$

For $k = 0$ **to** $k_{\max}$:

$$(u_k, v_k) = \texttt{MaxSingVec}(\nabla f(x_k))$$

$$x_{k+1} = (1 - \eta_k)x_k - \eta_k \alpha u_k v_k^*$$

End for

A storage-optimal framework: SketchyCGM

◦ Model problem: $\min_x \{f(Ax) : \|x\|_* \leq \alpha\}$

CGM

For $k = 0$ **to** $k_{\max}$:

$(u_k, v_k) = \texttt{MaxSingVec}(A^* \nabla f(Ax_k))$

$x_{k+1} = (1 - \eta_k) x_k - \eta_k \alpha u_k v_k^*$

End for

SketchyCGM

For $k = 0$ **to** $k_{\max}$:

$(u_k, v_k) = \texttt{MaxSingVec}(A^* \nabla f(z_k))$

$z_{k+1} = (1 - \eta_k) z_k - \eta_k \alpha \mathcal{A} u_k v_k^*$

$\mathcal{S} x_{k+1} = (1 - \eta_k) \mathcal{S} x_k - \eta_k \alpha \mathcal{S} u_k v_k^*$

End for

$\hat{x} = \text{SketchRecover}(\mathcal{S} x_k, r)$

◦ Drive iterations by the dual component $z = Ax$

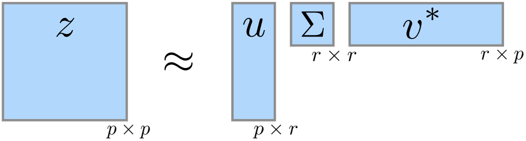
◦ Primal weighting via the streaming model

◦ Sketch & recover low-rank primal-solutions with guarantees

Yurtsever, Tropp, Udell, Cevher. Sketchy decisions: Convex low-rank matrix optimization with optimal storage, AISTATS
Tropp, Yurtsever, Udell, Cevher. Practical sketching algorithms for low-rank matrix approximation, SIMAX
Tropp, Yurtsever, Udell, Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data, NIPS
Tropp, Yurtsever, Udell, Cevher. More practical sketching algorithms for low-rank matrix approximation, under review

Brief detour: The bilateral sketch

- Let $z \in \mathbb{R}^{p \times p}$ be a large input matrix (approximately low-rank)



- Fix sketch size parameters (t, s) with $t, s \ll p$
- Draw independent random matrices $\Omega \in \mathbb{R}^{p \times t}$ and $\Psi \in \mathbb{R}^{s \times p}$

Range sketch: $y = z\Omega \in R^{p \times t}$

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} \quad & z & \quad \end{bmatrix} \begin{bmatrix} \Omega \end{bmatrix}$$

Co-range sketch: $w = \Psi z \in R^{s \times p}$

$$\begin{bmatrix} \quad & w & \quad \end{bmatrix} = \begin{bmatrix} \quad & \Psi & \quad \end{bmatrix} \begin{bmatrix} \quad & z & \quad \end{bmatrix}$$

Brief detour: Analysis of bilateral sketch

◦ Rigorous error bound from [SIMAX 2017]

▷ draw independent standard normal test matrices Ω and Ψ

▷ denote rank-t approximation produced by our single-view method by $\hat{z}$

▷ Then, for $t > r + 1$ and $s > t + 1$,

$$\mathbb{E}\|z - \hat{z}\|_F^2 \quad \le \quad \frac{t}{t-r-1}\,\frac{s}{s-t-1} \min_{\mathrm{rank}(x)\le r} \|z - x\|_F^2$$

▷ In particular, if $t = 2r + 1$ and $s = 4r + 2$,

$$\boxed{\mathbb{E}\|z - \hat{z}\|_F^2 \quad \le \quad 2 \min_{\mathrm{rank}(x)\le r} \|z - x\|_F^2}$$

Brief detour: Analysis of bilateral sketch

- Rigorous error bound from [SIMAX 2017]
  - ▷ draw independent standard normal test matrices Ω and Ψ
  - ▷ denote rank-t approximation produced by our single-view method by $\hat{z}$
  - ▷ Then, for $t > r + 1$ and $s > t + 1$,

$$\mathbb{E}\|z - \hat{z}\|_F^2 \quad \leq \quad \frac{t}{t-r-1}\,\frac{s}{s-t-1}\,\min_{\mathrm{rank}(x)\leq r}\|z - x\|_F^2$$

  - ▷ In particular, if $t = 2r + 1$ and $s = 4r + 2$,

$$\boxed{\mathbb{E}\|z - \hat{z}\|_F^2 \quad \leq \quad 2\min_{\mathrm{rank}(x)\leq r}\|z - x\|_F^2}$$

More on sketching:

Nyström [NIPS 2017] and Trilateral [under review] sketch
with performance improvements & storage efficiencies

Towards SDP's: A primal-dual formulation

◦ Consider the following auxiliary template

$$\min_{x} \left\{ \mathrm{tr}(cx) : Ax = b, x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Primal-SDP)}$$

▷ add a redundant trace constraint

◦ Write down the dual formulation

$$\underbrace{\max_{y} \min_{x} \left\{ \underbrace{tr(cx) + y^*(Ax - b) : \ x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho}_{:=\mathcal{L}(x,y)} \right\}}_{:=d(y)} \qquad \text{(Dual-SDP)}$$

▷ Lagrangian $\mathcal{L}(x, y)$

▷ dual function $d(y) := \min_{x} \mathcal{L}(x, y)$ $\Longrightarrow$ bounded subgradients!

A universal primal-dual (sub)gradient method

$$\underbrace{\max_y \min_x \left\{ \underbrace{tr(cx) + y^*(Ax - b) : \; x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho}_{:=\mathcal{L}(x,y)} \right\}}_{:=d(y)} \qquad \text{(Dual-SDP)}$$

- Subgradient of $d(y)$ can be computed as

$$\begin{aligned} u &= \texttt{MaxEigVec}(c + A^* y) \\ \dot{x} &= \rho u u^* \\ \nabla d(y) &= A\dot{x} - b \end{aligned}$$

- Universal primal-dual method (UPD) [NIPS 2015]
  - ▷ solve Dual-SDP with a subgradient method
  - ▷ recover primal as a weighted average of $\dot{x}_k$

Efficiency considerations for the dual problem

- Iteration complexity lower-bounds:
  - $\mathcal{O}(1/\varepsilon^2)$ for non-smooth problems (subgradients G-bounded) $\leftarrow$ Dual-SDP
  - $\mathcal{O}(1/\sqrt{\varepsilon})$ for L-smooth problems

- Adaptation to (local)-smoothness via line-search
  - requires an ascent (descent) lemma

$$\text{smooth} \implies d(y + \alpha \nabla d(y)) \geq d(y) + \frac{\alpha}{2}\|\nabla d(y)\|^2 \qquad (\forall \alpha \leq \frac{1}{L})$$

$$\text{non-smooth} \implies d(y + \alpha \nabla d(y)) \geq d(y) + \frac{\alpha}{2}\|\nabla d(y)\|^2 - \frac{\varepsilon}{2} \qquad (\forall \alpha \leq \frac{\varepsilon}{G^2})$$

UPD: The algorithm

$$\max_y \min_x \left\{ tr(cx) + y^*(Ax - b) : \; x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Dual-SDP)}$$

UPD

For $k = 0$ **to** $k_{\max}$:

$u_k = \texttt{MaxEigVec}(c + A^* y_k)$

$\dot{x}_k = \rho u_k u_k^*$ and $\nabla d_k = A\dot{x}_k - b$

$\alpha_k = 2\alpha_{k-1}$

While $d(y_k + \alpha_k \nabla d_k) < d(y_k) + \frac{\alpha_k}{2}\|\nabla d_k\|^2 - \frac{\varepsilon}{2}$

$\alpha_k = \alpha_k / 2$

End while

$y_{k+1} = y_k + \alpha_k \nabla d_k$

$\eta_k = \alpha_k / \sum_{i=0}^{k} \alpha_i$

$\bar{x}_{k+1} = (1 - \eta_k)\bar{x}_k + \eta_k \dot{x}_k$

End for

$$\Rightarrow |f(x_k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) + \frac{\varepsilon}{2} \quad \& \quad \|Ax_k - b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

UPD: The algorithm

$$\max_y \min_x \left\{ tr(cx) + y^*(Ax - b) : \; x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Dual-SDP)}$$

UPD

For $k = 0$ **to** $k_{\max}$:

$u_k = \texttt{MaxEigVec}(c + A^* y_k)$

$\dot{x}_k = \rho u_k u_k^*$ and $\nabla d_k = A\dot{x}_k - b$

$\alpha_k = 2\alpha_{k-1}$

While $d(y_k + \alpha_k \nabla d_k) < d(y_k) + \frac{\alpha_k}{2}\|\nabla d_k\|^2 - \frac{\varepsilon}{2}$

$\alpha_k = \alpha_k / 2$

End while

$y_{k+1} = y_k + \alpha_k \nabla d_k$

$\eta_k = \alpha_k / \sum_{i=0}^{k} \alpha_i$

$\bar{x}_{k+1} = (1 - \eta_k)\bar{x}_k + \eta_k \dot{x}_k$

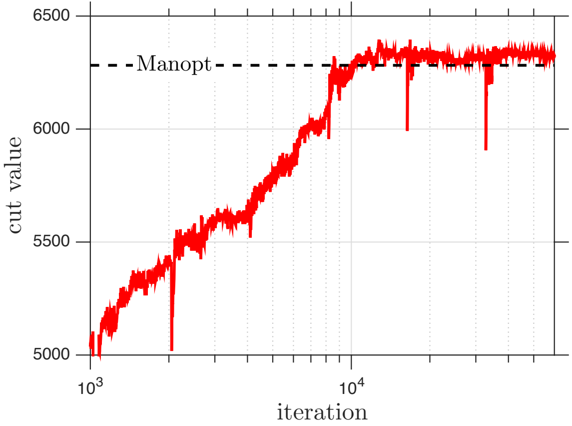
End for

there is also accelerated version AUPD

$$\Rightarrow |f(x_k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) + \frac{\varepsilon}{2} \quad \& \quad \|Ax_k - b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

Numerical experiment: Max-cut

- G67 - sparse graph with $p = 10{,}000$ nodes



Solved by SketchyUPD (with tail averaging): $\epsilon = 100$ and $r = 20$.

Universality through learning rate adaptation

- UPD & AUPD based on a line-search strategy
  - ✓ adapts to the smoothness
  - × requires ε to be set in advance
  - × line-search condition requires exact oracles
  - × converges to $\frac{\varepsilon}{2}$-suboptimal solution
  - ✓ does not require a bound on the dual solution norm

- A new adaptive dual space approach based on online learning
  - ✓ adapts to the smoothness
  - ✓ does not require ε to be set
  - ✓ can work with stochastic oracles $\Longrightarrow$ more robust!
  - ✓ converges to the true solution
  - × requires a bound on the dual solution norm (doubling trick works in practice)

Online to batch conversion

◦ Consider concave (possibly non-smooth) maximization

$$\max_{y \in \mathcal{Y}} \; d(y)$$

▷ $\mathcal{Y}$ is a bounded convex set

▷ $D = \max_{y,z \in \mathcal{Y}} \|y - z\|$

AdaGrad

For $k = 0$ **to** $k_{\max}$:

$$\left.\begin{aligned} y_{k+1} &= \mathcal{P}_{\mathcal{Y}}(y_k + \eta_k \nabla d(y_k)) \\ \bar{y}_{k+1} &= (1 - \tfrac{1}{k})\bar{y}_k + \tfrac{1}{k} y_k \end{aligned}\right\} \text{ with } \; \eta_k = D\left(2\sum_{\tau=0}^{k} \|\nabla d(y_k)\|^2\right)^{-\frac{1}{2}}$$

End for

◦ Convergence rates

▷ smooth $\quad d(\bar{y}_k) - d^\star = \mathcal{O}(\frac{1}{k})$

▷ non-smooth $\quad d(\bar{y}_k) - d^\star = \mathcal{O}(\frac{1}{\sqrt{k}})$

▷ stochastic $\quad \mathbb{E}[d(\bar{y}_k)] - d^\star = \mathcal{O}(\frac{1}{\sqrt{k}})$

New [LYC, NIPS2018]:

AdaGrad makes use of smoothness and small variance (σ^2) in stochastic setting

$$\mathbb{E}[d(\bar{y}_k)] - d^\star = \mathcal{O}\left(\frac{LD^2}{k} + \frac{\sigma D}{\sqrt{k}}\right)$$

Our accelerated adaptive gradient method (AcceleGrad)

- Consider concave (possibly non-smooth) maximization

$$\max_{y \in \mathcal{Y}} \; d(y)$$

  - ▷ $\mathcal{Y}$ is a bounded convex set
  - ▷ $D = \max_{y,z \in \mathcal{Y}} \|y - z\|$
  - ▷ G is bound on (sub)gradients

AcceleGrad [LYC, NIPS2018]

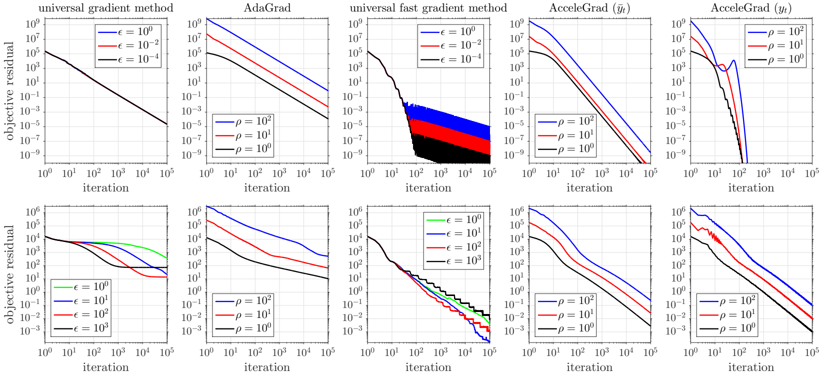
For $k = 0$ **to** $k_{\max}$:

$$\left.\begin{aligned}
x_{k+1} &= \tau_k z_k + (1 - \tau_k) y_k \\
z_{k+1} &= \mathcal{P}_{\mathcal{Y}}(z_k + \alpha_k \eta_k \nabla d(y_k)) \\
y_{k+1} &= x_{k+1} + \eta_k \nabla d(y_k) \\
\omega_k &= \alpha_k \Big/ \textstyle\sum_{\tau=0}^{k} \alpha_\tau \\
\bar{y}_{k+1} &= (1 - \omega_k)\bar{y}_k + \omega_k y_k
\end{aligned}\right\} \quad \text{with} \quad \begin{aligned}
\alpha_k &= \begin{cases} 1 & 0 \le k \le 2 \\ \frac{k+1}{4} & k \ge 3 \end{cases} \\
\eta_k &= 2D\left(G^2 + 2\sum_{\tau=0}^{k} \alpha_\tau^2 \|\nabla d(y_k)\|^2\right)^{-\frac{1}{2}}
\end{aligned}$$

End for

- Convergence rates
  - ▷ smooth $\quad d(\bar{y}_k) - d^\star = \mathcal{O}(\frac{1}{k^2})$
  - ▷ non-smooth $\quad d(\bar{y}_k) - d^\star = \mathcal{O}(\frac{1}{\sqrt{k}})$
  - ▷ stochastic $\quad \mathbb{E}[d(\bar{y}_k)] - d^\star = \tilde{\mathcal{O}}(\frac{1}{\sqrt{k}})$

An empirical comparison



smooth (top) $\quad \min_{\|x\|_2 \le D} \|Ax - b\|_2^2$

nonsmooth (top) $\quad \min_{\|x\|_2 \le D} \|Ax - b\|_1$

$A \in \mathbb{R}^{m \times p}$ and $x^\natural \in \mathbb{R}^p$ std. Gaussian

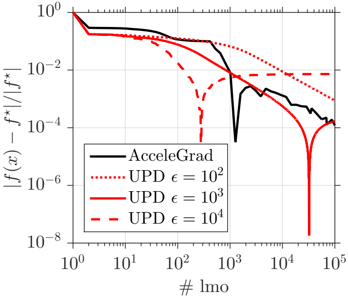
$b = Ax^\natural + \omega, \quad \omega \sim \mathcal{N}(0, \sigma^2),$

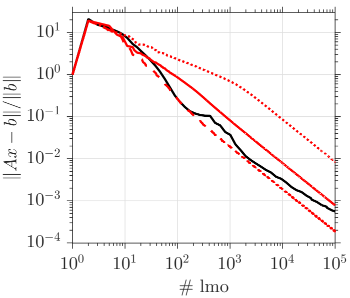
$m = 2000, \ p = 500$ and $\sigma^2 = 10^{-2}$

Numerical experiment: Max-cut

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \underbrace{\frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij})}_{\mathrm{tr}(cx)} : \underbrace{\mathrm{diag}(x) = 1}_{A(x)=b},\ x \succeq 0,\ x^* = x \right\}$$

○ UF sparse matrix collection: G1 dataset (800×800)





From dual to primal via a new penalty method: HCGM

- Consider the penalized problem

$$\min_{x} \left\{ \underbrace{\mathrm{tr}(cx) + \frac{1}{2\beta}\|Ax - b\|^2}_{=: f_\beta(x)} : x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Smoothed-SDP)}$$

  - ▷ penalized objective $f_\beta(x)$

- Homotopy-CGM:
  - ▷ CGM: $\mathcal{O}\left(\frac{1}{k}\right)$ rate on $f_\beta(x)$ when β is fixed
  - ▷ update $\beta_k = \beta_0/\sqrt{k}$ at every iteration

- Analysis:
  - ▷ $f_{\beta_k}(x_k) - f^\star = \mathcal{O}(1/\sqrt{k})$ & $\beta_k = \beta_0/\sqrt{k}$
  - ▷ smoothed gap reduction lemma [SIOPT 2018]

  $\Longrightarrow$ $|f(x_k) - f^\star| = \mathcal{O}(\frac{1}{\sqrt{k}})$ & $\|Ax - b\| = \mathcal{O}(\frac{1}{\sqrt{k}})$

  - ▷ also works with inexact oracle $\Longrightarrow$ EIGS can be computed in relative accuracy

HCGM: The algorithm

$$\min_x \left\{ \underbrace{\mathrm{tr}(cx) + \frac{1}{2\beta}\|Ax - b\|^2}_{=:f_\beta(x)} : x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Smoothed-SDP)}$$

<u>HCGM</u>

For $k = 0$ **to** $k_{\max}$:

$\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$

$\nabla f_{\beta_k} = c + \frac{1}{\beta_k} A^*(Ax_k - b)$

$u_k = \texttt{MaxEigVec}(\nabla f_{\beta_k})$

$\dot{x}_k = \rho u_k u_k^*$

$x_{k+1} = (1 - \eta_k)x_k + \eta_k \dot{x}_k$

End for

$$\Rightarrow |f(x_k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \;\&\; \|Ax - b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

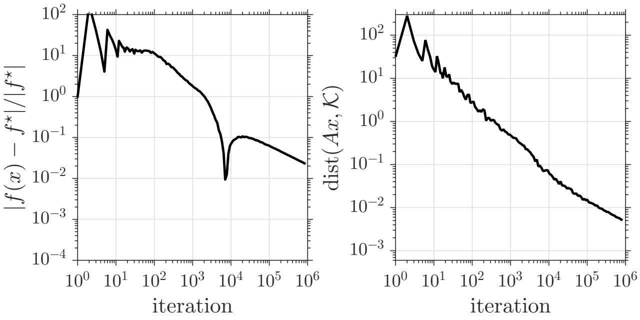
Numerical experiment: Clustering

∘ Model free k-means clustering SDP:

$$\min \left\{ \mathrm{tr}(cx) : \; x1 = 1, \; x \geq 0, \; x \succeq 0, \; x^* = x, \; \mathrm{tr}(x) = \rho \right\},$$

▷ $c \in \mathbb{R}^{p \times p}$: Euclidean distance matrix ($p = 10^3$)

∘ Preprocessing & setup & rounding as in (Mixon et. al., 2017)



(D.Mixon, S.Villar and R.Ward, Clustering subgaussian mixtures by semidefinite programming, 2017)

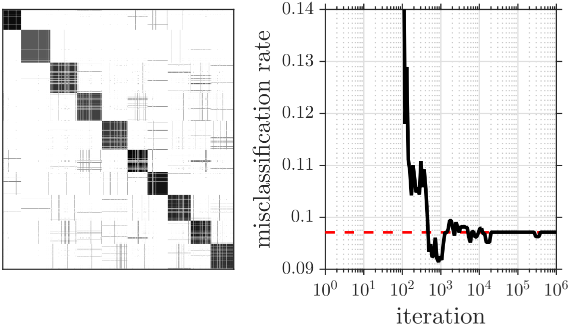
Numerical experiment: Clustering

◦ Model free k-means clustering SDP:

$$\min\left\{\mathrm{tr}(cx):\ x1=1,\ x\geq 0,\ x\succeq 0,\ x^*=x,\ \mathrm{tr}(x)=\rho\right\},$$

▷ $c \in \mathbb{R}^{p\times p}$: Euclidean distance matrix ($p = 10^3$)

◦ Preprocessing & setup & rounding as in (Mixon et. al., 2017)



(D.Mixon, S.Villar and R.Ward, Clustering subgaussian mixtures by semidefinite programming, 2017)

Extension of HCGM for stochastic SDPs: SHCGM

- Consider the penalized problem

$$\min_x \left\{ \mathbb{E}_\Omega \operatorname{tr}(\Omega x) : Ax = b, x \succeq 0, x^* = x, \operatorname{tr}(x) = \rho \right\} \qquad \text{(Stochastic-SDP)}$$

  - Ω denotes stream of data

- Stochastic-HCGM with insights from Mokhtari et al. (2018)
  - Accumulation of gradient estimates: Biased but with decreasing variance
  - No need to increase mini-batch size

SHCGM: The algorithm

$$\min_{x} \left\{ \underbrace{\mathbb{E}_{\Omega}\, \mathrm{tr}(\Omega x)}_{:= f(x,\Omega)} : Ax = b, x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \qquad \text{(Stochastic-SDP)}$$

SHCGM

For $k = 0$ **to** $k_{\max}$:

$\eta_k = \frac{9}{k+8}, \quad \beta_k = \frac{\beta_0}{\sqrt{k+8}}$ and $\rho_k = \frac{4}{(k+7)^{2/3}}$

$d_k = (1 - \rho_k) d_{k-1} + \rho_k \Omega_k$

$v_k = d_k + \frac{1}{\beta_k} A^*(Ax_k - b)$

$u_k = \mathtt{MaxEigVec}(v_k)$

$\dot{x}_k = \rho u_k u_k^*$

$x_{k+1} = (1 - \eta_k) x_k + \eta_k \dot{x}_k$

End for

$\Rightarrow \mathbb{E}|f(x_k, \Omega) - f^\star| = \mathcal{O}\left(\frac{1}{k^{1/3}}\right)$ & $\mathbb{E}\|Ax - b\| = \mathcal{O}\left(\frac{1}{k^{5/12}}\right)$

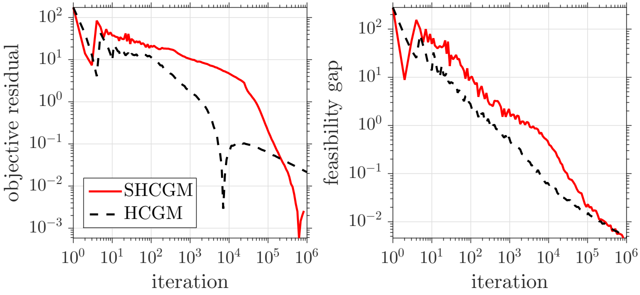
Numerical experiment: Clustering

◦ Model free k-means clustering SDP:

$$\min \left\{ \mathrm{tr}(cx) : \; x1 = 1, \; x \geq 0, \; x \succeq 0, \; x^* = x, \; \mathrm{tr}(x) = \rho \right\},$$

▷ $c \in \mathbb{R}^{p \times p}$: Euclidean distance matrix ($p = 10^3$)

◦ Preprocessing & setup & rounding as in (Mixon et. al., 2017)



SHCGM observes 1% of entries of c at each iteration.

(D.Mixon, S.Villar and R.Ward, Clustering subgaussian mixtures by semidefinite programming, 2017)

An augmented Lagrangian framework: CGAL

- Consider the augmented Lagrangian

$$\min_x \left\{ \underbrace{\underbrace{\mathrm{tr}(cx) + \frac{1}{2\beta}\|Ax-b\|^2}_{=:f_\beta(x)} + y^*(Ax-b)}_{=:\mathcal{L}_\beta(x,y)} : x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \quad \text{(SDP-AL)}$$

  - ▷ penalized objective $f_\beta(x)$
  - ▷ augmented Lagrangian $\mathcal{L}_\beta(x,y)$

- HCGM:
  - ▷ update $\beta_k = \beta_0/\sqrt{k}$ at every iteration

- Conditional gradient with augmented Lagrangian (CGAL):
  - ▷ update y_k and $\beta_k = \beta_0/\sqrt{k}$ at every iteration

CGAL: The algorithm

$$\min_x \left\{ \underbrace{\mathrm{tr}(cx) + \frac{1}{2\beta}\|Ax-b\|^2 + y^*(Ax-b)}_{=:\mathcal{L}_\beta(x,y)} : x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\} \quad \text{(SDP-AL)}$$

CGAL [YFC, ICML 2019]

For $k = 0$ **to** $k_{\max}$:

$\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$

$\nabla_x \mathcal{L}_{\beta_k} = c + \frac{1}{\beta_k} A^*(Ax_k - b) + A^* y_k$

$u_k = \mathtt{MaxEigVec}(\nabla_x \mathcal{L}_{\beta_k})$

$\dot{x}_k = \rho u_k u_k^*$

$x_{k+1} = (1-\eta_k)x_k + \eta_k \dot{x}_k$

$y_{k+1} = y_k + \sigma_k(Ax_{k+1} - b)$

End for

Challenge:

Choice of dual step-size σ_k

$$\Rightarrow |f(x^k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \;\&\; \|Ax-b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

CGAL: Choice of dual step-size

◦ Recurrence + smoothed gap reduction lemma [SIOPT 2018]

$$
\begin{aligned}
\mathcal{L}_{\beta_k}(x_{k+1}, y_{k+1}) - \mathcal{L}^\star \leq (1-\eta_k)\left(\mathcal{L}_{\beta_{k-1}}(x_k, y_k) + \mathcal{L}^\star\right) &- \frac{\eta_k}{2\beta_k}\|Ax_k - b\|^2 \\
&+ \eta_k^2 \frac{\|A\|^2}{2\beta_k}\|\dot{x}_k - x_k\|^2 + \sigma_k \|Ax_{k+1} - b\|^2 \\
&+ \frac{1}{2}(1-\eta_k)\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)\|Ax_k - b\|^2.
\end{aligned}
$$

◦ Choose σ_k

▷ negative terms + positive terms $\sim \quad \eta_k^2 \frac{\|A\|^2}{2\beta_k}\|\dot{x}_k - x_k\|^2$

CGAL: Approximate eigenvector computations

CGAL [YFC, under review]

For $k = 0$ **to** $k_{\max}$:

$\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$

$\nabla_k = c + \frac{1}{\beta_k} A^*(Ax_k - b) + A^* y_k$

$u_k = \texttt{MaxEigVecApprox}(\nabla_k)$ $\quad \Longrightarrow \quad$ $u_k^* \nabla_k u_k \leq \lambda_{\min}(\nabla_k) + \frac{1}{\sqrt{k+1}} \|\nabla_k\|$

$\dot{x}_k = \rho u_k u_k^*$

$x_{k+1} = (1 - \eta_k) x_k + \eta_k \dot{x}_k$

$y_{k+1} = y_k + \sigma_k (A x_{k+1} - b)$

End for

Lanczos algorithm with random start:

$\mathcal{O}(k^{1/4} \log(kp))$ matrix-vector mult.

$$\Rightarrow |f(x^k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \;\&\; \|Ax - b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

$\Rightarrow \mathcal{O}\left(\epsilon^{-2.5} \log \frac{p}{\epsilon}\right)$ matrix-vector mult. $+$ $\mathcal{O}\left((p^2 + d)\epsilon^{-2}\right)$ arithmetic oper.

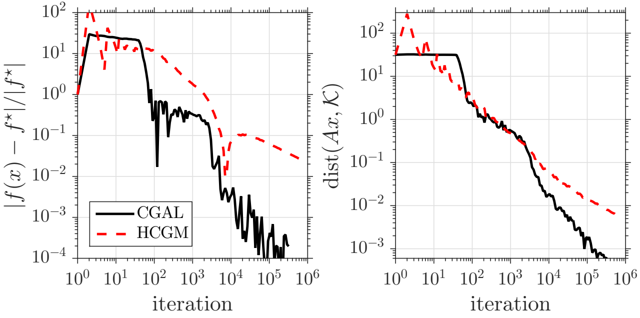
Numerical experiment: Clustering

◦ Model free k-means clustering SDP:

$$\min \left\{ \mathrm{tr}(cx) : \; x1 = 1, \; x \geq 0, \; x \succeq 0, \; x^* = x, \; \mathrm{tr}(x) = \rho \right\},$$

▷ $c \in \mathbb{R}^{p \times p}$: Euclidean distance matrix ($p = 10^3$)

◦ Preprocessing & setup & rounding as in (Mixon et. al., 2017)

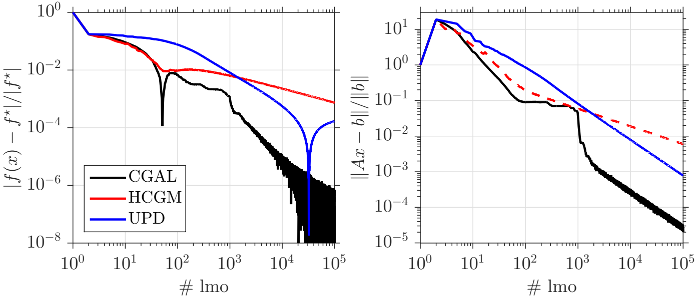


(D.Mixon, S.Villar and R.Ward, Clustering subgaussian mixtures by semidefinite programming, 2017)

Numerical experiment: Max-cut

$$\min_{x\in\mathbb{R}^{p\times p}} \left\{ \underbrace{\frac{1}{2}\sum_{\{i,j\}\in E} c_{ij}(1-x_{ij})}_{\mathrm{tr}(cx)} : \underbrace{\mathrm{diag}(x)=1}_{A(x)=b},\ x\succeq 0,\ x^*=x \right\}$$

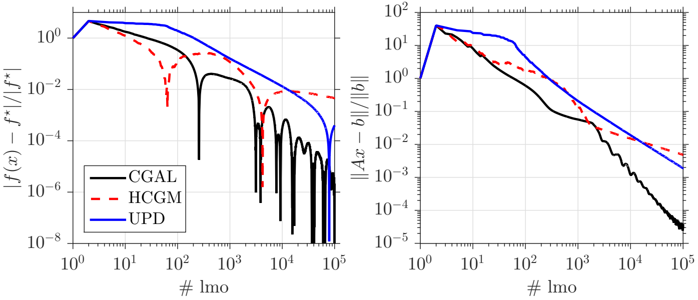
∘ UF sparse matrix collection: G1 dataset (800×800)



Numerical experiment: Max-cut

$$\min_{x \in \mathbb{R}^{p \times p}} \left\{ \underbrace{\frac{1}{2} \sum_{\{i,j\} \in E} c_{ij}(1 - x_{ij})}_{\mathsf{tr}(cx)} : \underbrace{\mathrm{diag}(x) = 1}_{A(x)=b},\ x \succeq 0,\ x^* = x \right\}$$

◦ UF sparse matrix collection: G40 dataset (2000×2000)

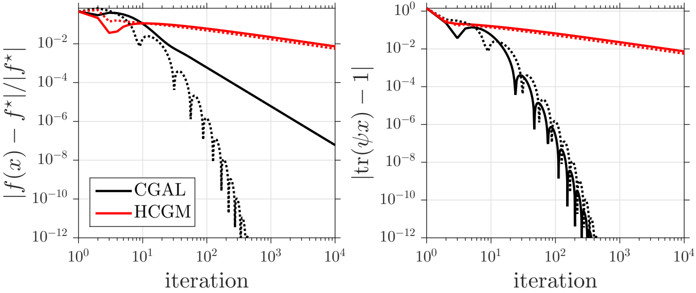


Numerical experiment: Generalized eigenvalue problem

- SDP relaxation:

$$\min \left\{ \mathrm{tr}(\phi x) : \ \mathrm{tr}(\psi x) = 1, \ x \geq 0, \ x \succeq 0, \ x^* = x \right\},$$

- $\psi \sim$ Gaussian iid.
- $\phi \sim$ Gaussian iid. (1000×1000).

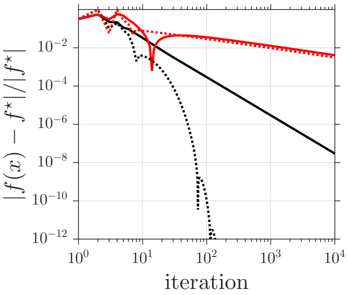


Numerical experiment: Generalized eigenvalue problem

◦ SDP relaxation:

$$\min\left\{\mathrm{tr}(\phi x) : \ \mathrm{tr}(\psi x) = 1, \ x \geq 0, \ x \succeq 0, \ x^* = x\right\},$$

◦ $\psi \sim$ Gaussian iid.
◦ $\phi \sim$ Solution of MaxCut SDP (G40 dataset, 2000×2000).



Other convex methods with streaming rank-1 updates

◦ **[FW-AL]** Frank-Wolfe splitting via augmented Lagrangian (Gidet et al. 2018)

✓ $\mathcal{O}(\frac{1}{k})$ rate in augmented Lagrangian residual

✓ $\mathcal{O}(\frac{1}{\sqrt{k}})$ rate in feasibility gap

× no guarantees on the objective residual

× dual step-size depends on unknown parameters (error bound constant)

◦ **[IAL]** An inexact augmented Lagrangian framework (Liu et al. 2018)

✓ a double loop method, where subproblems are solved by CGM

× many parameters to tune

✓ converges with $\mathcal{O}(\frac{1}{\sqrt{k}})$ rate (outer loop)

× multiple iterations of CGM at each iteration (bounded by $\mathcal{O}(k^2)$)

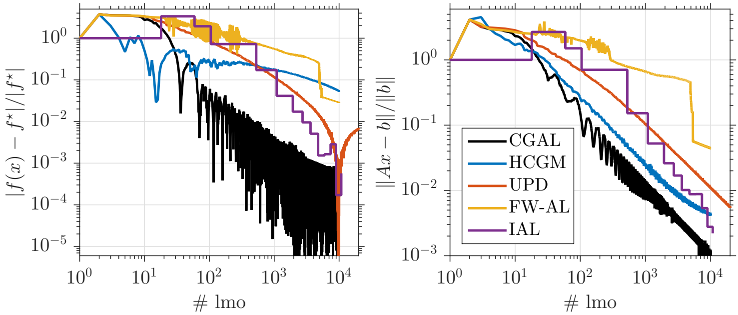
Gidel, G., Pedregosa, F., and Lacoste-Julien, S. Frank-Wolfe splitting via augmented Lagrangian method, AISTATS, 2018.

Liu, Y.-F., Liu, X., and Ma, S. On the non-ergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming, Math. Oper. Res., 2018.

Numerical experiment: Max-cut

$$\min_{x \in \mathbb{R}^{p\times p}} \Big\{ \underbrace{\frac{1}{2}\sum_{\{i,j\}\in E} c_{ij}(1 - x_{ij})}_{\mathrm{tr}(cx)} : \underbrace{\mathrm{diag}(x) = 1}_{A(x)=b},\ x \succeq 0,\ x^* = x \Big\}$$

- UF sparse matrix collection: GD97_b dataset (47×47)



Summary

- Use randomization as a key tool & Sketch the decision variable

- Benefits from convexity can be preserved

- A bonus extension in the sequel: Handling constraints stochastically

Stochastic gradient for almost sure constraints: SASC

◦ Consider the following problem

$$\min_{x \in \mathbb{R}^d} \{P(x) := \mathbb{E} f(x, \xi) + h(x)\}$$

$$A(\xi)x = b(\xi) \quad \xi\text{-almost surely}$$

◦ Penalized problem:

$$\min_{x \in \mathbb{R}^d} \{P_\beta(x) := \mathbb{E} f(x, \xi) + h(x) + \frac{1}{2\beta} \mathbb{E} \|A(\xi)x - b(\xi)\|^2\}$$

◦ Idea: Apply SPG updates to penalized problem.

◦ Challenge: Variance and Lipschitz constant of penalty term are unbounded as $\beta \to 0$.

◦ Technique: Homotopy on the penalty parameter.

Olivier Fercoq, Ahmet Alacaoglu, Ion Necoara, and VC, Almost surely constrained convex optimization, ICML 2019.

SASC: The algorithm

$$\min_{x\in\mathbb{R}^d}\{P(x) := \mathbb{E}f(x,\xi) + h(x)\}$$

$$A(\xi)x = b(\xi) \quad \xi\text{-almost surely}$$

SASC

For $s = 0$ **to** $s_{\max}$:

$m_s = \lfloor m_0\omega^s \rfloor$, $\alpha_s = \alpha_0\omega^{-s/2}$ and $\beta_s = 4\alpha_s \sup_\xi \|A(\xi)\|^2$

For $k = 0$ **to** $m_s - 1$:

Draw ξ.

$D(x_k^s, \xi) = \nabla f(x,\xi) + \frac{1}{\beta_s} A(\xi)^\top (A(\xi)x_k^s - b(\xi))$.

$x_{k+1}^s = \mathrm{prox}_{\alpha_s h}(x_k^s - \alpha_s D(x_k^s, \xi))$

End for

$x_0^{s+1} = x_{m_s}^s$.

$\bar{x}^s = \frac{1}{m_s}\sum_{k=1}^{m_s} x_k^s$.

$M_s = \sum_{i=0}^{s} m_i$.

End for

General convex $\Rightarrow$ $\mathbb{E}|P(\bar{x}^s) - P^\star| = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{M_s}}\right)$ & $\mathbb{E}\|A(\xi)\bar{x}^s - b(\xi)\| = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{M_s}}\right)$

Restricted strongly convex $\Rightarrow$ $\mathbb{E}|P(\bar{x}^s) - P^\star| = \tilde{\mathcal{O}}\left(\frac{1}{M_s}\right)$ & $\mathbb{E}\|A(\xi)\bar{x}^s - b(\xi)\| = \tilde{\mathcal{O}}\left(\frac{1}{M_s}\right)$

Numerical experiment: Basis pursuit

- Streaming basis pursuit with potentially multiple solutions:

$$\min_{x\in\mathbb{R}^d}\left\{\|x\|_1 : a_\xi^\top x = b_\xi, a.s.\right\},$$

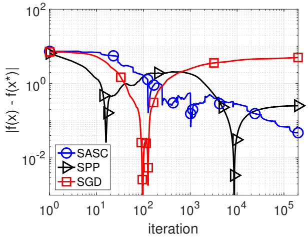
- Synthetic data generation

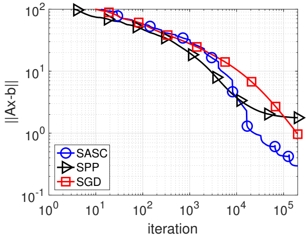
  $\Sigma_{i,j} = \rho^{|i-j|}$, where $\rho = 0.9$.

  $a_i \sim \mathcal{N}(0,\Sigma)$, a_i then centered and normalized.

  $b_i = a_i^\top x^\star$, where $x^\star \in \mathbb{R}^{100}$ is sparse.

  Because of centering,multiple solutions exist to the infinite system $a_\xi^\top x = b_\xi, a.s.$





- SGD does not converge to the sparse solution.