The Proximal-proximal Gradient Algorithm

Ting Kei Pong
PIMS Postdoctoral Fellow
Department of Computer Science
University of British Columbia
Vancouver

WCOM, Autumn
Oct 2013

Outline

- Motivations.
- The proximal-proximal gradient algorithm.
- Convergence and complexity results.
- Other related algorithms.
- Numerical results.

Motivations

- System realization problem: (Liu, Vandenberghe '08)

$$\min_z \quad \frac{1}{2}\|w \circ z - w \circ \bar{z}\|_F^2 + \mu\|\mathcal{H}(z)\|_*,$$

where $z = \begin{pmatrix} z_0 & \cdots & z_{j+k-1} \end{pmatrix} \in \mathbb{R}^{m \times n(j+k)}$, w is zero-one matrix,

$$\mathcal{H}(z) = \begin{pmatrix} z_0 & z_1 & \cdots & z_{k-1} \\ z_1 & \ddots & \ddots & z_k \\ \vdots & \ddots & \ddots & \vdots \\ z_{j-1} & z_j & \cdots & z_{j+k-2} \end{pmatrix} \in \mathbb{R}^{mj \times nk}.$$

- Logistic fused LASSO: (Ma, Zhang '13)

$$\min_{z \in \mathbb{R}^n, t \in \mathbb{R}} \sum_{i=1}^{m} \log(1 + \exp(-b_i(a_i^T z + t))) + \lambda_1\|z\|_1 + \lambda_2 \sum_{i=1}^{n-1} |z_{i+1} - z_i|.$$

General Problem

$$\min_{z} \quad h(z) + P(\mathcal{M}z),$$

where:

- h is smooth, ∇h is Lipschitz continuous with modulus L;
- P is proper closed convex, with “easy” proximal operator;
- $\mathcal{M}$ is nonzero linear map;
- Assume the solution set is nonempty and

$$\mathrm{Range}(\mathcal{M}) \cap \mathrm{ri}(\mathrm{dom}(P)) \neq \emptyset.$$

Proximal Operator

For a proper closed convex function P,

$$\mathrm{prox}_P(y) := \arg\min_z \left\{ P(z) + \frac{1}{2}\|z - y\|^2 \right\}.$$

This is well-defined for all y.

Proximal Operator

For a proper closed convex function P,

$$\mathrm{prox}_P(y) := \arg\min_z \left\{ P(z) + \frac{1}{2}\|z-y\|^2 \right\}.$$

This is well-defined for all y.

Some prox are "easy":

- If $P(z) = \mu\|z\|_1$, then $\mathrm{prox}_P(y) = \mathrm{sign}(y) \circ \max\{|y| - \mu, 0\}$.
- If $P(Z) = \mu\|Z\|_*$, then $\mathrm{prox}_P(Y) = U\mathrm{Diag}(\max\{\sigma(Y) - \mu, 0\})V^T$, where $Y = U\mathrm{Diag}(\sigma(Y))V^T$ is an SVD of Y.

Proximal Gradient Algorithm

$$\min_{z} \quad h(z) + P(\mathcal{M}z),$$

Replace the smooth part with a quadratic approximation:

Proximal Gradient Algorithm

$$\min_{z} \quad h(z) + P(\mathcal{M}z),$$

Replace the smooth part with a quadratic approximation:

For $t = 0, 1, 2, \ldots$, update

$$\begin{aligned} z^{t+1} &= \arg\min_{z} \left\{ \langle \nabla h(z^t), z - z^t \rangle + \frac{L}{2} \|z - z^t\|^2 + P(\mathcal{M}z) \right\} \\ &= \arg\min_{z} \left\{ \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 + P(\mathcal{M}z) \right\}. \end{aligned}$$

Proximal Gradient Algorithm

$$\min_{z} \quad h(z) + P(\mathcal{M}z),$$

Replace the smooth part with a quadratic approximation:

For $t = 0, 1, 2, \ldots$, update

$$\begin{aligned} z^{t+1} &= \arg\min_{z} \left\{ \langle \nabla h(z^t), z - z^t \rangle + \frac{L}{2} \|z - z^t\|^2 + P(\mathcal{M}z) \right\} \\ &= \arg\min_{z} \left\{ \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 + P(\mathcal{M}z) \right\}. \end{aligned}$$

Essentially computing prox for $\frac{1}{L} P \circ \mathcal{M}$: NOT necessarily easy...

Inexact Proximal Gradient

One solution: use iterative method to solve subproblem.

$$\min_z \left\{ \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 + P(\mathcal{M}z) \right\}$$
$$= \max_y \left\{ -\frac{1}{2L} \|\mathcal{M}^* y\|^2 + \langle \mathcal{M}^* y, z^t - \frac{1}{L} \nabla h(z^t) \rangle - P^*(y) \right\}.$$

Moreover, if $\tilde{y}^{t+1}$ solves the maximization problem, then

$$z^{t+1} = z^t - \frac{1}{L}(\nabla h(z^t) + \mathcal{M}^* \tilde{y}^{t+1})$$

solves the minimization problem. This will be the z-update.

Inexact Proximal Gradient

Solve the subproblem also using proximal gradient algorithm:

- Initialize z^0, y^0. Set $\beta = \frac{1}{L}$ and $\tau \geq \beta \|\mathcal{M}\|^2$.
- For $t = 0, 1, 2, \ldots$
  - ⋆ For $s = 0, 1, 2, \ldots$, starting with $u^0 = y^t$, (warm start)

$$u^{s+1} = \operatorname{prox}_{\tau^{-1} P^*}\left(u^s - \frac{1}{\tau}\left(\beta \mathcal{M}\mathcal{M}^* u^s - \mathcal{M}\left(z^t - \beta \nabla h(z^t)\right)\right)\right).$$

  - ⋆ Get approximate solution $y^{t+1} = u^{s+1}$.

  Update $z^{t+1} = z^t - \beta(\nabla h(z^t) + \mathcal{M}^* y^{t+1})$.

Proximal-proximal Gradient Algorithm

- Initialize z^0, y^0. Set $\beta \in (0, \frac{2}{L})$, $\gamma \in (0, 1 + \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\})$ and $\tau \geq \beta\|\mathcal{M}\|^2$.

- For $t = 0, 1, 2, \ldots$

$$\begin{cases} y^{t+1} & = \operatorname{prox}_{\tau^{-1}P^*}\left(y^t - \frac{1}{\tau}\left(\beta\mathcal{M}\mathcal{M}^* y^t - \mathcal{M}\left(z^t - \beta\nabla h(z^t)\right)\right)\right), \\ z^{t+1} & = z^t - \gamma\beta(\nabla h(z^t) + \mathcal{M}^* y^{t+1}). \end{cases}$$

Proximal-proximal Gradient Algorithm

- Initialize z^0, y^0. Set $\beta \in (0, \frac{2}{L})$, $\gamma \in (0, 1 + \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\})$ and $\tau \geq \beta\|\mathcal{M}\|^2$.
- For $t = 0, 1, 2, \ldots$

$$\begin{cases} y^{t+1} & = \operatorname{prox}_{\tau^{-1}P^*}\left(y^t - \frac{1}{\tau}\left(\beta\mathcal{M}\mathcal{M}^*y^t - \mathcal{M}\left(z^t - \beta\nabla h(z^t)\right)\right)\right), \\ z^{t+1} & = z^t - \gamma\beta(\nabla h(z^t) + \mathcal{M}^*y^{t+1}). \end{cases}$$

Remarks:

- This is basically a very inexact proximal gradient algorithm. No need to worry about inner loop accuracy ($s = 0$).
- Computing the prox of P^* is easy, by Moreau's identity.

Convergence

Fenchel dual problem:

$$v_{\text{opt}} := \min_{x,y} \quad h^*(x) + P^*(y) \\ \text{s.t.} \quad x + \mathcal{M}^* y = 0,$$

where $f^*(u) = \sup_z\{\langle u, z\rangle - f(z)\}$.

Fact 1 (P '13): Let $\{(y^t, z^t)\}$ be generated from the PPG algorithm, and set $x^{t+1} = \nabla h(z^t)$. Then $\{z^t\}$ converges to a primal optimal solution, and $\{(x^t, y^t)\}$ converges to a dual optimal solution. Moreover

$$-\frac{C_2}{\sqrt{N}} \le h^*(\bar{x}^N) + P^*(\bar{y}^N) - v_{\text{opt}} \le \frac{C_1}{N}, \quad \|\bar{x}^N + \mathcal{M}^*\bar{y}^N\| \le \frac{C_3}{\sqrt{N}},$$

where $(\bar{x}^N, \bar{y}^N) = \frac{1}{N}\sum_{t=1}^N (x^t, y^t)$.

Intuition Behind Convergence

The proximal gradient algorithm (with $\beta \in (0, \frac{2}{L})$ in place of $\frac{1}{L}$) is the same as the alternating minimization algorithm applied to the Fenchel dual (Tseng '91):

- Initialize z^0, y^0. Set $\beta \in (0, \frac{2}{L})$.
- For $t = 0, 1, 2, \ldots$

$$\begin{cases} x^{t+1} = \arg\min_x \left\{ h^*(x) - \langle z^t, x \rangle \right\}, \\ y^{t+1} \in \operatorname{Arg}\min_y \left\{ P^*(y) - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \| x^{t+1} + \mathcal{M}^* y \|^2 \right\}, \\ z^{t+1} = z^t - \beta(x^{t+1} + \mathcal{M}^* y^{t+1}). \end{cases}$$

Intuition Behind Convergence

Adding "proximal term", we get the PPG algorithm with $\gamma = 1$:

- Initialize z^0, y^0. Set $\beta \in (0, \frac{2}{L})$ and $\tau \geq \beta\|\mathcal{M}\|^2$.
- For $t = 0, 1, 2, \ldots$

$$\begin{cases} x^{t+1} = \arg\min_x \left\{h^*(x) - \langle z^t, x\rangle\right\}, \\ y^{t+1} \in \operatorname{Arg}\min_y \Big\{P^*(y) - \langle z^t, \mathcal{M}^* y\rangle + \frac{\beta}{2}\|x^{t+1} + \mathcal{M}^* y\|^2 \\ \qquad\qquad + \frac{1}{2}\langle y - y^t, [\tau\mathcal{I} - \beta\mathcal{M}\mathcal{M}^*](y - y^t)\rangle\Big\}, \\ z^{t+1} = z^t - \beta(x^{t+1} + \mathcal{M}^* y^{t+1}). \end{cases}$$

Facts about PPG

- The PPG algorithm reduces to the proximal gradient algorithm if $\tau = \beta = \frac{1}{L}$, $\gamma = 1$ and $\mathcal{M} = \mathcal{I}$; or, more generally, $\mathcal{M}\mathcal{M}^* = \mathcal{I}$.

- If $h(z) = \frac{1}{2}\|z - a\|^2$, $\beta = 1 = \frac{1}{L}$ and $\gamma = 1$, then the PPG algorithm reduces to the proximal gradient algorithm applied to the Fenchel dual directly.

- Problems in the form of

$$\min_z \quad h(z) + \sum_{i=1}^{m} P_i(z),$$

can be solved by setting

$$P(z_1, \ldots, z_m) = \sum_{i=1}^{m} P_i(z_i) \text{ and } \mathcal{M}z = (\underbrace{z, \ldots, z}_{m}).$$

Other Approaches for the General Problem

- A recent algorithm by (Condat '13) and (Vu '13) is closely related but yet different from the PPG algorithm.

- Convex-concave minimization maximization:

$$\min_{z} \max_{y} h(z) + \langle \mathcal{M}^* y, z \rangle - P^*(y).$$

A lot of algorithms for solving this type of problem and its variants; see the textbook by (Bauschke, Combettes '08). Most of them reduce to the modified forward-backward splitting method (Tseng '01).

Numerical Simulations

- Compare the PPG algorithm and the MFBS method on system realization problem.

$$\min_z \frac{1}{2}\|w \circ (z - \bar{z})\|_F^2 + \mu\|\mathcal{H}(z)\|_*.$$

  Instances randomly generated. Results averaged over 10 instances.

- Terminate when relative duality gap and relative dual infeasibility are below 10^{-4} and 2×10^{-5}, respectively.

- Two 2.4 GHz quad-core Intel E5620 Xeon 64-bit CPUs, 48 GB RAM, Matlab 7.14 (R2012a).

Numerical Simulations

- Matrix size: $210 \times 10k$.
- Set $\beta = 0.05/L$ for $\mu \geq 0.1$, and $\beta = 1/L$ else. Set $\tau = \beta\|\mathcal{H}\|^2$ and $\gamma = 1 + 0.95 \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\}$.

Table 1: Results for PPG algorithm and MFBS method

		PPG			MFBS		
k	μ	**iter**	**cpu**	**pobj/dobj/dfeas**	**iter**	**cpu**	**pobj/dobj/dfeas**
100	0.05	123	7.4	6.073e+0/6.072e+0/4.3e-6	108	7.3	6.073e+0/6.073e+0/1.5e-5
100	0.10	82	4.7	7.419e+0/7.419e+0/1.7e-5	299	19.4	7.419e+0/7.419e+0/1.9e-6
100	0.50	58	3.8	1.180e+1/1.180e+1/6.3e-6	97	6.9	1.180e+1/1.180e+1/4.8e-6
200	0.05	41	4.8	1.014e+1/1.014e+1/1.1e-5	191	24.3	1.014e+1/1.014e+1/1.9e-5
200	0.10	100	11.7	1.288e+1/1.288e+1/1.7e-5	177	22.8	1.288e+1/1.288e+1/4.9e-6
200	0.50	51	5.9	1.756e+1/1.755e+1/5.5e-6	93	12.1	1.756e+1/1.755e+1/2.7e-6
300	0.05	30	5.0	1.259e+1/1.259e+1/1.4e-5	224	43.2	1.259e+1/1.259e+1/1.9e-5
300	0.10	156	28.7	1.768e+1/1.768e+1/1.8e-5	95	20.1	1.768e+1/1.767e+1/1.2e-5
300	0.50	53	8.8	2.253e+1/2.253e+1/3.9e-6	111	20.9	2.253e+1/2.253e+1/2.0e-6

Conclusion and Future Directions

- The PPG algorithm admits easy subproblems per iteration
- The algorithm is an “inexact” proximal gradient algorithm, and can also be viewed as a proximal alternating minimization algorithm.
- Acceleration of the PPG algorithm?
- Consider other “inexact” algorithms?

Thanks for coming!