Some New Complexity Results for Composite Optimization

Guanghui (George) Lan

Georgia Institue of Technology
Joint work with Yuyuan Ouyang (Clemson) and Yi Zhou (Georgia Tech)

Workshop on Optimization without Borders, dedicated to Yuri Nesterov's 60th Birthday, Les Houches, France, December 7-12, 2016

General CP methods

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

- X closed and convex.
- Ψ is convex

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

Complexity: the number of (sub)gradient evaluations of Ψ –

- Ψ is smooth: $\mathcal{O}(1/\sqrt{\epsilon})$.
- Ψ is nonsmooth: $\mathcal{O}(1/\epsilon^2)$.
- Ψ is strongly convex: $\mathcal{O}(\log(1/\epsilon))$.

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.

Three Challenging Cases

- h or X are not necessarily simple.
- f given by the summation of many terms.
- f (or h) is possibly nonconvex.

Existing complexity results

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

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

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

Easy case: h simple, X simple

$Pr_{X,h}(y) := \operatorname{argmin}_{x \in X} \|y - x\|^2 + h(x)$ is easy to compute (e.g., compressed sensing). Complexity: $\mathcal{O}(1/\sqrt{\epsilon})$ (Nesterov 07, Tseng 08, Beck and Teboulle 09).

Existing complexity results

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

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

Complexity: number of iterations needed 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

$Pr_{X,h}(y) := \mathrm{argmin}_{x \in X} \|y - x\|^2 + h(x)$ is easy to compute (e.g., compressed sensing). Complexity: $\mathcal{O}(1/\sqrt{\epsilon})$ (Nesterov 07, Tseng 08, Beck and Teboulle 09).

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; $P_X := \text{argmin}_{x\in X}\|y-x\|^2$ 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

$L_{X,h}(y) := \text{argmin}_{x\in X}\langle y, x\rangle + h(x)$ 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: $\mathcal{O}(1/\epsilon)$.

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?

Composite problems

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

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

Existing algorithms evaluate ∇f and h' together at each iteration:

- Mirror-prox method (Juditsky, Nemirovski and Travel, 11):
$$\mathcal{O}\left\{\frac{L}{\epsilon} + \frac{M^2}{\epsilon^2}\right\}$$
- 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 $\mathcal{O}(1/\epsilon^2)$.

Bottleneck for composite problems

- The computation of ∇f, however, is often the bottleneck in comparison with that of h'.
  - The computation of ∇f invovles a large data set, while that of h' only involves a very sparse matrix (e.g., total variation minimization).
- 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'?

The gradient sliding algorithm

Algorithm 1 The gradient sliding (GS) algorithm

Input: Initial point $x_0 \in X$ and iteration limit N.
Let $\beta_k \geq 0, \gamma_k \geq 0$, and $T_k \geq 0$ be given and set $\bar{x}_0 = x_0$.
for $k = 1, 2, \ldots, N$ **do**

1. Set $\underline{x}_k = (1 - \gamma_k)\bar{x}_{k-1} + \gamma_k x_{k-1}$ and $g_k = \nabla f(\underline{x}_k)$.
2. Set $(x_k, \tilde{x}_k) = \mathrm{PS}(g_k, x_{k-1}, \beta_k, T_k)$.
3. 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**
$u_t = \mathrm{argmin}_{u \in X} \langle g + h'(u_{t-1}), u \rangle + \frac{\beta}{2} \|u - x\|^2 + \frac{\beta p_t}{2} \|u - u_{t-1}\|^2,$
$\tilde{u}_t = (1 - \theta_t)\tilde{u}_{t-1} + \theta_t u_t.$
end for
Set $x^+ = u_T$ and $\tilde{x}^+ = \tilde{u}_T$.

Note: $\| \cdot - \cdot \|^2/2$ can be replaced by the more general Bregman distance $V(x, u) = \omega(u) - \omega(x) - \langle \nabla\omega(x), u - x \rangle$.

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 PS proedure

Proposition

If $\{p_t\}$ *and* $\{\theta_t\}$ *in the* PS *procedure satisfy*

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

then for any $t \geq 1$ *and* $u \in X$,

$$\Phi(\tilde{u}_t) - \Phi(u) + \frac{\beta(t+1)(t+2)}{2t(t+3)} \|u_t - u\|^2 \leq \frac{M^2}{\beta(t+3)} + \frac{\beta\|u_0 - u\|^2}{t(t+3)}.$$

Convergence of the GS algorithm

Theorem

Suppose that the previous conditions on $\{p_t\}$ and $\{\theta_t\}$ hold, and that N is given a priori. If

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

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

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

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

Complexity of the GS algorithm

Denote $D_X := \max_{x_1, x_2 \in X} \|x_1 - x_2\|$ and set $\tilde{D} = 3D_X^2/4$.

The number of gradient evaluations of ∇f is bounded by

$$\sqrt{\frac{3LD_X^2}{\epsilon}}$$

and the number of subgradient evaluations of h' is given by $\sum_{k=1}^{N} T_k$, which is bounded by

$$\frac{4M^2D_X^2}{\epsilon^2} + \sqrt{\frac{3LD_X^2}{\epsilon}}.$$

Consequence

Significantly reduce the number of gradient evaluations of ∇f from $\mathcal{O}(1/\epsilon^2)$ to $\mathcal{O}(1/\sqrt{\epsilon})$, even though the whole objective function Ψ is nonsmooth in general.

Extensions

- Gradient sliding for $\min_{x \in X} f(x) + h(x)$:

	total iter.	∇f
h general nonsmooth	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\sqrt{\epsilon})$
h structured nonsmooth	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\sqrt{\epsilon})$
f strongly convex	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(\log(1/\epsilon))$

- Conditional gradient sliding methods for problems with more complicated feasible set.

	total iter. (LO oracle)	∇f
f convex	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\sqrt{\epsilon})$
f strongly convex	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(\log(1/\epsilon))$

The problem of interest

Problem: $\Psi^* := \min_{x \in X} \left\{ \Psi(x) := \sum_{i=1}^m f_i(x) + h(x) + \mu\, \omega(x) \right\}$.

- X closed and convex.
- f_i smooth convex: $\|\nabla f_i(x_1) - \nabla f_i(x_2)\|_* \le L_i \|x_1 - x_2\|$.
- h simple, e.g., l_1 norm.
- ω strongly convex with modulus 1 w.r.t. an arbitrary norm.
- $\mu \ge 0$.
- Subproblem $\mathrm{argmin}_{x \in X} \langle g, x\rangle + h(x) + \mu\, \omega(x)$ is easy.
- Denote $f(x) \equiv \sum_{i=1}^m f_i(x)$ and $L \equiv \sum_{i=1}^m L_i$. f is smooth with Lipschitz constant $\le L$.

Stochastic subgradient descent for nonsmooth problems

- General stochastic programming (SP): $\min_{x \in X} \mathbb{E}_\xi[F(x, \xi)]$.
- Reformulation of the finite sum problem as SP:
  - $\xi \in \{1, \ldots, m\}$, $\mathrm{Prob}\{\xi = i\} = \nu_i$, and $F(x, i) = \nu_i^{-1} f_i(x) + h(x) + \mu\omega(x)$, $i = 1, \ldots, m$.
- Iteration complexity: $\mathcal{O}(1/\epsilon^2)$ or $\mathcal{O}(1/\epsilon)$ ($\mu > 0$).
- Iteration cost: m times cheaper than deterministic first-order methods.
- Save up to a factor of $\mathcal{O}(m)$ subgradient computations.
- For details, see Nemirovski et. al. (09).

Required ∇f_i's in the smooth case

For simplicity, focus on the strongly convex case ($\mu > 0$).
Goal: find a solution $\bar{x} \in X$ s.t. $\|\bar{x} - x^*\| \le \epsilon \|x^0 - x^*\|$.

- Nesterov's optimal method (Nesterov 83):
$$\mathcal{O}\left\{ m\sqrt{\frac{L_f}{\mu}} \log \frac{1}{\epsilon} \right\},$$
- Accelerated stochastic approximation (Lan 12, Ghadimi and Lan 13):
$$\mathcal{O}\left\{ \sqrt{\frac{L_f}{\mu}} \log \frac{1}{\epsilon} + \frac{\sigma^2}{\mu\epsilon} \right\}$$

Note: the optimality of the latter bound for general SP does not preclude more efficient algorithms for the finite-sum problem.

Randomized incremental gradient methods

Each iteration requires a randomly selected $\nabla f_i(x)$.

- Stochastic average gradient (SAG) by Schmidt, Roux and Bach 13:
$$\mathcal{O}\left((m + L/\mu) \log \frac{1}{\epsilon}\right).$$
- Similar results were obtained in Johnson and Zhang 13, Defazio et al. 14...
- Worse dependence on the L/μ than Nesterov's method.

Coordinate ascent in the dual

$\min\left\{\sum_{i=1}^m \phi_i(a_i^T x) + h(x)\right\}$, h strongly convex w.r.t. l_2 norm.

All these coordinate algorithms achieve $\mathcal{O}\left\{m + \sqrt{\frac{mL}{\mu}} \log \frac{1}{\epsilon}\right\}$.

- Shalev-Shwartz and Zhang 13, 15 (restarting stochastic dual ascent),
- Lin, Lu and Xiao, 14 (Nesterov, Fercoq and P. Richtárik's), see also Zhang and Xiao 14 (Chambolle and Pock),
- Dang and Lan 14 (non-strongly convex), $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(1/\sqrt{\epsilon})$.

Some issues:

- Deal with a more special class of problems.
- Require $\mathrm{argmin}\{\langle g, y\rangle + \phi_i^*(y) + \|y\|_*^2\}$, not incremental gradient methods.

Open problems and our research

Problems:

- Can we accelerate the convergence of randomized incremental gradient method?
- What is the best possible performance we can expect?

Our contributions:

- A primal-dual gradient (PDG) method = a primal-dual look to Nesterov's method.
- A randomized PDG (RPDG).
- A new lower complexity bound.
- A game-theoretic interpretation for acceleration.

Catalyst: Lin, Mairal, and Harchaoui 15.

Reformulation and game/economic interpretation

Let J_f be the conjugate function of f. Consider

$$\Psi^* := \min_{x\in X} \left\{ h(x) + \mu\,\omega(x) + \max_{g\in\mathcal{G}} \langle x, g\rangle - J_f(g) \right\}$$

- The buyer purchases products from the supplier.
- The unit price is given by $g \in \mathbb{R}^n$.
- X, h and ω are constraints and other local cost for the buyer.
- The profit of supplier: revenue ($\langle x, g\rangle$) - local cost $J_f(g)$.

How to achieve equilibrium?

Current order quantity x^0, and product price g^0.

Proximity control functions:

$$
\begin{aligned}
P(x^0, x) &:= \omega(x) - [\omega(x^0) + \langle \omega'(x^0), x - x^0 \rangle]. \\
D_f(g_i^0, y_i) &:= J_f(g) - [J_f(g^0) + \langle J_f'(g^0), g - g^0 \rangle].
\end{aligned}
$$

Dual prox-mapping:

$$\mathcal{M}_{\mathcal{G}}(-\tilde{x}, g^0, \tau) := \arg\min_{g \in \mathcal{G}} \{ \langle -\tilde{x}, g \rangle + J_f(g) + \tau D_f(g^0, g) \}.$$

$\tilde{x}$ is the given or predicted demand. Maximize the profit, but not too far away from g^0.

Primal prox-mapping:

$$\mathcal{M}_X(g, x^0, \eta) := \arg\min_{x \in X} \{ \langle g, x \rangle + h(x) + \mu\omega(x) + \eta P(x^0, x) \}.$$

g is the given or predicted price. Minimize the cost, but not too far way from x^0.

The deterministic PDG

Algorithm 2 The primal-dual gradient method

Let $x^0 = x^{-1} \in X$, and the nonnegative parameters $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.
Set $g^0 = \nabla f(x^0)$.
for $t = 1, \ldots, k$ **do**
Update $z^t = (x^t, y^t)$ according to

$$\begin{aligned} \tilde{x}^t &= \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \\ g^t &= \mathcal{M}_{\mathcal{G}}(-\tilde{x}^t, g^{t-1}, \tau_t). \\ x^t &= \mathcal{M}_X(g^t, x^{t-1}, \eta_t). \end{aligned}$$

end for

A game/economic interpretation

- The supplier predicts the buyer's demand based on historical information: $\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}$.
- The supplier seeks to maximize predicted profit, but not too far away from g^{t-1}: $g^t = \mathcal{M}_{\mathcal{G}}(-\tilde{x}^t, g^{t-1}, \tau_t)$.
- The buyer tries to minimize the cost, but not too far away from x^{t-1}: $x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t)$.

PDG in gradient form

Algorithm 3 PDG method in gradient form

Input: Let $x^0 = x^{-1} \in X$, and the nonnegative parameters $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.
Set $\underline{x}^0 = x^0$.
for $t = 1, 2, \ldots, k$ **do**

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}.$$
$$\underline{x}^t = \left(\tilde{x}^t + \tau_t \underline{x}^{t-1}\right) / (1 + \tau_t).$$
$$g^t = \nabla f(\underline{x}^t).$$
$$x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t).$$

end for

Idea: set $J_f'(g^{t-1}) = \underline{x}^{t-1}$.

Relation to Nesterov's method

A variant of Nesterov's method:

$$
\begin{aligned}
\underline{x}^t &= (1-\theta_t)\bar{x}^{t-1} + \theta_t x^{t-1}. \\
x^t &= M_X(\textstyle\sum_{i=1}^m \nabla f_i(\underline{x}^t), x^{t-1}, \eta_t). \\
\bar{x}^t &= (1-\theta_t)\bar{x}^{t-1} + \theta_t x^t.
\end{aligned}
$$

Note that

$$
\underline{x}^t = (1-\theta_t)\underline{x}^{t-1} + (1-\theta_t)\theta_{t-1}(x^{t-1} - x^{t-2}) + \theta_t x^{t-1}.
$$

Equivalent to PDG with $\tau_t = (1-\theta_t)/\theta_t$ and $\alpha_t = \theta_{t-1}(1-\theta_t)/\theta_t$.

Nesterov's acceleration: looking-ahead dual players.
Gradient descent: myopic dual players ($\alpha_t = \tau_t = 0$ in PDG).

Convergence of PDG (or Nesterov's variant)

Theorem

Define $\bar{x}^k := (\sum_{t=1}^k \theta_t)^{-1} \sum_{t=1}^k (\theta_t x^t)$. *Suppose that*
$\tau_t = \sqrt{\frac{2L_f}{\mu}}$, $\eta_t = \sqrt{2L_f\mu}$, $\alpha_t = \alpha \equiv \frac{\sqrt{2L_f/\mu}}{1+\sqrt{2L_f/\mu}}$, *and* $\theta_t = \frac{1}{\alpha^t}$.
Then,

$$\begin{aligned} P(x^k, x^*) &\le \frac{\mu+L_f}{\mu}\alpha^k P(x^0, x^*). \\ \Psi(\bar{x}^k) - \Psi(x^*) &\le \mu(1-\alpha)^{-1}\left[1 + \frac{L_f}{\mu}(2 + \frac{L_f}{\mu})\right]\alpha^k P(x^0, x^*). \end{aligned}$$

Theorem

If $\tau_t = \frac{t-1}{2}$, $\eta_t = \frac{4L_f}{t}$, $\alpha_t = \frac{t-1}{t}$, *and* $\theta_t = t$, *then*
$\Psi(\bar{x}^k) - \Psi(x^*) \le \frac{8L_f}{k(k+1)} P(x^0, x^*)$.

A multi-dual-player reformulation

- Let $J_i : \mathcal{Y}_i \to \mathbb{R}$ be the conjugate functions of f_i and $\mathcal{Y}_i$, $i = 1, \ldots, m$, denote the dual spaces.
  $\min_{x \in X} \left\{ h(x) + \mu\, \omega(x) + \max_{y_i \in \mathcal{Y}_i} \langle x, \sum_i y_i \rangle - \sum_i J_i(y) \right\},$
- Define their new dual prox-functions and dual prox-mappings as

$$
\begin{aligned}
D_i(y_i^0, y_i) \quad &:= \quad J_i(y_i) - [J_i(y_i^0) + \langle J_i'(y_i^0), y_i - y_i^0 \rangle], \\
\mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}, y_i^0, \tau) \quad &:= \quad \arg\min_{y_i \in \mathcal{Y}_i} \left\{ \langle -\tilde{x}, y \rangle + J_i(y_i) + \tau D_i(y_i^0, y_i) \right\}.
\end{aligned}
$$

The RPDG method

Algorithm 4 The RPDG method

Let $x^0 = x^{-1} \in X$, and $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.
Set $y_i^0 = \nabla f_i(x^0)$, $i = 1, \ldots, m$.
for $t = 1, \ldots, k$ **do**
  Choose i_t according to $\mathrm{Prob}\{i_t = i\} = p_i$, $i = 1, \ldots, m$.

$$
\begin{aligned}
\tilde{x}^t &= \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \\
y_i^t &= \begin{cases} \mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}^t, y_i^{t-1}, \tau_t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \\
\tilde{y}_i^t &= \begin{cases} p_i^{-1}(y_i^t - y_i^{t-1}) + y_i^{t-1}, & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \\
x^t &= \mathcal{M}_X(\textstyle\sum_{i=1}^m \tilde{y}_i^t, x^{t-1}, \eta_t).
\end{aligned}
$$

end for

RPDG in gradient form

Algorithm 5 RPDG

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

Choose i_t according to $\mathrm{Prob}\{i_t = i\} = p_i$, $i = 1, \ldots, m$.

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}.$$

$$\underline{x}_i^t = \begin{cases} (1+\tau_t)^{-1}\left(\tilde{x}^t + \tau_t \underline{x}_i^{t-1}\right), & i = i_t, \\ \underline{x}_i^{t-1}, & i \neq i_t. \end{cases}$$

$$y_i^t = \begin{cases} \nabla f_i(\underline{x}_i^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases}$$

$$x^t = \mathcal{M}_X(g^{t-1} + (p_{i_t}^{-1} - 1)(y_{i_t}^t - y_{i_t}^{t-1}), x^{t-1}, \eta_t).$$

$$g^t = g^{t-1} + y_{i_t}^t - y_{i_t}^{t-1}.$$

end for

Game-theoretic interpretation for RPDG

- The suppliers predict the buyer's demand as before.
- Only one randomly selcted supplier will change his/her price, arriving at y^t.
- The buyer would have used y^t as the price, but the algorithm converges slowly (a worse depedence on m) (Dang and Lan 14).
- Add a dual prediction (estimation) step, i.e., $\tilde{y}^t$ s.t. $\mathbb{E}_t[\tilde{y}_i^t] = \hat{y}_i^t$, where $\hat{y}_i^t := \mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}^t, y_i^{t-1}, \tau_i^t)$.
- The buyer uses $\tilde{y}^t$ to determine the order quantity.

Rate of Convergence

Theorem

Let $C = \frac{8L}{\mu}$. *and*

$$
\begin{aligned}
p_i &= \mathrm{Prob}\{i_t = i\} = \frac{1}{2m} + \frac{L_i}{2L}, i = 1, \ldots, m, \\
\tau_t &= \frac{\sqrt{(m-1)^2+4mC}-(m-1)}{2m}, \\
\eta_t &= \frac{\mu\sqrt{(m-1)^2+4mC}+\mu(m-1)}{2}, \\
\alpha_t &= \alpha := 1 - \frac{1}{(m+1)+\sqrt{(m-1)^2+4mC}}.
\end{aligned}
$$

Then

$$
\begin{aligned}
\mathbb{E}[P(x^k, x^*)] &\leq (1 + \frac{3L_f}{\mu})\alpha^k P(x^0, x^*), \\
\mathbb{E}[\Psi(\bar{x}^k)] - \Psi^* &\leq \alpha^{k/2}(1-\alpha)^{-1}\left[\mu + 2L_f + \frac{L_f^2}{\mu}\right] P(x^0, x^*).
\end{aligned}
$$

The iteration complexity of RPGD

- To find a point $\bar{x} \in X$ s.t. $\mathbb{E}[P(\bar{x}, x^*)] \le \epsilon$: $\mathcal{O}\left\{(m + \sqrt{\frac{mL}{\mu}}) \log\left[\frac{P(x^0, x^*)}{\epsilon}\right]\right\}$.
- To find a point $\bar{x} \in X$ s.t. $\mathrm{Prob}\{P(\bar{x}, x^*) \le \epsilon\} \ge 1 - \lambda$ for some $\lambda \in (0, 1)$: $\mathcal{O}\left\{(m + \sqrt{\frac{mL}{\mu}}) \log\left[\frac{P(x^0, x^*)}{\lambda\epsilon}\right]\right\}$.
- Similar results hold for the ergodic sequence in terms of function values.
- A factor of up to $\mathcal{O}\left\{\min\{\sqrt{\frac{L}{\mu}}, \sqrt{m}\}\right\}$ savings on gradient computation (or price changes), at the price of more order transactions.

Lower complexity bound

$$\min_{x_i\in\mathbb{R}^{\tilde n}, i=1,\ldots,m} \left\{\Psi(x) := \sum_{i=1}^m \left[f_i(x_i) + \tfrac{\mu}{2}\|x_i\|_2^2\right]\right\}.$$

$$f_i(x_i) = \frac{\mu(\mathcal{Q}-1)}{4}\left[\tfrac{1}{2}\langle Ax_i, x_i\rangle - \langle e_1, x_i\rangle\right].\ \tilde n \equiv n/m,$$

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & \kappa \end{pmatrix}, \kappa = \frac{\sqrt{\mathcal{Q}}+3}{\sqrt{\mathcal{Q}}+1}.$$

Theorem

Denote $q := (\sqrt{\mathcal{Q}}-1)/(\sqrt{\mathcal{Q}}+1)$. *Then the iterates* $\{x^k\}$ *generated by a randomized incremental gradient method must satisfy* $\frac{\mathbb{E}[\|x^k-x^*\|_2^2]}{\|x^0-x^*\|_2^2} \geq \frac{1}{2}\exp\left(-\frac{4k\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2-4\sqrt{\mathcal{Q}}}\right)$ *for any* $n \geq \underline{n}(m,k) \equiv [m\log\left[\left(1-(1-q^2)/m\right)^k/2\right]]/(2\log q)$.

Complexity

Corollary

The number of gradient evaluations performed by a randomized incremental gradient method for finding a solution $\bar{x} \in X$ *s.t.* $\mathbb{E}[\|\bar{x} - x^*\|_2^2] \le \epsilon$ *cannot be smaller than* $\Omega\left\{\left(\sqrt{m\mathcal{C}} + m\right) \log \frac{\|x^0 - x^*\|_2^2}{\epsilon}\right\}$ *if* n *is sufficiently large.*

Other results in the paper

- Generalization to problems without strong convexity.
- Lower complexity bound for randomized coordinate descent methods.

What's new?

- Gradient sliding algorithms for complex composite optimization.
  - Saving gradient computation significantly without increasing # of iterations.
- An optimal randomized incremental gradient for finite-sum optimization.
  - Saving gradient computation at the expense of more iterations.
- New lower complexity bound and game-theoretic interpretation for first-order methods.