Some New Complexity Results for Composite Optimization

Guanghui (George) Lan

Georgia Institue of Technology

Symposium on Frontiers in Big Data
University of Illinois at Urbana Champaign
September 23, 2016

Background

Big-data Era: In 2012, IBM reported that 2.5 quintillion (10^{18}) bytes of data are created everyday.

- Internet acts as a rich data source, e.g., 2.9 million emails sent every second, 20 hours video uploaded to Youtube every minute.
- Better sensor technology.
- Widespread use of computer simulation.

Opportunities: transform raw data into useful knowledge to support decision-making, e.g., in healthcare, national security, energy and transportation etc.

Machine Learning

Given a set of observed data $S = \{(u_i, v_i)\}_{i=1}^m$, drawn from a certain unknown distribution $\mathcal{D}$ on $U \times V$.

- Goal: to describe the relation between u_i and v_i's for prediction.
- Applications: predicting strokes and seizures, identifying heart failure, stopping credit card fraud, predicting machine failure, identifying spam, ...
- Classic models:
  - Lasso regression: $\min_x \mathbb{E}[(\langle x, u\rangle - v)^2] + \rho\|x\|_1$.
  - Support vector machine: $\min \mathbb{E}_{u,v}[\max\{0, v\langle x, u\rangle] + \rho\|x\|_2^2$.
  - Deep learning: $\min_x \mathbb{E}_{u,v}(F(u,x) - v)^2 + \rho\|Ux\|_1$

Inverse Problems

Given external observations b of a hidden black-box system, to recover the unknown parameters x of the system.

- The relation between b and x, e.g., $Ax = b$, is typically given.
  - However, the system is underdetermined, and b is noisy.
- Applications: medical imaging, locations of oil and mineral deposits, cracks and interfaces within materials.
- Classic models:
  - Total variation minimization: $\min_x \|Ax - b\|^2 + \lambda \mathrm{TV}(x)$.
  - Compressed sensing: $\min_x \|Ax - b\|^2 + \lambda \|x\|_1$.
  - Matrix completion: $\min_x \|Ax - b\|^2 + \lambda \sum_i \sigma_i(x)$.

Composite optimization problems

We consider composite problems which can be modeled as

$$\Psi^* = \min_{x \in X} \{\Psi(x) := f(x) + h(x)\}.$$

Here, $f : X \to \mathbb{R}$ is a smooth and expensive term (data fitting), $h : X \to \mathbb{R}$ is a nonsmooth regularization term (solution structures), and X is a closed convex set.

Much of my previous research

- f given as an expectation or finite-sum.
- f is possibly nonconvex and stochastic.

e.g., mirror descent stochastic approximation (Nemirovski, Juditsky, Lan and Shapiro 07), accelerated stochastic approximation (Lan 08); Nonconvex stochastic gradient descent (Ghadimi and Lan 12)

Complexity for composite optimization

Problem: $\Psi^* := \min_{x \in X} \{\Psi(x) := f(x) + h(x)\}$.

Focus of this talk: h is not necessarily simple

- More solution structural properties, e.g., total variation, group sparsity, and graph-based regularization ...
- Extension: X is not necessarily simple.

First-order methods: iterative methods which operate with the gradients (subgradients) of f and h.

Complexity: number of iterations to find an ϵ-solution, i.e., a point $\bar{x} \in X$ s.t. $\Psi(\bar{x}) - \Psi^* \leq \epsilon$.

Easy case: h simple, X simple

$P_{X,h}(y) := \text{argmin}_{x \in X} \|y - x\|^2 + h(x)$ is easy to compute (e.g., compressed sensing). Complexity: $\mathcal{O}(1/\sqrt{\epsilon})$ (Nesterov 07).

Complexity for composite optimization

Problem: $\Psi^* := \min_{x\in X} \{\Psi(x) := f(x) + h(x)\}$.

Focus of this talk: h is not necessarily simple

- More solution structural properties, e.g., total variation, group sparsity, and graph-based regularization ...
- Extension: X is not necessarily simple.

First-order methods: iterative methods which operate with the gradients (subgradients) of f and h.

Complexity: number of iterations to find an ϵ-solution, i.e., a point $\bar{x} \in X$ s.t. $\Psi(\bar{x}) - \Psi^* \leq \epsilon$.

Easy case: h simple, X simple

$P_{X,h}(y) := \mathrm{argmin}_{x\in X}\|y - x\|^2 + h(x)$ is easy to compute (e.g., compressed sensing). Complexity: $\mathcal{O}(1/\sqrt{\epsilon})$ (Nesterov 07).

More difficult cases

h general, X simple

h is a general nonsmooth function; $P_X := \mathrm{argmin}_{x \in X} \|y - x\|^2$ is easy to compute. Complexity: $\mathcal{O}(1/\epsilon^2)$.

h structured, X simple

h is structured, e.g., $h(x) = \max_{y \in Y} \langle Ax, y \rangle$; P_X is easy to compute. Complexity: $\mathcal{O}(1/\epsilon)$.

h simple, X complicated

$L_{X,h}(y) := \mathrm{argmin}_{x \in X} \langle y, x \rangle + h(x)$ is easy to compute (e.g., matrix completion).Complexity: $\mathcal{O}(1/\epsilon)$.

More difficult cases

h general, X simple

h is a general nonsmooth function; $P_X := \mathrm{argmin}_{x \in X} \|y - x\|^2$ is easy to compute. Complexity: $\mathcal{O}(1/\epsilon^2)$.

h structured, X simple

h is structured, e.g., $h(x) = \max_{y \in Y} \langle Ax, y \rangle$; P_X is easy to compute. Complexity: $\mathcal{O}(1/\epsilon)$.

h simple, X complicated

$L_{X,h}(y) := \mathrm{argmin}_{x \in X} \langle y, x \rangle + h(x)$ is easy to compute (e.g., matrix completion).Complexity: $\mathcal{O}(1/\epsilon)$.

More difficult cases

h general, X simple

h is a general nonsmooth function; $P_X := \operatorname{argmin}_{x \in X} \|y - x\|^2$ is easy to compute. Complexity: $\mathcal{O}(1/\epsilon^2)$.

h structured, X simple

h is structured, e.g., $h(x) = \max_{y \in Y} \langle Ax, y \rangle$; P_X is easy to compute. Complexity: $\mathcal{O}(1/\epsilon)$.

h simple, X complicated

$L_{X,h}(y) := \operatorname{argmin}_{x \in X} \langle y, x \rangle + h(x)$ is easy to compute (e.g., matrix completion).Complexity: $\mathcal{O}(1/\epsilon)$.

Motivation

h simple, X simple	$\mathcal{O}(1/\sqrt{\epsilon})$	100	
h general, X simple	$\mathcal{O}(1/\epsilon^2)$	10^8	
h structured, X simple	$\mathcal{O}(1/\epsilon)$	10^4	
h simple, X complicated	$\mathcal{O}(1/\epsilon)$	10^4	

More general h or more complicated X

$\Downarrow$

Slow convergence of first-order algorithms

$\Downarrow$

A large number of gradient evaluations of ∇f

Motivation

h simple, X simple	$\mathcal{O}(1/\sqrt{\epsilon})$	100	☺
h general, X simple	$\mathcal{O}(1/\epsilon^2)$	10^8	☹
h structured, X simple	$\mathcal{O}(1/\epsilon)$	10^4	☹
h simple, X complicated	$\mathcal{O}(1/\epsilon)$	10^4	☹

More general h or more complicated X

$\Downarrow$

Slow convergence of first-order algorithms

$\not\Downarrow$?

A large number of gradient evaluations of ∇f

Question: Can we skip the computation of ∇f?

Our approach: gradient sliding algorithms

- Gradient sliding: h general, X simple (Lan).
- Accelerated gradient sliding: h structured, X simple (with Yuyuan Ouyang).
- Conditional gradient sliding: h simple, X complicated (with Yi Zhou).

Nonsmooth composite problems

$$\Psi^* = \min_{x \in X} \{\Psi(x) := f(x) + h(x)\}.$$

- f is smooth, i.e., $\exists L > 0$ s.t. $\forall x, y \in X$, $\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$.
- h is nonsmooth, i.e., $\exists M > 0$ s.t. $\forall x, y \in X$, $|h(x) - h(y)| \leq M\|y - x\|$.
- P_X is simple to compute.

Question

How many number of gradient evaluations of ∇f and subgradient evaluations of h' are needed to find an ϵ-solution?

Existing Algorithms

Best-known complexity given by accelerated stochastic approximation (Lan, 12):

$$\mathcal{O}\left\{\sqrt{\frac{L}{\epsilon}} + \frac{M^2}{\epsilon^2}\right\}$$

Issue:

Whenever the second term dominates, the number of gradient evaluations ∇f is given by $\mathcal{O}(1/\epsilon^2)$.

- The computation of ∇f, however, is often the bottleneck.
  - The computation of ∇f invovles a large data set, while that of h' only involves a very sparse matrix.
- Can we reduce the number of gradient evaluations for ∇f from $\mathcal{O}(1/\epsilon^2)$ to $\mathcal{O}(1/\sqrt{\epsilon})$, while still maintaining the optimal $\mathcal{O}(1/\epsilon^2)$ bound on subgradient evaluations for h'?

Review of proximal gradient methods

The model function

Suppose h is relatively simple, e.g., $h(x) = \|x\|_1$.
For a given $x \in X$, let

$$m_\Psi(x,u) := l_f(x,u) + h(u), \quad \forall u \in X,$$
$$l_f(x;y) := f(x) + \langle \nabla f(x), y - x \rangle.$$

Clearly, by the convexity of f,

$$m_\Psi(x,u) \le \Psi(u) \le m_\Psi(x,u) + \tfrac{L}{2}\|u - x\|^2, \quad \forall u \in X.$$

for any $u \in X$

Bregman Distance

Let ω be a strongly convex function with modulus ν and define the Bregman distance $V(x,u) = \omega(u) - \omega(x) - \langle \nabla\omega(x), u - x \rangle$.

$$m_\Psi(x,u) \le \Psi(u) \le m_\Psi(x,u) + \tfrac{L}{\nu} V(x,u), \quad \forall u \in X.$$

Review of proximal gradient descent

$m_\Psi(x,u) = l_f(x,u) + h(u)$ is a good approximation of $\Psi(u)$ when u is "close" enough to x.

Proximal gradient iterations

$$x_k = \text{argmin}_{u\in X}\left\{l_f(x_{k-1},u) + h(u) + \beta_k V(x_{k-1},u)\right\}.$$

Iteration complexity: $\mathcal{O}(1/\epsilon)$.

Accelerated gradient iterations

$$\begin{aligned} \underline{x}_k &= (1-\gamma_k)\bar{x}_{k-1} + \gamma_k x_{k-1},\\ x_k &= \text{argmin}_{u\in X}\left\{\Phi_k(u) := l_f(\underline{x}_k,u) + h(u) + \beta_k V(x_{k-1},u)\right\},\\ \bar{x}_k &= (1-\gamma_k)\bar{x}_{k-1} + \gamma_k x_k. \end{aligned}$$

Iteration complexity: $\mathcal{O}(1/\sqrt{\epsilon})$.

How about a general nonsmooth h?

Old approach: linearizing h (Lan 08, 12)

Iteration Complexity: $\mathcal{O}\left\{\sqrt{\frac{LV(x_0,x^*)}{\epsilon}}+\frac{M^2V(x_0,x^*)}{\epsilon^2}\right\}$.

New approach: gradient sliding

Key idea: keep h in the subproblem, and apply an iterative method to solve the subproblem.
Observation: the subproblem is strongly convex, but nonsmooth, and the strong convexity modulus vanishes.

Challenges

- How accurately to solve the subproblem?
- Do we need to modify the accelerated gradient iterations?

The gradient sliding algorithm

Algorithm 1 The gradient sliding (GS) algorithm

Input: Initial point $x_0 \in X$ and iteration limit N.
Let $\beta_k \ge 0, \gamma_k \ge 0$, and $T_k \ge 0$ be given and set $\bar{x}_0 = x_0$.
for $k = 1, 2, \ldots, N$ **do**
  Set $\underline{x}_k = (1 - \gamma_k)\bar{x}_{k-1} + \gamma_k x_{k-1}$ and $g_k = \nabla f(\underline{x}_k)$.
  Set $(x_k, \tilde{x}_k) = \mathrm{PS}(g_k, x_{k-1}, \beta_k, T_k)$.
  Set $\bar{x}_k = (1 - \gamma_k)\bar{x}_{k-1} + \gamma_k \tilde{x}_k$.
end for
Output: $\bar{x}_N$.

PS: the prox-sliding procedure.

The PS procedure

Procedure $(x^+, \tilde{x}^+) = \mathrm{PS}(g, x, \beta, T)$

Let the parameters $p_t > 0$ and $\theta_t \in [0, 1]$, $t = 1, \ldots$, be given.
Set $u_0 = \tilde{u}_0 = x$.
for $t = 1, 2, \ldots, T$ **do**

$$u_t = \mathrm{argmin}_{u \in X} \langle g + h'(u_{t-1}), u \rangle + \beta[V(x, u) + p_t V(u_{t-1}, u)],$$
$$\tilde{u}_t = (1 - \theta_t)\tilde{u}_{t-1} + \theta_t u_t.$$

end for
Set $x^+ = u_T$ and $\tilde{x}^+ = \tilde{u}_T$.

Note:

$$\begin{aligned} V(x, u) + p_t V(u_{t-1}, u) = {} & (1 + p_t)\omega(u) \\ & -[\omega(x) + \langle \omega'(x), u - x \rangle] \\ & -p_t[\omega(u_{t-1}) + \langle \omega'(u_{t-1}), u - u_{t-1} \rangle]. \end{aligned}$$

Remarks

When supplied with $g(\cdot)$, $x \in X$, β, and T, the PS procedure computes a pair of approximate solutions $(x^+, \tilde{x}^+) \in X \times X$ for the problem of:

$$\mathrm{argmin}_{u \in X} \left\{ \Phi(u) := \langle g, u \rangle + h(u) + \frac{\beta}{2} \|u - x\|^2 \right\}.$$

In each iteration, the subproblem is given by

$$\mathrm{argmin}_{u \in X} \left\{ \Phi_k(u) := \langle \nabla f(\underline{x}_k), u \rangle + h(u) + \frac{\beta_k}{2} \|u - x_k\|^2 \right\}.$$

Convergence of the GS algorithm

Theorem

Suppose that $\{p_t\}$ and $\{\theta_t\}$ in the PS procedure are set to

$$p_t = \frac{t}{2} \quad \text{and} \quad \theta_t = \frac{2(t+1)}{t(t+3)},$$

and that for N given a priori

$$\beta_k = \frac{2L}{k}, \quad \gamma_k = \frac{2}{k+1}, \quad \text{and} \quad T_k = \left\lceil \frac{M^2 N k^2}{\tilde{D} L^2} \right\rceil$$

for some $\tilde{D} > 0$, then

$$\Psi(\bar{x}_N) - \Psi(x^*) \le \frac{L}{\nu N(N+1)} \left(3V(x_0, x^*) + 2\tilde{D} \right).$$

Complexity bounds

- Gradient computation of ∇f: $\mathcal{O}(\sqrt{L/\epsilon})$.
- Sugradient computation of h': $\sum_k T_k = \mathcal{O}(M^2/\epsilon^2)$.

Remark: Do NOT need N given a priori if X is bounded.

Structured convex optimization

Observation: most nonsmooth terms h have certain structures.

Motivating problem: saddle point problem (SPP)

$$\psi^* \equiv \min_{x\in X} \left\{ \psi(x) := f(x) + \max_{y\in Y}\langle Kx, y\rangle - J(y) \right\}.$$

- $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^n$ are closed convex sets
- $0 \le f(x) - l_f(u,x) \le \frac{L}{2}\|x-u\|^2, \ \forall x,u \in X$, where $l_f(u,x) := f(u) + \langle \nabla f(u), x-u\rangle$
- $J(\cdot)$ is convex "simple": the subproblem related to $J(\cdot)$ can be solved efficiently.
- A special case: $Y = \operatorname{dom} J$, i.e., $\min_{x\in X} \psi(x) := f(x) + J^*(Kx)$

Review of Nesterov's Smoothing Scheme (05)

- Approximate ψ by a smooth convex function
$$\psi_\rho^* := \min_{x \in X} \{\psi_\rho(x) := f(x) + h_\rho(x)\},$$
with
$$h_\rho(x) := \max_{y \in Y} \langle Kx, y \rangle - J(y) - \rho W(y_0, y)$$
for some $\rho > 0$, where $y_0 \in Y$ and $W(y_0, \cdot)$ is a strongly convex function.
- By properly choosing ρ and applying the optimal gradient method, one can compute an ε-solution of SPP in at most
$$\mathcal{O}\left(\sqrt{\frac{L}{\varepsilon}} + \frac{\|K\|}{\varepsilon}\right)$$
iterations.

Other related methods for SPP

Nesterov's work has inspired much research to utilize the saddle-point structure.

- Smoothing technique: Auslender and Teboulle (06); Lan, Lu and Monteiro (06); Tseng (08).
- Mirror-prox methods: Nemirovski (04); He, Juditsky and Nemirovski (13); Chen, Lan and Ouyang (14).
- Acclerated prox-level methods: Lan (13); Chen, Lan, Ouyang, and Zhang (14).
- Primal-dual or ADMM: Monteiro and Svaiter (10), He and Yuan (11); Chambolle and Pork (11); Chen, Lan and Ouyang (13); Sun, Luo and Ye (15)...

Some of these methods can achieve exactly the same complexity bound as Nesterov (05).

Significant issues

Bottleneck

The computation of ∇f is often much more expensive than the evaluation of the linear operators K and K^T.

Nesterov's smoothing scheme or related methods

- Gradient evaluations of ∇f: $\mathcal{O}\left(\sqrt{L/\varepsilon} + \|K\|/\varepsilon\right)$.
- Operator evaluations of K and K^T: $\mathcal{O}\left(\sqrt{L/\varepsilon} + \|K\|/\varepsilon\right)$.

The gradient sliding method

- Gradient evaluations of ∇f: $\mathcal{O}\left(\sqrt{L/\varepsilon}\right)$.
- Operator evaluations of K and K^T: $\mathcal{O}\left(\sqrt{L/\varepsilon} + \|K\|^2/\varepsilon^2\right)$.

Open problems and our research

Question

Can we still preserve the optimal $\mathcal{O}(1/\epsilon)$ complexity bound by utilizing only $\mathcal{O}(1/\sqrt{\epsilon})$ gradient computations of ∇f to find an ϵ-solution of SPP?

Our approach:

- Develop new algorithms and complexity bounds for minimizing the summation of two smooth convex functions.
- Apply these results to the smooth approximation of SPP.
- Demonstrate significant savings on gradient computation for both smooth and saddle point problems.

Smooth composite optimization

Problem: $\phi^* := \min_{x \in X} \{\phi(x) := f(x) + h(x)\}$.

$$0 \le f(x) - l_f(u,x) \le L\|x-u\|^2/2, \ \forall x, u \in X$$
$$0 \le h(x) - l_h(u,x) \le L\|x-u\|^2/2, \ \forall x, u \in X$$

Assumption: $M \ge L$.

- Traditional methods assume one can only compute $\nabla \phi$.
- Iteration complexity: $\mathcal{O}(\sqrt{(L+M)/\epsilon})$.
- This bound is optimal in the black-box setting.

Question

Can we gain anything by accessing ∇f and ∇h separately?

Basic ideas of accelerated gradient sliding (AGS)

Idea 1

Inspired by gradient sliding, keep h inside projection (or prox-mapping).

Idea 2

Using a few modified accelerated gradient iterations to solve the prox-mapping

$$\min_{u \in X} g_k(u) + h(u) + \beta V(x_{k-1}, u).$$

Challenges

- How to modify standard accelerated gradient iterations?
- How to analyze these nested accelerated gradient iterations?

The AGS method

Algorithm 2 The accelerated gradient sliding method

Choose $x_0 \in X$. Set $\overline{x}_0 = x_0$.
for $k = 1, \ldots, N$ **do**
  Update $(\underline{x}_k, x_k, \overline{x}_k)$ by

$$\begin{aligned}
\underline{x}_k &= (1-\gamma_k)\overline{x}_{k-1} + \gamma_k x_{k-1}, \\
g_k(\cdot) &= l_f(\underline{x}_k, \cdot), \\
(x_k, \tilde{x}_k) &= ProxAG(g_k, \overline{x}_{k-1}, x_{k-1}, \lambda_k, \beta_k, T_k), \\
\overline{x}_k &= (1-\lambda_k)\overline{x}_{k-1} + \lambda_k \tilde{x}_k.
\end{aligned}$$

end for
Output $\overline{x}_N$.

The ProxAG procedure

$(x^+, \tilde{x}^+) = ProxAG(g, \overline{x}, x, \lambda, \beta, \gamma, T)$

Set $\tilde{u}_0 = \overline{x}$ and $u_0 = x$.

for $t = 1, \ldots, T$ **do**

Update $(\underline{u}_t, u_t, \tilde{u}_t)$ by

$$\begin{aligned} \underline{u}_t &= (1-\lambda)\overline{x} + \lambda(1-\alpha_t)\tilde{u}_{t-1} + \lambda\alpha_t u_{t-1}, \\ u_t &= \operatorname{argmin}_{u \in X} g(u) + l_h(\underline{u}_t, u) + \beta V(x, u) \\ &\qquad + (\beta p_t + q_t) V(u_{t-1}, u), \\ \tilde{u}_t &= (1-\alpha_t)\tilde{u}_{t-1} + \alpha_t u_t, \end{aligned}$$

end for

Output $x^+ = u_T$ and $\tilde{x}^+ = \tilde{u}_T$.

Complexity of AGS

Theorem

Suppose that the parameters of AGS are set to

$$\gamma_k = \frac{2}{k+1}, T_k \equiv T := \left\lceil \sqrt{\frac{M}{L}} \right\rceil, \lambda_k = \begin{cases} 1 & k = 1, \\ \frac{\gamma_k (T+1)(T+2)}{T(T+3)} & k > 1, \end{cases}$$

$$\beta_k = \frac{3L\gamma_k}{\nu k \lambda_k}, \quad \alpha_t = \frac{2}{t+2}, \quad p_t = \frac{t}{2} \text{ and } q_t = \frac{6M}{\nu k(t+1)}.$$

Then

$$\phi(\overline{x}_k) - \phi^* \le \frac{30L}{\nu k(k+1)} V_X(x_0, x^*).$$

- # computations of ∇f: $N = \mathcal{O}\left(\sqrt{L/\varepsilon}\right)$
- # computations of ∇h: $NT = \mathcal{O}\left(\sqrt{M/\varepsilon}\right)$
- For traditional methods, both were $\mathcal{O}\left(\sqrt{(L+M)/\varepsilon}\right)$
- More savings on ∇f if M/L is large.

Application to the saddle point problem

$$\psi^* \equiv \min_{x\in X}\left\{\psi(x) := f(x) + \max_{y\in Y}\langle Kx, y\rangle - J(y)\right\}$$

SPP-A

Let $W(\cdot,\cdot)$ be the prox-function associated with Y with modulus σ and assume $\Omega := \max_{v\in Y} W(y_0, v)$. Define

$$\psi_\rho^* := \min_{x\in X}\left\{\psi_\rho(x) := f(x) + h_\rho(x)\right\},$$
$$h_\rho(x) := \max_{y\in Y}\langle Kx, y\rangle - J(y) - \rho W(y_0, y).$$

Then

$$\psi_\rho(x) \le \psi(x) \le \psi_\rho(x) + \rho\Omega,\ \forall x \in X.$$

- If $\rho = \varepsilon/(2\Omega)$, then an $(\varepsilon/2)$-solution to SPP-A is also an ε-solution to SPP.
- SPP-A is a smooth composite problem with $h(x) = h_\rho(x)$ and $M = \|K\|^2/(\rho\sigma)$.

New complexity for saddle point optimization

Theorem

Let $\varepsilon > 0$ be given and assume that $2\|K\|^2\Omega > \varepsilon\omega L$. If we apply the AGS method SPP-A (with $h = h_\rho$ and $\rho = \varepsilon/(2\sigma)$), then the total number of gradient evaluations of ∇f and linear operator evaluations of K (and K^T) in order to find an ε-solution of SPP can be bounded by

$$\mathcal{O}\left(\sqrt{\frac{LV(x_0,x^*)}{\nu\varepsilon}}\right)$$

and

$$\mathcal{O}\left(\frac{\|K\|\sqrt{V(x_0,x^*)\Omega}}{\sqrt{\nu\sigma}\varepsilon}\right),$$

respectively.

Strongly convex problems

Now suppose that

$$\frac{\mu}{2}\|x-u\|^2 \le f(x) - l_f(u,x) \le \frac{L}{2}\|x-u\|^2, \ \forall x, u \in X.$$

Algorithm 3 The multi-stage AGS algorithm with dynamic smoothing

Choose $v_0 \in X$, accuracy ε, smoothing parameter ρ_0, iteration limit N_0, and initial estimate Δ_0 of SPP s.t. $\psi(v_0) - \psi^* \le \Delta_0$.

for $s = 1, \ldots, S$ **do**

Run the AGS algorithm to problem SPP-A with $\rho = 2^{-s/2}\rho_0$ (where $h = h_\rho$, $x_0 = v_{s-1}$, and $N = N_0$), and let $v_s = \overline{x}_N$.

end for

Output v_S.

New complexity for strongly convex saddle point problems

Theorem

Suppose that $\Omega\|K\|^2 \max\left\{\sqrt{15\Delta_0/\varepsilon}, 1\right\} \geq 2\sigma\Delta_0 L$ *for some given* $\varepsilon > 0$*. If*

$$N_0 = 3\sqrt{\frac{2L}{\nu\mu}},\ \ S = \log_2 \max\left\{\frac{15\Delta_0}{\varepsilon}, 1\right\},\ \ \textit{and}\ \rho_0 = \frac{4\Delta_0}{\Omega 2^{S/2}},$$

then the total number of gradient evaluations of ∇f *and operator evaluations involving* K *and* K^T *can be bounded by*

$$\mathcal{O}\left\{\sqrt{\frac{L}{\nu\mu}} \log \frac{\Delta_0}{\varepsilon}\right\}$$

and

$$\mathcal{O}\left\{\frac{\sqrt{\Omega}\|K\|}{\sqrt{\mu\Delta_0\nu\sigma}}\sqrt{\frac{\Delta_0}{\varepsilon}}\right\},$$

respectively.

Portfolio optimization

Markowitz mean-variance optimal portfolio:

$$\min_{x\in\Delta^n} \phi(x) := x^T(A^T\mathcal{F}A+\mathcal{D})x \quad \text{s.t. } b^Tx \geq \eta,$$

where $\Delta^n := \{x \in \mathbb{R}^n | \sum_{i=1}^n x_i = 1, x_i \geq 0, i = 1, \ldots, n\}$.

A market return model (e.g., Goldfarb and Iyengar 03): $q = b + A^T f + \varepsilon$.

- $q \in \mathbb{R}^n$: random return with mean $b \in \mathbb{R}^n$
- $f \in \mathbb{R}^m$: factors driving the market (e.g., $f \sim N(0, \mathcal{F})$)
- $A \in \mathbb{R}^{m\times n}$: matrix of factor loadings of the n assets
- $\varepsilon \sim N(0, \mathcal{D})$: random vector of residual returns
- The return of portfolio x now follows the distribution $q^Tx \sim N(b^Tx, x^T(A^T\mathcal{F}A + \mathcal{D})x)$

Experimental settings with portfolio optimization

A special case of smooth composite optimization with

$$f(x) = x^T \mathcal{D} x, h(x) = x^T (A^T \mathcal{F} A) x,$$
$$X = \{x \in \Delta^n | b^T x \geq \eta\},$$
$$M = \lambda_{max}(A^T \mathcal{F} A), \text{ and } L = \lambda_{max}(\mathcal{D}).$$

- In practice we have $m < n$
- Consequently, the computational cost for gradient evaluation of ∇f is more expensive than that of ∇h
- The eigenvalues of $\mathcal{D}$ are much smaller than that of $A^T \mathcal{F} A$
- The Lipschitz constants L and M satisfy $L < M$.

Numerical results for portfolio optimization

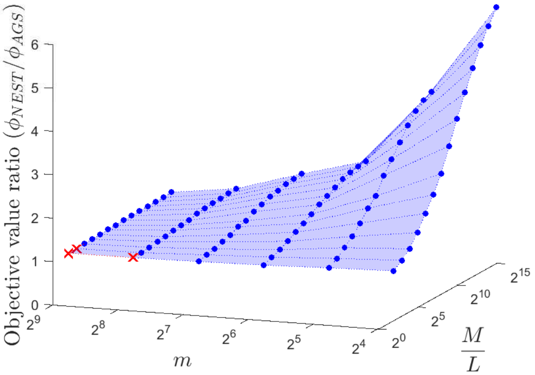

Figure: Ratio of objective values of AGS and NEST in terms of different choices of dimension m and ratio M/L, after running the same amount of CPU time.

Savings on gradient computation

Table: Numbers of gradient evaluations of ∇f and ∇h performed by the AGS method with $M/L = 1024$, after running the same amount of CPU time as 300 iterations of NEST.

m	# ∇f	# ∇h	ϕ_{NEST}/ϕ_{AGS}
16	104	3743	382.5%
32	100	3599	278.6%
64	95	3419	183.3%
128	65	2339	152.8%
256	42	1499	120.1%
512	27	936	104.8%

Savings on gradient computation

Table: Numbers of gradient evaluations of ∇f and ∇h performed by the AGS method with $m = 64$.

M/L	# ∇f	# ∇h	ϕ_{NEST}/ϕ_{AGS}
2^{15}	23	4471	212.5%
2^{14}	31	4327	210.5%
2^{13}	41	4097	206.5%
2^{12}	57	4038	201.6%
2^{11}	72	3648	192.4%
2^{10}	95	3419	183.3%
2^{9}	114	2961	173.3%
2^{8}	143	2698	161.7%
2^{7}	164	2132	150.5%
2^{6}	186	1859	140.1%

Image reconstruction

Total variation (TV) image reconstruction:

$$\min_{x \in \mathbb{R}^n} \left\{ \psi(x) := \tfrac{1}{2}\|Ax - b\|^2 + \eta \|Dx\|_{2,1} \right\}.$$

- $x \in \mathbb{R}^n$: image to be reconstructed
- $\|Dx\|_{2,1}$: TV semi-norm
- D being the finite difference operator
- A: measurement matrix
- b: observed data

Equivalent to:

$$\min_{x \in \mathbb{R}^n} \tfrac{1}{2}\|Ax - b\|^2 + \max_{y \in Y} \eta \langle Dx, y \rangle,$$

$$Y := \{y \in \mathbb{R}^{2n} : \|y\|_{2,\infty} := \max_{i=1,\ldots,n} \|(y^{(2i-1)}, y^{(2i)})^T\|_2 \le 1\}.$$

A special case of SPP

$$f(x) := \tfrac{1}{2}\|Ax - b\|^2, K := \eta D, \text{ and } J(y) \equiv 0,$$

$$L = \lambda_{max}(A^T A) \text{ and } \|K\| = \eta\sqrt{8}.$$

Numerical results for image reconstruction

Table: Numbers of gradient evaluations of ∇f and ∇h performed by the AGS method with ground truth image "Cameraman".

η, ρ	# ∇f	# K	ϕ_{AGS}	ϕ_{NEST}
$\eta = 1, \rho = 10^{-5}$	52	37416	723.8	8803.1
$\eta = 10^{-1}, \rho = 10^{-5}$	173	12728	183.2	2033.5
$\eta = 10^{-2}, \rho = 10^{-5}$	198	1970	27.2	38.3
$\eta = 10^{-1}, \rho = 10^{-7}$	51	36514	190.2	8582.1
$\eta = 10^{-1}, \rho = 10^{-6}$	118	27100	183.2	6255.6
$\eta = 10^{-1}, \rho = 10^{-5}$	173	12728	183.2	2033.5
$\eta = 10^{-1}, \rho = 10^{-4}$	192	4586	183.8	267.2
$\eta = 10^{-1}, \rho = 10^{-3}$	201	2000	190.4	191.2
$\eta = 10^{-1}, \rho = 10^{-2}$	199	794	254.2	254.2

Summary

$$\min_X \{\psi(x) := f(x) + h(x)\}$$

Classes	# iteration	# ∇f
f smooth, h nonsmooth	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(\sqrt{L/\epsilon})$
f smooth, h smooth	$\mathcal{O}(\sqrt{M/\epsilon})$	$\mathcal{O}(\sqrt{L/\epsilon})$
f smooth, h saddle	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(\sqrt{L/\epsilon})$
f strongly convex, h saddle	$\mathcal{O}(\sqrt{1/\epsilon})$	$\mathcal{O}(\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$

- Numerical experiments further confirm these theoretical results.

References

- G. Lan, "Gradient Sliding for Composite Optimization", *Mathematical Programming*, 159 (1), 201-235, 2016.
- G. Lan and Y. Zhou, "Conditional Gradient Sliding for Convex Optimization", *SIAM Journal on Optimization*, 26(2), 1379-1409, 2016.
- G. Lan and Y. Ouyang, "Accelerated Gradient Sliding for Structured Convex Optimization", submitted, 09/2016.

Thanks!