Chapter 2
Optimization Algorithms

Convex Optimization for Machine Learning & Computer Vision
WS 2019/20

Tao Wu
Zhenzhang Ye

Computer Vision Group
Department of Informatics
TU Munich

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Gradient-based Methods

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Overview of this section

Unconstrained, differentiable, possibly nonconvex optimization

Problem setting:

$$\text{minimize } J(u) \quad \text{over } u \in \mathbb{E}.$$

Assume:

1. $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable.
2. There exists a global minimizer u^*. (Typically, an optim algorithm seeks for a local minimizer s.t. $\nabla J(u^*) = 0$.)

Overview of this section

Unconstrained, differentiable, possibly nonconvex optimization

Problem setting:

$$\text{minimize } J(u) \quad \text{over } u \in \mathbb{E}.$$

Assume:

1. $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable.
2. There exists a global minimizer u^*. (Typically, an optim algorithm seeks for a local minimizer s.t. $\nabla J(u^*) = 0$.)

Methods under consideration:

1. (Scaled) gradient descent.
2. Line search method.
3. Majorize-minimize method.

Tao Wu
Zhenzhang Ye

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Overview of this section

Unconstrained, differentiable, possibly nonconvex optimization

Problem setting:

$$\text{minimize } J(u) \quad \text{over } u \in \mathbb{E}.$$

Assume:

1. $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable.
2. There exists a global minimizer u^*. (Typically, an optim algorithm seeks for a local minimizer s.t. $\nabla J(u^*) = 0$.)

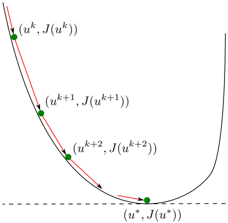
Methods under consideration:

1. (Scaled) gradient descent.
2. Line search method.
3. Majorize-minimize method.

Analytical questions:

1. Convergence (or not); global vs. local convergence.
2. Convergence rate (in special cases).

Descent method



Descent method

Initialize $u^0 \in \mathbb{E}$. Iterate with $k = 0, 1, 2, ...$

1. If the stopping criteria $\|\nabla J(u^k)\| \leq \epsilon$ is *not* satisfied, then continue; otherwise return u^k and stop.
2. Choose a **descent direction** $d^k \in \mathbb{E}$ s.t.
$$\left\langle \nabla J(u^k), d^k \right\rangle < 0.$$
3. Choose an "appropriate" step size $\tau^k > 0$, and update
$$u^{k+1} = u^k + \tau^k d^k.$$

Descent direction

Theorem

If $\langle \nabla J(u^k), d^k \rangle < 0$, then $J(u^k + \tau d^k) < J(u^k)$ for all sufficiently small $\tau > 0$.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Descent direction

Theorem

If $\left\langle \nabla J(u^k), d^k \right\rangle < 0$, then $J(u^k + \tau d^k) < J(u^k)$ for all sufficiently small $\tau > 0$.

Proof: Use the Taylor expansion:

$$
\begin{aligned}
J(u^k + \tau d^k) &= J(u^k) + \tau \left\langle \nabla J(u^k), d^k \right\rangle + o(\tau) \\
&= J(u^k) + \tau \left(\left\langle \nabla J(u^k), d^k \right\rangle + o(1) \right) < J(u^k) \quad \text{as } \tau \to 0^+.
\end{aligned}
$$

Descent direction

Theorem

If $\left\langle \nabla J(u^k), d^k \right\rangle < 0$, then $J(u^k + \tau d^k) < J(u^k)$ for all sufficiently small $\tau > 0$.

<u>Proof</u>: Use the Taylor expansion:

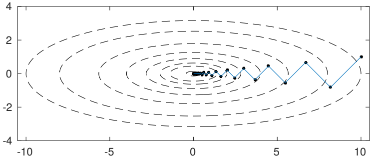
$$J(u^k + \tau d^k) = J(u^k) + \tau \left\langle \nabla J(u^k), d^k \right\rangle + o(\tau)$$
$$= J(u^k) + \tau \left(\left\langle \nabla J(u^k), d^k \right\rangle + o(1) \right) < J(u^k) \quad \text{as } \tau \to 0^+.$$

Choices of descent direction

1. Gradient/Steepest descent:
   $d^k = -\nabla J(u^k) = \arg\min_{d \in \mathbb{E},\, \|d\| \le 1} \left\langle \nabla J(u^k), d \right\rangle$.
2. Scaled gradient: $d^k = -(H^k)^{-1} \nabla J(u^k)$.
3. Newton: $H^k = \nabla^2 J(u^k)$, assuming J is twice continuously differentiable and $\nabla^2 J(u^k)$ is spd.
4. Quasi-Newton: $H^k \approx \nabla^2 J(u^k)$, H^k is spd.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

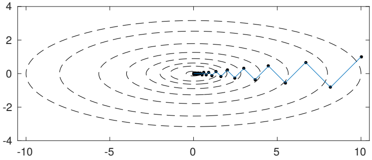
Gradient descent with exact line search



- Gradient descent with *exact* line search:

$$u^{k+1} = u^k - \tau^k \nabla J(u^k),$$
$$\tau^k = \arg\min_{\tau \geq 0} J(u^k - \tau \nabla J(u^k)).$$

Gradient descent with exact line search



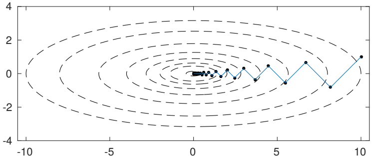
- Gradient descent with *exact* line search:

$$u^{k+1} = u^k - \tau^k \nabla J(u^k),$$
$$\tau^k = \arg\min_{\tau \geq 0} J(u^k - \tau \nabla J(u^k)).$$

- Case study: $J(u) = \frac{1}{2}\langle u, Qu\rangle - \langle b, u\rangle$, matrix Q is spd.
  - $\nabla J(u) = Qu - b, \quad \|\cdot\|_Q^2 \equiv \langle \cdot, Q\cdot\rangle.$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Gradient descent with exact line search



- Gradient descent with *exact* line search:

$$u^{k+1} = u^k - \tau^k \nabla J(u^k),$$
$$\tau^k = \arg\min_{\tau \geq 0} J(u^k - \tau \nabla J(u^k)).$$

- Case study: $J(u) = \frac{1}{2}\langle u, Qu\rangle - \langle b, u\rangle$, matrix Q is spd.
  - $\nabla J(u) = Qu - b, \quad \|\cdot\|_Q^2 \equiv \langle \cdot, Q\, \cdot \rangle.$
  - $\tau^k = \arg\min_{\tau\geq 0} J(u^k - \tau\nabla J(u^k)) = \dfrac{\|\nabla J(u^k)\|^2}{\|\nabla J(u^k)\|_Q^2} \quad \Rightarrow$

$$\|u^{k+1} - u^*\|_Q^2 = \left(1 - \frac{\|\nabla J(u^k)\|^4}{\|\nabla J(u^k)\|_Q^2 \|\nabla J(u^k)\|_{Q^{-1}}^2}\right) \|u^k - u^*\|_Q^2$$
$$\leq \left(\frac{\lambda_{\max}(Q) - \lambda_{\min}(Q)}{\lambda_{\max}(Q) + \lambda_{\min}(Q)}\right)^2 \|u^k - u^*\|_Q^2.$$

Inexact line search

Backtracking line search

- Sufficient decrease condition (let $c_1 \in (0, 1)$):

$$J(u^k + \tau d^k) \leq J(u^k) + c_1 \tau \left\langle \nabla J(u^k), d^k \right\rangle. \qquad \text{(A)}$$

- Curvature condition (let $c_2 \in (c_1, 1)$):

$$\left\langle \nabla J(u^k + \tau d^k), d^k \right\rangle \geq c_2 \left\langle \nabla J(u^k), d^k \right\rangle. \qquad \text{(C)}$$

Inexact line search

Backtracking line search

- Sufficient decrease condition (let $c_1 \in (0,1)$):

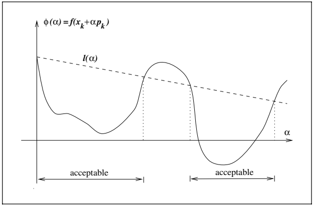
$$J(u^k + \tau d^k) \leq J(u^k) + c_1 \tau \left\langle \nabla J(u^k), d^k \right\rangle . \qquad \text{(A)}$$

- Curvature condition (let $c_2 \in (c_1, 1)$):

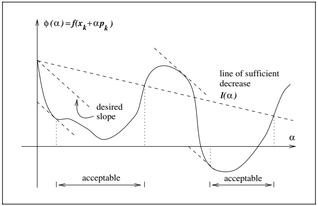
$$\left\langle \nabla J(u^k + \tau d^k), d^k \right\rangle \geq c_2 \left\langle \nabla J(u^k), d^k \right\rangle . \qquad \text{(C)}$$

- (A) $\rightsquigarrow$ **Armijo** line search; (A) & (C) $\rightsquigarrow$ **Wolfe-Powell** l.s.

Armijo l.s.



Wolfe-Powell l.s.



Convergence of backtracking line search

Lemma (feasibility of line search)

Assume that $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable, $\left\langle \nabla J(u^k), d^k \right\rangle < 0 \ \forall k$, and $0 < c_1 < c_2 < 1$. Then there exists an open interval in which the step size τ satisfies (A) and (C).

Proof: on board.

Convergence of backtracking line search

Lemma (feasibility of line search)

Assume that $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable, $\langle \nabla J(u^k), d^k \rangle < 0 \ \forall k$, and $0 < c_1 < c_2 < 1$. Then there exists an open interval in which the step size τ satisfies (A) and (C).

<u>Proof</u>: on board.

Theorem (Zoutendijk)

Assume that $J : \mathbb{E} \to \mathbb{R}$ is cont'ly differentiable, and (A) and (C) are both satisfied with $0 < c_1 < c_2 < 1$ for each k. In addition, J is μ-Lipschitz differentiable on $\{u \in \mathbb{E} : J(u) \leq J(u^0)\}$. Then

$$\sum_{k=0}^{\infty} \frac{\left| \langle \nabla J(u^k), d^k \rangle \right|^2}{\|d^k\|^2} < \infty.$$

<u>Proof</u>: on board.

Remark

If $\dfrac{\left| \langle \nabla J(u^k), d^k \rangle \right|}{\|\nabla J(u^k)\| \|d^k\|} \geq \text{constant} > 0$, then $\lim_{k \to \infty} \|\nabla J(u^k)\| = 0$.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Majorize-minimize method

Majorizing function

A function $\widehat{J}(\cdot\,; u)$ is a **majorant** of J at $u \in \mathbb{E}$ if

$$\begin{cases} \widehat{J}(u; u) = J(u), \\ \widehat{J}(\cdot\,; u) \geq J(\cdot). \end{cases}$$

Majorize-minimize method

Majorizing function

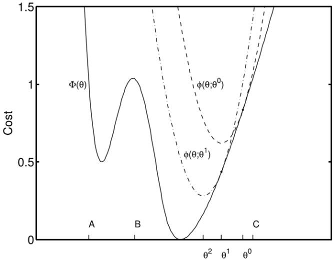
A function $\widehat{J}(\cdot\,; u)$ is a **majorant** of J at $u \in \mathbb{E}$ if

$$\begin{cases} \widehat{J}(u; u) = J(u), \\ \widehat{J}(\cdot\,; u) \geq J(\cdot). \end{cases}$$

Majorize-minimize (MM) algorithm

Let $\widehat{J}(\cdot; u)$ majorize J $\forall u \in \mathbb{E}$. Then the MM iteration reads:

$$u^{k+1} \in \arg\min_u \widehat{J}(u; u^k).$$



Gradient descent as MM

Remark

1. Monotonic decrease of objectives:

$$J(u^{k+1}) \leq \widehat{J}(u^{k+1}; u^k) \leq \widehat{J}(u^k; u^k) = J(u^k).$$

2. Efficiency of MM relies on the choice of the majorant $\widehat{J}(\cdot; u)$, i.e., $\widehat{J}(\cdot; u)$ is easy to minimize.
3. Common choices of $\widehat{J}(\cdot; u)$ are quadratics.

Gradient descent as MM

Remark

1. Monotonic decrease of objectives:

$$J(u^{k+1}) \le \widehat{J}(u^{k+1}; u^k) \le \widehat{J}(u^k; u^k) = J(u^k).$$

2. Efficiency of MM relies on the choice of the majorant $\widehat{J}(\cdot; u)$, i.e., $\widehat{J}(\cdot; u)$ is easy to minimize.
3. Common choices of $\widehat{J}(\cdot; u)$ are quadratics.

Gradient descent as MM

- Observe that $u^{k+1} = u^k - \tau \nabla J(u^k)$ iff

$$u^{k+1} = \arg\min_u J(u^k) + \left\langle \nabla J(u^k), u - u^k \right\rangle + \frac{1}{2\tau}\|u - u^k\|^2.$$

- When $J(u^k) + \langle \nabla J(u^k), \cdot - u^k \rangle + \frac{1}{2\tau}\| \cdot - u^k\|^2 \ge J(\cdot)$ holds?

Gradient descent as MM

Lemma

Assume that $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz differentiable. Then $\forall u, v \in \mathbb{E}$:

$$|J(v) - J(u) - \langle \nabla J(u), v - u \rangle| \leq \frac{\mu}{2} \|v - u\|^2.$$

Proof: on board.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Gradient descent as MM

Lemma

Assume that $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz differentiable. Then $\forall u, v \in \mathbb{E}$:

$$|J(v) - J(u) - \langle \nabla J(u), v - u \rangle| \leq \frac{\mu}{2}\|v - u\|^2.$$

Proof: on board.

Theorem (convergence of gradient descent)

Assume that $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz differentiable. Then the gradient descent iteration

$$u^{k+1} = u^k - \tau \nabla J(u^k)$$

with $\tau \in (0, 1/\mu]$ yields $\lim_{k\to\infty} \nabla J(u^k) = 0$.

Proof: on board.

Tao Wu
Zhenzhang Ye

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Gradient descent as MM

Lemma

Assume that $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz differentiable. Then $\forall u, v \in \mathbb{E}$:

$$|J(v) - J(u) - \langle \nabla J(u), v - u \rangle| \leq \frac{\mu}{2} \|v - u\|^2.$$

<u>Proof</u>: on board.

Theorem (convergence of gradient descent)

Assume that $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz differentiable. Then the gradient descent iteration

$$u^{k+1} = u^k - \tau \nabla J(u^k)$$

with $\tau \in (0, 1/\mu]$ yields $\lim_{k \to \infty} \nabla J(u^k) = 0$.

<u>Proof</u>: on board.

Recipe of convergence

By solving the surrogate problem in MM, we achieve: (1) sufficient decrease in the objective; (2) inexact optimality condition which matches the exact OC in the limit.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Proximal Algorithms

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Agenda for the rest of the chapter

- Four proximal algorithms in convex optimization:
  - Forward-backward splitting (FBS) ($=$ proximal gradient method).
  - Alternating direction method of multipliers (ADMM).
  - Primal-dual hybrid gradient (PDHG).
  - Douglas-Rachford splitting (DRS), Peaceman-Rachford splitting (PRS).
- A common theme: “operator splitting” & extensively use prox operations.

Agenda for the rest of the chapter

- Four proximal algorithms in convex optimization:
  - Forward-backward splitting (FBS) ($=$ proximal gradient method).
  - Alternating direction method of multipliers (ADMM).
  - Primal-dual hybrid gradient (PDHG).
  - Douglas-Rachford splitting (DRS), Peaceman-Rachford splitting (PRS).
- A common theme: "operator splitting" & extensively use prox operations.
- We'll cover: (1) Derivation of algorithms; (2) Connections between algorithms; (3) Demo in Python; and ...
- Unified convergence analysis.
- Acceleration techniques.

Forward-backward splitting

- Consider the convex optimization problem:

$$\min_u F(u) + G(u),$$

with G differentiable and F possibly non-differentiable.

- Its minimizer is characterized by

$$0 \in \partial F(u) + \nabla G(u).$$

Forward-backward splitting

- Consider the convex optimization problem:

$$\min_u F(u) + G(u),$$

with G differentiable and F possibly non-differentiable.

- Its minimizer is characterized by

$$0 \in \partial F(u) + \nabla G(u).$$

- **Forward-backward splitting** (FBS):

$$\begin{aligned} u^{k+1} &= \mathrm{prox}_{\tau F}(u^k - \tau \nabla G(u^k)) \\ &= (I + \tau \partial F)^{-1} \circ (I - \tau \nabla G)(u^k). \end{aligned}$$

- FBS as *semi-implicit Euler scheme*:

$$\frac{u^{k+1} - u^k}{\tau} \in -\partial F(u^{k+1}) - \nabla G(u^k).$$

Example: Split feasibility problem

Split feasibility problem

Given nonempty, closed, convex sets $C_1 \subset \mathbb{E}_1$, $C_2 \subset \mathbb{E}_2$, and linear operator $K : \mathbb{E}_1 \to \mathbb{E}_2$, find $u \in \mathbb{E}_1$ s.t. $u \in C_1,\ Ku \in C_2$.

- Variational model:

$$\min_{u \in \mathbb{E}_1} \delta_{C_1}(u) + \frac{1}{2}\|Ku - \mathrm{proj}_{C_2}(Ku)\|^2.$$

Note that $\frac{1}{2}\|v - \mathrm{proj}_{C_2}(v)\|^2 = \mathrm{env}_1\, \delta_{C_2}(v)$.

Example: Split feasibility problem

Split feasibility problem

Given nonempty, closed, convex sets $C_1 \subset \mathbb{E}_1$, $C_2 \subset \mathbb{E}_2$, and linear operator $K : \mathbb{E}_1 \to \mathbb{E}_2$, find $u \in \mathbb{E}_1$ s.t. $u \in C_1,\ Ku \in C_2$.

- Variational model:

$$\min_{u \in \mathbb{E}_1} \delta_{C_1}(u) + \frac{1}{2}\|Ku - \mathrm{proj}_{C_2}(Ku)\|^2.$$

  Note that $\frac{1}{2}\|v - \mathrm{proj}_{C_2}(v)\|^2 = \mathrm{env}_1\, \delta_{C_2}(v)$.

- Optimality condition:

$$0 \in \partial \delta_{C_1}(u) + K^\top (I - \mathrm{proj}_{C_2})(Ku).$$

  Recall that $\nabla \mathrm{env}_1\, \delta_{C_2}(v) = (I - \mathrm{prox}_{\delta_{C_2}})(v)$.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Example: Split feasibility problem

Split feasibility problem

Given nonempty, closed, convex sets $C_1 \subset \mathbb{E}_1$, $C_2 \subset \mathbb{E}_2$, and linear operator $K : \mathbb{E}_1 \to \mathbb{E}_2$, find $u \in \mathbb{E}_1$ s.t. $u \in C_1,\ Ku \in C_2$.

- Variational model:

$$\min_{u \in \mathbb{E}_1} \delta_{C_1}(u) + \frac{1}{2}\|Ku - \mathrm{proj}_{C_2}(Ku)\|^2.$$

  Note that $\frac{1}{2}\|v - \mathrm{proj}_{C_2}(v)\|^2 = \mathrm{env}_1\, \delta_{C_2}(v)$.

- Optimality condition:

$$0 \in \partial \delta_{C_1}(u) + K^\top (I - \mathrm{proj}_{C_2})(Ku).$$

  Recall that $\nabla\, \mathrm{env}_1\, \delta_{C_2}(v) = (I - \mathrm{prox}_{\delta_{C_2}})(v)$.

- Apply FBS $\Rightarrow$

$$\begin{aligned} u^{k+1} &= (I + \tau \partial \delta_{C_1})^{-1}(u^k - \tau K^\top (I - \mathrm{proj}_{C_2})(Ku^k)) \\ &= \mathrm{proj}_{C_1}(u^k - \tau K^\top (I - \mathrm{proj}_{C_2})(Ku^k)). \end{aligned}$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Alternating direction method of multipliers

- Consider

$$\min_{u,v} \; J(u,v) = F(v) + G(u) + \delta\{Ku - v = 0\},$$

given proper, convex, lsc functions F, G and matrix K.

Alternating direction method of multipliers

- Consider

$$\min_{u,v} \; J(u,v) = F(v) + G(u) + \delta\{Ku - v = 0\},$$

given proper, convex, lsc functions F, G and matrix K.

- *Augmented Lagrangian* ($\tau > 0$):

$$\mathcal{L}_\tau(u,v;p) = F(v) + G(u) + \langle p, Ku - v\rangle + \frac{\tau}{2}\|Ku - v\|^2,$$

such that

$$\min_{u,v} J(u,v) = \inf_{u,v}\sup_{p} \mathcal{L}_\tau(u,v;p).$$

Alternating direction method of multipliers

- Consider

$$\min_{u,v} \; J(u,v) = F(v) + G(u) + \delta\{Ku - v = 0\},$$

given proper, convex, lsc functions F, G and matrix K.

- *Augmented Lagrangian* ($\tau > 0$):

$$\mathcal{L}_\tau(u,v;p) = F(v) + G(u) + \langle p, Ku - v\rangle + \frac{\tau}{2}\|Ku - v\|^2,$$

such that

$$\min_{u,v} J(u,v) = \inf_{u,v}\sup_{p} \mathcal{L}_\tau(u,v;p).$$

- **Alternating direction method of multipliers** (ADMM):

$$\begin{cases} u^{k+1} \in \arg\min_u G(u) + \left\langle p^k, Ku \right\rangle + \frac{\tau}{2}\|Ku - v^k\|^2, \\ v^{k+1} \in \arg\min_v F(v) - \left\langle p^k, v \right\rangle + \frac{\tau}{2}\|Ku^{k+1} - v\|^2, \\ p^{k+1} = p^k + \tau(Ku^{k+1} - v^{k+1}). \end{cases}$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Example: Consensus ADMM

- **Empirical risk minimization** (ERM):

$$\min_u \; F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(u),$$

where G_i represents the training error on sample (x_i, y_i):

$$G_i(u) = \text{loss}(h(x_i; u), y_i),$$

and F represents the model prior.

Example: Consensus ADMM

- **Empirical risk minimization** (ERM):

$$\min_{u} \; F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(u),$$

where G_i represents the training error on sample (x_i, y_i):

$$G_i(u) = \mathrm{loss}(h(x_i; u), y_i),$$

and F represents the model prior.

- **Consensus optimization**:

$$\min_{u,\{v_i\}} \; F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(v_i)$$
$$\text{s.t.} \;\; v_i = u \quad \forall i \in \{1, ..., n\}.$$

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Example: Consensus ADMM

- Consensus optimization:

$$
\begin{aligned}
\min_{u,\{v_i\}} \quad & F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(v_i) \\
\text{s.t.} \quad & v_i = u \quad \forall i \in \{1, ..., n\}.
\end{aligned}
$$

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Example: Consensus ADMM

- Consensus optimization:

$$\min_{u,\{v_i\}} \; F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(v_i)$$

$$\text{s.t.} \;\; v_i = u \quad \forall i \in \{1, ..., n\}.$$

- Augmented Lagrangian:

$$\mathcal{L}_\tau(u, \{v_i\}, \{p_i\}) = F(u) + \frac{1}{n}\sum_{i=1}^{n}\left(G_i(v_i) + \langle p_i, v_i - u\rangle + \frac{\tau}{2}\|v_i - u\|^2 \right).$$

Example: Consensus ADMM

- Consensus optimization:

$$\min_{u,\{v_i\}} \; F(u) + \frac{1}{n}\sum_{i=1}^{n} G_i(v_i)$$
$$\text{s.t. } \; v_i = u \quad \forall i \in \{1, ..., n\}.$$

- Augmented Lagrangian:

$$\mathcal{L}_\tau(u, \{v_i\}, \{p_i\}) = F(u) + \frac{1}{n}\sum_{i=1}^{n}\left(G_i(v_i) + \langle p_i, v_i - u\rangle + \frac{\tau}{2}\|v_i - u\|^2\right).$$

- **Consensus ADMM**:

$$u^{k+1} = \mathrm{prox}_{F/\tau}\left(\frac{1}{n}\sum_{i=1}^{n}\left(v_i^k + \frac{1}{\tau}p_i^k\right)\right),$$
$$\forall i: \; v_i^{k+1} = \mathrm{prox}_{G_i/\tau}\left(u^{k+1} - \frac{1}{\tau}p_i^k\right),$$
$$\forall i: \; p_i^{k+1} = p_i^k + \tau(v_i^{k+1} - u^{k+1}).$$

Primal-dual hybrid gradient

- By Fenchel-Rockafellar duality theorem, we reformulate

$$\min_u \; F(Ku) + G(u)$$

as the saddle-point problem:

$$\sup_p \inf_u \langle p, Ku \rangle + G(u) - F^*(p).$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Primal-dual hybrid gradient

- By Fenchel-Rockafellar duality theorem, we reformulate

$$\min_{u} \; F(Ku) + G(u)$$

as the saddle-point problem:

$$\sup_{p} \inf_{u} \langle p, Ku \rangle + G(u) - F^*(p).$$

- **Primal-dual hybrid gradient** (PDHG) ($st > \|K\|^2$):

$$u^{k+1} = \arg\min_{u} \left\langle u, K^\top p^k \right\rangle + G(u) + \frac{s}{2}\|u - u^k\|^2,$$
$$p^{k+1} = \arg\min_{p} -\left\langle K(2u^{k+1} - u^k), p \right\rangle + F^*(p) + \frac{t}{2}\|p - p^k\|^2.$$

- Optimality conditions for the updates:

$$0 \in \partial G(u^{k+1}) + K^\top p^k + s(u^{k+1} - u^k),$$
$$0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + t(p^{k+1} - p^k).$$

Scaled primal-dual hybrid gradient

- Recall PDGH:

$$
\begin{aligned}
0 &\in \partial G(u^{k+1}) + K^\top p^k + s(u^{k+1} - u^k), \\
0 &\in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + t(p^{k+1} - p^k).
\end{aligned}
$$

- Replace s, t by spd matrices $S, T \rightsquigarrow$ Scaled PDHG:

$$
\begin{aligned}
0 &\in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k), \\
0 &\in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k).
\end{aligned}
$$

- Scaled PDHG in compact form:

$$
0 \in \begin{bmatrix} S & -K^\top \\ -K & T \end{bmatrix} \left(\begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} - \begin{bmatrix} u^k \\ p^k \end{bmatrix} \right) + \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.
$$

Scaled primal-dual hybrid gradient

- Recall PDGH:

$$0 \in \partial G(u^{k+1}) + K^\top p^k + s(u^{k+1} - u^k),$$
$$0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + t(p^{k+1} - p^k).$$

- Replace s, t by spd matrices S, $T \rightsquigarrow$ Scaled PDHG:

$$0 \in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k),$$
$$0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k).$$

- Scaled PDHG in compact form:

$$0 \in \begin{bmatrix} S & -K^\top \\ -K & T \end{bmatrix} \left(\begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} - \begin{bmatrix} u^k \\ p^k \end{bmatrix} \right) + \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.$$

- Scaled PDHG is a **customized proximal iteration**:

$$\boxed{0 \in M(\xi^{k+1} - \xi^k) + R(\xi^{k+1})} \;\Leftrightarrow\; \boxed{\xi^{k+1} = (M + R)^{-1} M \xi^k}$$

- Sufficient conditions to ensure convergence:
(1) M is spd matrix; (2) R is maximal monotone operator.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Interpret ADMM as customized proximal iteration

- Recall ADMM (with reordered updates):

$$v^{k+1} \in \arg\min_{v} F(v) - \left\langle p^k, v \right\rangle + \frac{\tau}{2}\|Ku^k - v\|^2, \tag{1}$$

$$p^{k+1} = p^k + \tau(Ku^k - v^{k+1}), \tag{2}$$

$$u^{k+1} \in \arg\min_{u} G(u) + \left\langle p^{k+1}, Ku \right\rangle + \frac{\tau}{2}\|Ku - v^{k+1}\|^2. \tag{3}$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Interpret ADMM as customized proximal iteration

- Recall ADMM (with reordered updates):

$$v^{k+1} \in \arg\min_v F(v) - \left\langle p^k, v \right\rangle + \frac{\tau}{2}\|Ku^k - v\|^2, \tag{1}$$

$$p^{k+1} = p^k + \tau(Ku^k - v^{k+1}), \tag{2}$$

$$u^{k+1} \in \arg\min_u G(u) + \left\langle p^{k+1}, Ku \right\rangle + \frac{\tau}{2}\|Ku - v^{k+1}\|^2. \tag{3}$$

- ADMM as customized proximal iteration:

$$(1) \;\Rightarrow\; 0 \in \partial F(v^{k+1}) - p^k + \tau(v^{k+1} - Ku^k), \tag{4}$$

$$(3) \;\Rightarrow\; 0 \in \partial G(u^{k+1}) + K^\top p^{k+1} + \tau K^\top (Ku^{k+1} - v^{k+1}), \tag{5}$$

$$(2),(4) \;\Rightarrow\; p^{k+1} \in \partial F(v^{k+1}) \;\Leftrightarrow\; v^{k+1} \in \partial F^*(p^{k+1}), \tag{6}$$

$$(2),(5) \;\Rightarrow\; 0 \in \partial G(u^{k+1}) + K^\top(2p^{k+1} - p^k) + \tau K^\top K(u^{k+1} - u^k), \tag{7}$$

$$(2),(6) \;\Rightarrow\; 0 \in -Ku^k + \frac{1}{\tau}(p^{k+1} - p^k) + \partial F^*(p^{k+1}), \tag{8}$$

$$(7),(8) \;\Rightarrow\; 0 \in \begin{bmatrix} \tau K^\top K & K^\top \\ K & \frac{1}{\tau} I \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix} + \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.$$

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Reflection operator

- Given a proper, convex, lsc function $J : \mathbb{E} \to \overline{\mathbb{R}}$ and $\tau > 0$, we call

$$\mathrm{refl}_{\tau J} = 2\,\mathrm{prox}_{\tau J} - I = 2(I + \tau\partial J)^{-1} - I$$

the **reflection operator** on ∂J.

- In a more general definition for "refl", ∂J is replaced by a *maximal monotone operator*.
  - We don't formally introduce maximal monotone operator.
  - Fact: For any proper, convex, lsc function J, ∂J is indeed a maximal monotone operator.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Reflection operator

- Given a proper, convex, lsc function $J : \mathbb{E} \to \overline{\mathbb{R}}$ and $\tau > 0$, we call

$$\mathrm{refl}_{\tau J} = 2\,\mathrm{prox}_{\tau J} - I = 2(I + \tau\partial J)^{-1} - I$$

the **reflection operator** on ∂J.

- In a more general definition for "refl", ∂J is replaced by a *maximal monotone operator*.
  - We don't formally introduce maximal monotone operator.
  - Fact: For any proper, convex, lsc function J, ∂J is indeed a maximal monotone operator.

- Fixed points of $\mathrm{refl}_{\tau J}$:

$$\begin{aligned} & u = \mathrm{refl}_{\tau J}(u) \\ \Leftrightarrow\ & u = 2\,\mathrm{prox}_{\tau J}(u) - u \\ \Leftrightarrow\ & u = \mathrm{prox}_{\tau J}(u) \\ \Leftrightarrow\ & 0 \in \partial J(u). \end{aligned}$$

Douglas-Rachford- & Peaceman-Rachford splitting

- Consider the *monotone inclusion* problem:

$$0 \in \partial F(u) + \partial G(u).$$

Douglas-Rachford- & Peaceman-Rachford splitting

- Consider the *monotone inclusion* problem:

$$0 \in \partial F(u) + \partial G(u).$$

- **Douglas-Rachford splitting** (DRS):

$$\begin{cases} u^{k+1} = \operatorname{prox}_{\tau G}(v^k), \\ v^{k+1} = v^k - u^{k+1} + \operatorname{prox}_{\tau F}(2u^{k+1} - v^k). \end{cases} \tag{DRS}$$

- **Peaceman-Rachford splitting** (PRS):

$$\begin{cases} u^{k+1} = \operatorname{prox}_{\tau G}(v^k), \\ v^{k+1} = v^k - 2u^{k+1} + 2\operatorname{prox}_{\tau F}(2u^{k+1} - v^k). \end{cases} \tag{PRS}$$

- DRS & PRS in compact forms:

$$v^{k+1} = \left(\frac{1}{2}I + \frac{1}{2}\operatorname{refl}_{\tau F} \circ \operatorname{refl}_{\tau G}\right)(v^k), \tag{DRS'}$$

$$v^{k+1} = (\operatorname{refl}_{\tau F} \circ \operatorname{refl}_{\tau G})(v^k). \tag{PRS'}$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Douglas-Rachford- & Peaceman-Rachford splitting

Fixed points of DRS & PRS:

$$
\begin{aligned}
& v = \mathrm{refl}_{\tau F}(\mathrm{refl}_{\tau G}(v)) = 2\,\mathrm{prox}_{\tau F}(\mathrm{refl}_{\tau G}(v)) - \mathrm{refl}_{\tau G}(v) \\
\Leftrightarrow\ & \mathrm{prox}_{\tau F}(\mathrm{refl}_{\tau G}(v)) = \mathrm{prox}_{\tau G}(v) \\
\Leftrightarrow\ & \mathrm{refl}_{\tau G}(v) \in (I + \tau\partial F)(\mathrm{prox}_{\tau G}(v)) \\
\Leftrightarrow\ & 2\,\mathrm{prox}_{\tau G}(v) - v \in \mathrm{prox}_{\tau G}(v) + \tau\partial F(\mathrm{prox}_{\tau G}(v)) \\
\Leftrightarrow\ & \mathrm{prox}_{\tau G}(v) - v \in \tau\partial F(\mathrm{prox}_{\tau G}(v)) \\
\Leftrightarrow\ & u = \mathrm{prox}_{\tau G}(v), \quad u - v \in \tau\partial F(u) \\
\Leftrightarrow\ & v \in u + \tau\partial G(u), \quad u - v \in \tau\partial F(u) \\
\Leftrightarrow\ & 0 \in \partial F(u) + \partial G(u).
\end{aligned}
$$

Interpret DRS as customized proximal iteration

- Apply DRS to: $\min_u F(u) + G(u)$. $\Rightarrow$

$$u^{k+1} = \mathrm{prox}_{\tau G}(v^k), \tag{1}$$
$$v^{k+1} = v^k - u^{k+1} + \mathrm{prox}_{\tau F}(2u^{k+1} - v^k). \tag{2}$$

Interpret DRS as customized proximal iteration

- Apply DRS to: $\min_u F(u) + G(u)$. $\Rightarrow$

$$u^{k+1} = \mathrm{prox}_{\tau G}(v^k), \tag{1}$$
$$v^{k+1} = v^k - u^{k+1} + \mathrm{prox}_{\tau F}(2u^{k+1} - v^k). \tag{2}$$

- DRS as customized proximal iteration ($p^k := (u^k - v^k)/\tau$):

$$\begin{aligned}
(1) \;&\Leftrightarrow\; u^{k+1} = \mathrm{prox}_{\tau G}(u^k - \tau p^k) \;\Leftrightarrow\; u^k - \tau p^k \in (I + \tau\partial G)u^{k+1} \\
&\Leftrightarrow\; 0 \in (u^{k+1} - u^k)/\tau + p^k + \partial G(u^{k+1}), \qquad (3) \\
(2) \;&\Leftrightarrow\; 2u^{k+1} - u^k + \tau p^k = \tau p^{k+1} + \mathrm{prox}_{\tau F}(2u^{k+1} - u^k + \tau p^k) \\
&\Rightarrow\; \tau p^{k+1} = (I - \mathrm{prox}_{\tau F})(2u^{k+1} - u^k + \tau p^k) \\
&\Leftrightarrow\; p^{k+1} = \mathrm{prox}_{\frac{1}{\tau}F^*}((2u^{k+1} - u^k)/\tau + p^k) \quad \text{by Moreau's identity} \\
&\Leftrightarrow\; (2u^{k+1} - u^k)/\tau + p^k \in \left(I + \frac{1}{\tau}\partial F^*\right)(p^{k+1}) \\
&\Leftrightarrow\; 0 \in \tau(p^{k+1} - p^k) + \partial F^*(p^{k+1}) - (2u^{k+1} - u^k), \qquad (4)
\end{aligned}$$

$$(3), (4) \;\Rightarrow\; 0 \in \begin{bmatrix} \frac{1}{\tau}I & -I \\ -I & \tau I \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix} + \begin{bmatrix} \partial G & I \\ -I & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.$$

Demo: Image segmentation

- Variational model:

$$\min_{u:\Omega\to\Delta^L} \sum_{j\in\Omega} \Big(\delta\{u_j \in \Delta^L\} + \langle u_j, f_j\rangle\Big) + \alpha \sum_{l=1}^{L} \|\nabla u^l\|_1,$$

where Δ^L is the probability simplex in $\mathbb{R}^L$.

- Segmentation results:

image

segmentation ($L = 4$)

- Demo code in PyTorch (possibly accelerated by GPU) is provided on the course webpage.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Convergence Theory

Tao Wu
Zhenzhang Ye

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Fixed-point iteration

Fixed-point iteration

Proximal algorithm as *fixed-point iteration*:

$$u^{k+1} = \Phi(u^k).$$

Its convergence depends on the property of Φ.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Fixed-point iteration

Fixed-point iteration

Proximal algorithm as *fixed-point iteration*:

$$u^{k+1} = \Phi(u^k).$$

Its convergence depends on the property of Φ.

Definition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$ (equipped with $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$), and $\Phi : C \to \mathbb{E}$. Then Φ is:

1. μ-Lipschitz with modulus $\mu \geq 0$ if

$$\forall u, v \in C : \|\Phi(u) - \Phi(v)\| \leq \mu \|u - v\|.$$

2. **contractive** if Φ is μ-Lipschitz with modulus $\mu \in [0, 1)$.
3. **nonexpansive** if Φ is 1-Lipschitz.

Tao Wu
Zhenzhang Ye

Fixed-point iteration

Definition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$ (equipped with $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$), and $\Phi : C \to \mathbb{E}$. Then Φ is:

1. μ-Lipschitz with modulus $\mu \geq 0$ if

$$\forall u, v \in C : \|\Phi(u) - \Phi(v)\| \leq \mu \|u - v\|.$$

2. **contractive** if Φ is μ-Lipschitz with modulus $\mu \in [0, 1)$.
3. **nonexpansive** if Φ is 1-Lipschitz.

Remark

1. If Φ is contractive (mod. $\mu \in [0, 1)$), then by **Banach fixed point theorem** the iteration $u^{k+1} = \Phi(u^k)$ converges to the unique fixed point u^* linearly: $\|u^k - u^*\| \leq \mu^k \|u^0 - u^*\|$.

Fixed-point iteration

Definition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$ (equipped with $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$), and $\Phi : C \to \mathbb{E}$. Then Φ is:

1. μ-Lipschitz with modulus $\mu \geq 0$ if
$$\forall u, v \in C : \|\Phi(u) - \Phi(v)\| \leq \mu \|u - v\|.$$
2. **contractive** if Φ is μ-Lipschitz with modulus $\mu \in [0, 1)$.
3. **nonexpansive** if Φ is 1-Lipschitz.

Remark

1. If Φ is contractive (mod. $\mu \in [0, 1)$), then by **Banach fixed point theorem** the iteration $u^{k+1} = \Phi(u^k)$ converges to the unique fixed point u^* linearly: $\|u^k - u^*\| \leq \mu^k \|u^0 - u^*\|$.
2. Unfortunately, Banach fixed point theorem does not apply here. Most proximal algorithms consist of nonexpansive operators Φ (including proj, prox, and refl), which are not contractive but "averaged' operators".

Averaged operator

Definition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$. Then a map $\Phi : C \to \mathbb{E}$ is α-**averaged**, with $\alpha \in (0, 1)$, if there exists a nonexpansive operator $\Psi : C \to \mathbb{E}$ such that

$$\Phi = (1 - \alpha)I + \alpha\Psi.$$

In particular, "$\frac{1}{2}$-averaged" is also called **firmly nonexpansive**.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Averaged operator

Definition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$. Then a map $\Phi : C \to \mathbb{E}$ is α-**averaged**, with $\alpha \in (0,1)$, if there exists a nonexpansive operator $\Psi : C \to \mathbb{E}$ such that

$$\Phi = (1-\alpha)I + \alpha\Psi.$$

In particular, "$\frac{1}{2}$-averaged" is also called **firmly nonexpansive**.

Proposition

Let C be a nonempty, closed, convex subset of $\mathbb{E}$, $\Phi : C \to \mathbb{E}$, and $\alpha \in (0,1)$. Then the following statements are equivalent:

1. Φ is α-averaged.
2. $(1-\frac{1}{\alpha})I + \frac{1}{\alpha}\Phi$ is nonexpansive.
3. $\forall u, v \in C : \|\Phi(u) - \Phi(v)\|^2 \leq \|u-v\|^2 - \frac{1-\alpha}{\alpha}\|(I-\Phi)(u) - (I-\Phi)(v)\|^2$.
4. $\forall u, v \in C : \|\Phi(u) - \Phi(v)\|^2 + (1-2\alpha)\|u-v\|^2 \leq$
   $2(1-\alpha)\langle u-v, \Phi(u)-\Phi(v)\rangle$.

Proof: on board.

Averaged operator in proximal algorithms

- Recall the customized proximal iteration:

$$u^{k+1} = \Phi^{(\text{cpi})}(u^k), \quad \Phi^{(\text{cpi})} = (M + R)^{-1} M,$$

  for given spd matrix M and monotone operator R.
  - One can verify that $\Phi^{(\text{cpi})}$ is firmly nonexpansive under the scaled norm $\|\cdot\|_M = \sqrt{\langle \cdot, M \cdot \rangle}$.

Averaged operator in proximal algorithms

- Recall the customized proximal iteration:

$$u^{k+1} = \Phi^{(\text{cpi})}(u^k), \ \ \Phi^{(\text{cpi})} = (M + R)^{-1} M,$$

for given spd matrix M and monotone operator R.

  - One can verify that $\Phi^{(\text{cpi})}$ is firmly nonexpansive under the scaled norm $\| \cdot \|_M = \sqrt{\langle \cdot, M \cdot \rangle}$.

- Recall Douglas-Rachford splitting (in compact form):

$$v^{k+1} = \Phi^{(\text{drs})}(v^k), \ \ \Phi^{(\text{drs})} = \frac{1}{2} I + \frac{1}{2} \operatorname{refl}_{\tau F} \circ \operatorname{refl}_{\tau G},$$

for some proper, convex, lsc functions $F, G : \mathbb{E} \to \overline{\mathbb{R}}$.

  - Since $\operatorname{refl}_{\tau F} = 2 \operatorname{prox}_{\tau F} - I$ is nonexpansive and so is $\operatorname{refl}_{\tau G}$, $\Phi^{(\text{drs})}$ is firmly nonexpansive.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Averaged operator in proximal algorithms

- Recall the customized proximal iteration:

$$u^{k+1} = \Phi^{(\mathrm{cpi})}(u^k), \quad \Phi^{(\mathrm{cpi})} = (M+R)^{-1}M,$$

  for given spd matrix M and monotone operator R.
  - One can verify that $\Phi^{(\mathrm{cpi})}$ is firmly nonexpansive under the scaled norm $\|\cdot\|_M = \sqrt{\langle \cdot, M\cdot \rangle}$.

- Recall Douglas-Rachford splitting (in compact form):

$$v^{k+1} = \Phi^{(\mathrm{drs})}(v^k), \quad \Phi^{(\mathrm{drs})} = \frac{1}{2}I + \frac{1}{2}\,\mathrm{refl}_{\tau F} \circ \mathrm{refl}_{\tau G},$$

  for some proper, convex, lsc functions $F, G : \mathbb{E} \to \overline{\mathbb{R}}$.
  - Since $\mathrm{refl}_{\tau F} = 2\,\mathrm{prox}_{\tau F} - I$ is nonexpansive and so is $\mathrm{refl}_{\tau G}$, $\Phi^{(\mathrm{drs})}$ is firmly nonexpansive.

- Recall forward-backward splitting:

$$u^{k+1} = \Phi^{(\mathrm{fbs})}(u^k), \quad \Phi^{(\mathrm{fbs})} = \mathrm{prox}_{\tau F} \circ (I - \tau \nabla G),$$

  where G is μ-Lipschitz differentiable and $\tau \in (0, 2/\mu)$.
  - As a consequence of the Baillon-Haddad Theorem (next slide), $I - \tau\nabla G$ is an averaged operator. Hence, $\Phi^{(\mathrm{fbs})}$ is a composition of two averaged operators (again averaged).

Averaged operator in gradient descent

Theorem (Baillon-Haddad)

Let $J : \mathbb{E} \to \mathbb{R}$ be a convex, continuously differentiable function. Then ∇J is a nonexpansive operator iff ∇J is firmly nonexpansive.

Proof: on board.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Averaged operator in gradient descent

Theorem (Baillon-Haddad)

Let $J : \mathbb{E} \to \mathbb{R}$ be a convex, continuously differentiable function. Then ∇J is a nonexpansive operator iff ∇J is firmly nonexpansive.

Proof: on board.

Corollary

Assume $G : \mathbb{E} \to \mathbb{R}$ is convex and μ-Lipschitz differentiable, and $\tau = 2\alpha/\mu$ with $\alpha \in (0, 1)$. Then $I - \tau\nabla G$ is α-averaged.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Averaged operator in gradient descent

Theorem (Baillon-Haddad)

Let $J : \mathbb{E} \to \mathbb{R}$ be a convex, continuously differentiable function. Then ∇J is a nonexpansive operator iff ∇J is firmly nonexpansive.

Proof: on board.

Corollary

Assume $G : \mathbb{E} \to \mathbb{R}$ is convex and μ-Lipschitz differentiable, and $\tau = 2\alpha/\mu$ with $\alpha \in (0, 1)$. Then $I - \tau\nabla G$ is α-averaged.

Proof: Since $\frac{1}{\mu}\nabla G$ is nonexpansive, by the Baillon-Haddad theorem, $\frac{1}{\mu}\nabla G$ is firmly nonexpansive, i.e., $\exists\Psi : \mathbb{E} \to \mathbb{E}$ nonexpansive s.t. $\frac{1}{\mu}\nabla G = \frac{1}{2}I + \frac{1}{2}\Psi$. Hence,

$$I - \tau\nabla G = (1 - \frac{\tau\mu}{2})I - \frac{\tau\mu}{2}\Psi = (1 - \alpha)I + \alpha(-\Psi),$$

i.e. $I - \tau\nabla G$ is α-averaged.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Composition of averaged operators

In forward-backward splitting,

$$\Phi^{(\text{fbs})} = \text{prox}_{\tau F} \circ \left(I - \frac{2\alpha}{\mu} \nabla G \right)$$

appears as the composition of a $\frac{1}{2}$-averaged operator $\text{prox}_{\tau F}$ and an α-averaged operator $I - \frac{2\alpha}{\mu} \nabla G$ with $\alpha \in (0, 1)$.

Composition of averaged operators

In forward-backward splitting,

$$\Phi^{(\mathrm{fbs})} = \mathrm{prox}_{\tau F} \circ \left(I - \frac{2\alpha}{\mu} \nabla G \right)$$

appears as the composition of a $\frac{1}{2}$-averaged operator $\mathrm{prox}_{\tau F}$ and an α-averaged operator $I - \frac{2\alpha}{\mu}\nabla G$ with $\alpha \in (0, 1)$.

Theorem (composition of averaged operators)

Let C be a nonempty, closed, convex subset of $\mathbb{E}$. For each $i \in \{1, \ldots, m\}$, let $\alpha_i \in (0, 1)$ and $\Phi_i : C \to C$ be an α_i-averaged operator. Then

$$\Phi = \Phi_m \circ \ldots \circ \Phi_1$$

is α-averaged with

$$\alpha = \frac{m}{m - 1 + \frac{1}{\max_{1 \le i \le m} \alpha_i}}.$$

<u>Proof</u>: on board.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Convex combination of averaged operators

Theorem (convex combination of averaged operators)

Let C be a nonempty, closed, convex subset of $\mathbb{E}$. For each $i \in \{1, ..., m\}$, let $\alpha_i \in (0, 1)$, $\omega_i \in (0, 1)$ and $\Phi_i : C \to \mathbb{E}$ be an α_i-averaged operator. If $\sum_{i=1}^{m} \omega_i = 1$ and $\alpha = \max_{1 \le i \le m} \alpha_i$, then

$$\Phi = \sum_{i=1}^{m} \omega_i \Phi_i$$

is α-averaged.

Proof: as exercise.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Convergence of averaged-operator iterations

Theorem (Krasnoselskii)

Let C be a nonempty, closed, convex subset of $\mathbb{E}$, and $u^{k+1} = \Phi(u^k)$ for $k = 0, 1, 2, \ldots$ where $\Phi : C \to C$ satisfies:

1. Φ is α-averaged for some $\alpha \in (0, 1)$.
2. Φ has at least one fixed point.

Then $\{u^k\}$ converges to a fixed point of Φ.

<u>Proof</u>: on board.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Extension of Krasnoselskii Theorem

> **Theorem (Krasnoselskii-Mann)**
>
> Let C be a nonempty, closed, convex subset of $\mathbb{E}$, and $u^{k+1} = (1-\tau^k)u^k + \tau^k \Psi(u^k)$ for $k = 0, 1, 2, \ldots$ where $\{\tau^k\} \subset [0, 1]$ s.t.
>
> $$\sum_{k=0}^{\infty} \tau^k (1-\tau^k) = \infty,$$
>
> and $\Psi : C \to C$ satisfies:
>
> 1. Ψ is nonexpansive.
> 2. Ψ has at least one fixed point.
>
> Then the sequence $\{u^k\}$ converges to a fixed point of Ψ.
>
> Proof: on board.

Extension of Krasnoselskii Theorem

Theorem (Krasnoselskii-Mann)

Let C be a nonempty, closed, convex subset of $\mathbb{E}$, and $u^{k+1} = (1 - \tau^k)u^k + \tau^k \Psi(u^k)$ for $k = 0, 1, 2, \ldots$ where $\{\tau^k\} \subset [0, 1]$ s.t.

$$\sum_{k=0}^{\infty} \tau^k (1 - \tau^k) = \infty,$$

and $\Psi : C \to C$ satisfies:

1. Ψ is nonexpansive.
2. Ψ has at least one fixed point.

Then the sequence $\{u^k\}$ converges to a fixed point of Ψ.

Proof: on board.

Remarks

1. Condition $\sum_{k=0}^{\infty} \tau^k(1 - \tau^k) = \infty$ is fulfilled if $\{\tau^k\} \subset [\epsilon, 1 - \epsilon]$ for some $\epsilon \in (0, 1/2]$.
2. Decay rate of fixed-point residual: $\|u^{k+1} - u^k\| = o(1/\sqrt{k})$.

Convergence in infinite dimensional space

Theorem (Krasnoselskii in Hilbert space)

Let C be a nonempty, closed, convex subset of a (real) Hilbert space $\mathbb{H}$, and $u^{k+1} = \Phi(u^k)$ for $k = 0, 1, 2, \ldots$ where $\Phi : C \to C$ satisfies:

1. Φ is α-averaged for some $\alpha \in (0, 1)$.
2. Φ has at least one fixed point.

Then $\{u^k\}$ converges *weakly* to a fixed point of Φ.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Convergence in infinite dimensional space

Theorem (Krasnoselskii in Hilbert space)

Let C be a nonempty, closed, convex subset of a (real) Hilbert space $\mathbb{H}$, and $u^{k+1} = \Phi(u^k)$ for $k = 0, 1, 2, \ldots$ where $\Phi : C \to C$ satisfies:

1. Φ is α-averaged for some $\alpha \in (0, 1)$.
2. Φ has at least one fixed point.

Then $\{u^k\}$ converges *weakly* to a fixed point of Φ.

Proof: ... $\Rightarrow \|u^{k+1} - \bar{u}\|^2 \leq \|u^0 - \bar{u}\|^2 - \frac{1-\alpha}{\alpha} \sum_{l=0}^{k} \|(I - \Phi)(u^l)\|^2$
$\Rightarrow$ (i) $\|u^k - \bar{u}\| \searrow c \geq 0$; (ii) $\sum_{k=0}^{\infty} \|(I - \Phi)(u^k)\|^2 < \infty$.
(i) $\Rightarrow \{u^k\}$ converges weakly to $u^* \in C$ along a subsequence;
(ii) & "demiclosedness principle" $\Rightarrow u^* - \Phi(u^*) = 0$. $\Rightarrow$ ... $\square$

Lemma (demiclosedness principle)

Let C be a nonempty, closed, convex subset of a (real) Hilbert space $\mathbb{H}$, and $\Phi : C \to \mathbb{H}$ be nonexpansive. For any sequence $\{u^k\} \subset C$ s.t. $\{u^k\}$ weakly converges to $u \in C$ and $u^k - \Phi(u^k)$ strongly converges to $v \in \mathbb{H}$, we have $u - \Phi(u) = v$.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Linear convergence under strong monotonicity

- Recall the customized proximal iteration:

$$0 \in M(u^{k+1} - u^k) + R(u^{k+1}),$$

where M is spd matrix, R is (maximal) monotone operator.

Linear convergence under strong monotonicity

- Recall the customized proximal iteration:

$$0 \in M(u^{k+1} - u^k) + R(u^{k+1}),$$

where M is spd matrix, R is (maximal) monotone operator.

- Let $u^* = \lim_{k\to\infty} u^k$, $0 \in R(u^*)$, and $\xi^{k+1} \in R(u^{k+1})$ s.t.

$$\begin{aligned} 0 &= \left\langle u^{k+1} - u^*, u^{k+1} - u^k \right\rangle_M + \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle \\ &= \frac{1}{2}\|u^{k+1} - u^*\|_M^2 - \frac{1}{2}\|u^k - u^*\|_M^2 + \frac{1}{2}\|u^{k+1} - u^k\|_M^2 \\ &\quad + \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle . \end{aligned}$$

Linear convergence under strong monotonicity

- Recall the customized proximal iteration:

$$0 \in M(u^{k+1} - u^k) + R(u^{k+1}),$$

where M is spd matrix, R is (maximal) monotone operator.

- Let $u^* = \lim_{k\to\infty} u^k$, $0 \in R(u^*)$, and $\xi^{k+1} \in R(u^{k+1})$ s.t.

$$\begin{aligned} 0 &= \left\langle u^{k+1} - u^*, u^{k+1} - u^k \right\rangle_M + \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle \\ &= \frac{1}{2}\|u^{k+1} - u^*\|_M^2 - \frac{1}{2}\|u^k - u^*\|_M^2 + \frac{1}{2}\|u^{k+1} - u^k\|_M^2 \\ &\quad + \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle . \end{aligned}$$

- Previously, we only assume R is monotone

$$\begin{aligned} &\Rightarrow \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle \geq 0 \\ &\Rightarrow \frac{1}{2}\|u^{k+1} - u^*\|_M^2 \leq \frac{1}{2}\|u^k - u^*\|_M^2 - \frac{1}{2}\|u^{k+1} - u^k\|_M^2. \end{aligned}$$

- Next we shall assume R is “strongly monotone”.

Linear convergence under strong monotonicity

Strongly monotone operator

- R is said μ-**strongly monotone** if $R - \mu I$ is monotone.
- For proper, convex, lsc function J, ∂J is μ-strongly monotone iff J is μ-strongly convex, i.e., $J - \frac{\mu}{2}\|\cdot\|^2$ is convex.

Linear convergence under strong monotonicity

Strongly monotone operator

- R is said μ-**strongly monotone** if $R - \mu I$ is monotone.
- For proper, convex, lsc function J, ∂J is μ-strongly monotone iff J is μ-strongly convex, i.e., $J - \frac{\mu}{2}\|\cdot\|^2$ is convex.

- R is μ-strongly monotone

$$
\begin{aligned}
&\Rightarrow \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle \geq \mu \|u^{k+1} - u^*\|^2 \\
&\Rightarrow \left(\frac{1}{2} + \frac{\mu}{\lambda_{\max}(M)} \right) \|u^{k+1} - u^*\|_M^2 \\
&\qquad \leq \frac{1}{2}\|u^{k+1} - u^*\|_M^2 + \mu\|u^{k+1} - u^*\|^2 \leq \frac{1}{2}\|u^k - u^*\|_M^2 \\
&\Rightarrow \|u^{k+1} - u^*\|_M \leq \sqrt{\frac{1}{1 + 2\mu/\lambda_{\max}(M)}} \|u^k - u^*\|_M.
\end{aligned}
$$

Linear convergence under strong monotonicity

Strongly monotone operator

- R is said μ-**strongly monotone** if $R - \mu I$ is monotone.
- For proper, convex, lsc function J, ∂J is μ-strongly monotone iff J is μ-strongly convex, i.e., $J - \frac{\mu}{2}\|\cdot\|^2$ is convex.

- R is μ-strongly monotone

$$
\begin{aligned}
&\Rightarrow \left\langle u^{k+1} - u^*, \xi^{k+1} - 0 \right\rangle \geq \mu \|u^{k+1} - u^*\|^2 \\
&\Rightarrow \left(\frac{1}{2} + \frac{\mu}{\lambda_{\max}(M)}\right) \|u^{k+1} - u^*\|_M^2 \\
&\qquad \leq \frac{1}{2}\|u^{k+1} - u^*\|_M^2 + \mu\|u^{k+1} - u^*\|^2 \leq \frac{1}{2}\|u^k - u^*\|_M^2 \\
&\Rightarrow \|u^{k+1} - u^*\|_M \leq \sqrt{\frac{1}{1 + 2\mu/\lambda_{\max}(M)}}\|u^k - u^*\|_M.
\end{aligned}
$$

- Recall in PDHG:

$$
R = \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix}.
$$

R is μ-strongly monotone $\Leftrightarrow$ G, F^* are μ-strongly convex;
F^* is μ-strongly convex $\Leftrightarrow$ F is $\frac{1}{\mu}$-Lipschitz differentiable.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Acceleration Techniques

Tao Wu
Zhenzhang Ye

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Outline of the section

1. Accelerating gradient step:
   - Second-order methods (Newton).
   - Multistep methods:
     - Heavy-ball method (Polyak).
     - Accelerated gradient method (Nesterov).
   - Embedding acceleration into proximal algorithms.
2. Preconditioning proximal algorithms:
   - Preconditioned PDHG algorithm.
   - Diagonal preconditioners (Pock/Chambolle).
   - Application to problems on weighted graphs.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Newton's method

- Consider minimizing $J : \mathbb{E} \to \mathbb{R}$. J is convex and twice continuously differentiable.
- Classical Newton method:

$$d^k = -[\nabla^2 J(u^k)]^{-1} \nabla J(u^k), \quad u^{k+1} = u^k + d^k.$$

- ..., which minimizes the local quadratic model:

$$d^k = \arg\min_d J(u^k) + \left\langle \nabla J(u^k), d \right\rangle + \frac{1}{2} \left\langle d, \nabla^2 J(u^k) d \right\rangle.$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Newton's method

- Consider minimizing $J : \mathbb{E} \to \mathbb{R}$. J is convex and twice continuously differentiable.
- Classical Newton method:

$$d^k = -[\nabla^2 J(u^k)]^{-1} \nabla J(u^k), \quad u^{k+1} = u^k + d^k.$$

- ..., which minimizes the local quadratic model:

$$d^k = \arg\min_d J(u^k) + \left\langle \nabla J(u^k), d \right\rangle + \frac{1}{2} \left\langle d, \nabla^2 J(u^k) d \right\rangle.$$

- Local quadratic convergence near u^*, where $\nabla J(u^*) = 0$ and $\nabla^2 J(u^*)$ is spd:

$$\begin{aligned}
&\|u^{k+1} - u^*\| = \|u^k - u^* - [\nabla^2 J(u^k)]^{-1} \nabla J(u^k)\| \\
&\leq \|[\nabla^2 J(u^k)]^{-1}\| \|\nabla^2 J(u^k)(u^k - u^*) - (\nabla J(u^k) - \nabla J(u^*))\| \\
&= O(\|u^k - u^*\|^2).
\end{aligned}$$

- Can we use Newton step in proximal gradient method?

Proximal Newton method

$$\min_{u \in \mathbb{E}} F(u) + G(u),$$

where F is convex (possibly non-differentiable), G is convex and twice continuously differentiable.

Proximal Newton method

Initialize $u^0 \in \mathbb{E}$. Iterate with $k = 0, 1, 2, ...$

1. $d^k = \arg\min_d F(u^k + d) + \langle \nabla G(u^k), d \rangle + \frac{1}{2} \langle d, \nabla^2 G(u^k) d \rangle$.
2. $u^{k+1} = u^k + d^k$.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Proximal Newton method

$$\min_{u \in \mathbb{E}} F(u) + G(u),$$

where F is convex (possibly non-differentiable), G is convex and twice continuously differentiable.

Proximal Newton method

Initialize $u^0 \in \mathbb{E}$. Iterate with $k = 0, 1, 2, \ldots$

1. $d^k = \arg\min_d F(u^k + d) + \langle \nabla G(u^k), d \rangle + \frac{1}{2} \langle d, \nabla^2 G(u^k) d \rangle$.
2. $u^{k+1} = u^k + d^k$.

Theorem (local quadratic convergence of proximal Newton)

The proximal Newton method converges locally quadratically to the (global) minimizer u^* if $\nabla^2 G(u^*)$ is spd.

Proof: on board.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Proximal Newton method

$$\min_{u \in \mathbb{E}} F(u) + G(u),$$

where F is convex (possibly non-differentiable), G is convex and twice continuously differentiable.

Proximal Newton method

Initialize $u^0 \in \mathbb{E}$. Iterate with $k = 0, 1, 2, \dots$

1. $d^k = \arg\min_d F(u^k + d) + \langle \nabla G(u^k), d \rangle + \frac{1}{2} \langle d, \nabla^2 G(u^k) d \rangle$.
2. $u^{k+1} = u^k + d^k$.

Theorem (local quadratic convergence of proximal Newton)

The proximal Newton method converges locally quadratically to the (global) minimizer u^* if $\nabla^2 G(u^*)$ is spd.

Proof: on board.

Remark

1. Ensure global convergence via backtracking line search.
2. Computation of d^k can be involved even if prox_F is easy.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Heavy-ball (momentum) acceleration

Heavy-ball method

Assume J is convex and differentiable. Initialize $u^0 \in \mathbb{E}$ and $u^{-1} = u^0$. Iterate with step sizes $\tau, \theta > 0$:

$$u^{k+1} = u^k - \tau \nabla J(u^k) + \theta(u^k - u^{k-1}).$$

Heavy-ball (momentum) acceleration

Heavy-ball method

Assume J is convex and differentiable. Initialize $u^0 \in \mathbb{E}$ and $u^{-1} = u^0$. Iterate with step sizes $\tau, \theta > 0$:

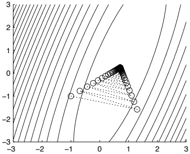
$$u^{k+1} = u^k - \tau \nabla J(u^k) + \theta(u^k - u^{k-1}).$$

- Invented by [Polyak, 1964]; $u^k - u^{k-1}$ is the *momentum*.
- Discretization of the 2nd-order ODE:

$$\theta \ddot{u} + (1-\theta)\dot{u} + \tau \nabla J(u) = 0.$$

- Alternative form implemented in machine learning:

$$v^{k+1} = \theta v^k - \tau \nabla J(u^k),$$
$$u^{k+1} = u^k + v^{k+1}.$$



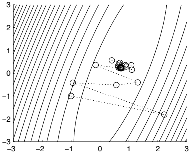


Figure: gradient descent (left) vs. heavy ball (right).

Heavy-ball method

- Quantitative analysis of heavy-ball method:

$$u^{k+1} = u^k - \tau \nabla J(u^k) + \theta(u^k - u^{k-1}).$$

$$\begin{bmatrix} u^{k+1} - u^* \\ u^k - u^* \end{bmatrix} = \begin{bmatrix} u^k + \theta(u^k - u^{k-1}) - u^* - \tau(\nabla J(u^k) - \nabla J(u^*)) \\ u^k - u^* \end{bmatrix}$$

$$= \begin{bmatrix} u^k + \theta(u^k - u^{k-1}) - u^* - \tau \nabla^2 J(\widetilde{u}^k)(u^k - u^*) \\ u^k - u^* \end{bmatrix} \quad (\widetilde{u}^k \in [u^k, u^*])$$

$$= \begin{bmatrix} (1+\theta)I - \tau \nabla^2 J(\widetilde{u}^k) & -\theta I \\ I & 0 \end{bmatrix} \begin{bmatrix} u^k - u^* \\ u^{k-1} - u^* \end{bmatrix} =: A^k \begin{bmatrix} u^k - u^* \\ u^{k-1} - u^* \end{bmatrix}.$$

Heavy-ball method

- Quantitative analysis of heavy-ball method:

$$u^{k+1} = u^k - \tau \nabla J(u^k) + \theta(u^k - u^{k-1}).$$

$$\begin{bmatrix} u^{k+1} - u^* \\ u^k - u^* \end{bmatrix} = \begin{bmatrix} u^k + \theta(u^k - u^{k-1}) - u^* - \tau(\nabla J(u^k) - \nabla J(u^*)) \\ u^k - u^* \end{bmatrix}$$

$$= \begin{bmatrix} u^k + \theta(u^k - u^{k-1}) - u^* - \tau \nabla^2 J(\widetilde{u}^k)(u^k - u^*) \\ u^k - u^* \end{bmatrix} \quad (\widetilde{u}^k \in [u^k, u^*])$$

$$= \begin{bmatrix} (1+\theta)I - \tau \nabla^2 J(\widetilde{u}^k) & -\theta I \\ I & 0 \end{bmatrix} \begin{bmatrix} u^k - u^* \\ u^{k-1} - u^* \end{bmatrix} =: A^k \begin{bmatrix} u^k - u^* \\ u^{k-1} - u^* \end{bmatrix}.$$

- <u>Lemma</u>: Assume $\forall k : \mathrm{sr}(A^k) \le \rho$, then $\exists \epsilon_k \to 0^+$ s.t. $\|A^k A^{k-1} \cdots A^0\| \le (\rho + \epsilon_k)^k \ \forall k$. (sr = spectral radius)

Theorem

Assume $\forall k : \mu I \preceq \nabla^2 J(\widetilde{u}^k) \preceq LI$ for some constants $\mu, L > 0$. If $\theta \ge \max\{|1 - \sqrt{\tau\mu}|, |1 - \sqrt{\tau L}|\}^2$, then $\mathrm{sr}(A^k) = \sqrt{\theta} \ \ \forall k$.

<u>Proof</u>: on board.

- Apply the theorem with $\tau = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$, $\theta = \left(\frac{\sqrt{L/\mu}-1}{\sqrt{L/\mu}+1}\right)^2$

$\Rightarrow$ asymptotic rate $\rho = \frac{\sqrt{L/\mu}-1}{\sqrt{L/\mu}+1}$.

Nesterov's Accelerated gradient method

Minimize J that is convex and continuously differentiable.
Assume ∇J is L-Lipschitz continuous.

Accelerated gradient method

Initialize $u^0 \in \mathbb{E}$, and $u^{-1} = u^0,\ \beta^0 = 1$. Iterate with step size $0 < \tau \leq 1/L$:

1. $\beta^{k+1} = (1 + \sqrt{1 + 4(\beta^k)^2})/2,\ \theta^k = (\beta^k - 1)/\beta^{k+1}$.
2. $v^k = u^k + \theta^k(u^k - u^{k-1})$.
3. $u^{k+1} = v^k - \tau \nabla J(v^k)$.

- Originated from [Nesterov, 1983].
- The gradient is evaluated at the *extrapolated* point v^k.
- The analysis of this scheme is somewhat technical.

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Multistep proximal gradient method

We embed multistep acceleration into proximal gradient for:

$$\min_{u \in \mathbb{E}} F(u) + G(u),$$

where F is convex (possibly non-differentiable), G is convex and twice continuously differentiable, and $\mu I \preceq \nabla^2 G(\cdot) \preceq LI$.

Proximal heavy-ball method

Initialize $u^0 \in \mathbb{E}$, and set $u^{-1} = u^0,\ \tau = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2},\ \theta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$.
Iterate with $k = 0, 1, 2, ...$

$$u^{k+1} = \mathrm{prox}_{\tau F}(u^k - \tau \nabla G(u^k) + \theta(u^k - u^{k-1})).$$

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Multistep proximal gradient method

We embed multistep acceleration into proximal gradient for:

$$\min_{u \in \mathbb{E}} F(u) + G(u),$$

where F is convex (possibly non-differentiable), G is convex and twice continuously differentiable, and $\mu I \preceq \nabla^2 G(\cdot) \preceq LI$.

Proximal heavy-ball method

Initialize $u^0 \in \mathbb{E}$, and set $u^{-1} = u^0,\ \tau = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2},\ \theta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$.
Iterate with $k = 0, 1, 2, ...$

$$u^{k+1} = \text{prox}_{\tau F}(u^k - \tau \nabla G(u^k) + \theta(u^k - u^{k-1})).$$

Proximal accelerated gradient method

Initialize $u^0 \in \mathbb{E}$, and set $u^{-1} = u^0,\ \beta^0 = 1,\ 0 < \tau \leq 1/L$.
Iterate with $k = 0, 1, 2, ...$

1. $\beta^{k+1} = (1 + \sqrt{1 + 4(\beta^k)^2})/2,\ \theta^k = (\beta^k - 1)/\beta^{k+1}$.
2. $v^k = u^k + \theta^k(u^k - u^{k-1})$.
3. $u^{k+1} = \text{prox}_{\tau F}(v^k - \tau \nabla G(v^k))$.

Preconditioning iterative linear solvers

- Consider solving the linear system

$$Qu = b \quad \Leftrightarrow \quad \min_u \frac{1}{2} \langle u, Qu \rangle - \langle b, u \rangle ,$$

where $b \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$ is spd.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Preconditioning iterative linear solvers

- Consider solving the linear system

$$Qu = b \quad \Leftrightarrow \quad \min_u \frac{1}{2} \langle u, Qu \rangle - \langle b, u \rangle ,$$

  where $b \in \mathbb{R}^n,\ Q \in \mathbb{R}^{n \times n}$ is spd.
- Define the *condition number* $\kappa_Q = \dfrac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}$, then
  - Convergence rate for steepest descent: $\dfrac{\kappa_Q - 1}{\kappa_Q + 1}$.
  - Convergence rate for conjugate gradient: $\dfrac{\sqrt{\kappa_Q} - 1}{\sqrt{\kappa_Q} + 1}$.

Preconditioning iterative linear solvers

- Consider solving the linear system

$$Qu = b \quad \Leftrightarrow \quad \min_u \frac{1}{2}\langle u, Qu\rangle - \langle b, u\rangle\,,$$

  where $b \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times n}$ is spd.

- Define the *condition number* $\kappa_Q = \dfrac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}$, then
  - Convergence rate for steepest descent: $\dfrac{\kappa_Q - 1}{\kappa_Q + 1}$.
  - Convergence rate for conjugate gradient: $\dfrac{\sqrt{\kappa_Q} - 1}{\sqrt{\kappa_Q} + 1}$.
  - Preconditioning (or rescaling) with spd $M \in \mathbb{R}^{n\times n}$:

$$\begin{cases} \widehat{Q} = M^{-1/2}QM^{-1/2},\ \widehat{u} = M^{1/2}u,\ \widehat{b} = M^{-1/2}b, \\ \text{Solve: } \min_{\widehat{u}} \frac{1}{2}\langle \widehat{u}, \widehat{Q}\widehat{u}\rangle - \langle \widehat{b}, \widehat{u}\rangle, \quad \text{ideally with } \kappa_{\widehat{Q}} \ll \kappa_Q. \end{cases}$$

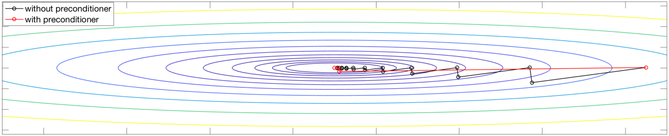


Figure: Steepest descent: without precond. vs. with precond.

Preconditioning PDHG

- Recall the saddle-point problem:

$$\max_{p} \min_{u} \langle p, Ku \rangle + G(u) - F^*(p).$$

- Recall the scaled PDHG:

$$0 \in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k), \quad \{\text{primal update}\}$$
$$0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k). \quad \{\text{dual update}\}$$

- Compact-form PDHG:

$$0 \in \begin{bmatrix} S & -K^\top \\ -K & T \end{bmatrix} \left(\begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} - \begin{bmatrix} u^k \\ p^k \end{bmatrix} \right) + \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.$$

Preconditioning PDHG

- Recall the saddle-point problem:

$$\max_{p} \min_{u} \langle p, Ku \rangle + G(u) - F^*(p).$$

- Recall the scaled PDHG:

$$0 \in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k), \quad \{\text{primal update}\}$$
$$0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k). \quad \{\text{dual update}\}$$

- Compact-form PDHG:

$$0 \in \begin{bmatrix} S & -K^\top \\ -K & T \end{bmatrix} \left(\begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} - \begin{bmatrix} u^k \\ p^k \end{bmatrix} \right) + \begin{bmatrix} \partial G & K^\top \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}.$$

- Here S is primal preconditioner, T is dual preconditioner:

$$\begin{cases} \widehat{u} = S^{1/2}u, \ \widehat{p} = T^{1/2}p, \ \widehat{K} = T^{-1/2}KS^{-1/2}, \\ \widehat{G} = G \circ S^{-1/2}, \ \widehat{F} = F \circ T^{1/2}. \\ \text{Solve: } \max_{\widehat{p}} \min_{\widehat{u}} \langle \widehat{p}, \widehat{K}\widehat{u} \rangle + \widehat{G}(\widehat{u}) - \widehat{F}^*(\widehat{p}). \end{cases}$$

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Preconditioning PDHG

- Here $\color{red}{S}$ is primal preconditioner, $\color{blue}{T}$ is dual preconditioner:

$$\begin{cases} \widehat{u} = \color{red}{S}^{1/2}u, \ \widehat{p} = \color{blue}{T}^{1/2}p, \ \widehat{K} = \color{blue}{T}^{-1/2}K\color{red}{S}^{-1/2}, \\ \widehat{G} = G \circ \color{red}{S}^{-1/2}, \ \widehat{F} = F \circ \color{blue}{T}^{1/2}. \\ \text{Solve: } \max_{\widehat{p}} \min_{\widehat{u}} \langle \widehat{p}, \widehat{K}\widehat{u} \rangle + \widehat{G}(\widehat{u}) - \widehat{F}^*(\widehat{p}). \end{cases}$$

- PDHG on $(\widehat{u}, \widehat{p})$:

$$0 \in \partial \widehat{G}(\widehat{u}^{k+1}) + \widehat{K}^\top \widehat{p}^k + (\widehat{u}^{k+1} - \widehat{u}^k),$$
$$0 \in \partial \widehat{F}^*(\widehat{p}^{k+1}) - \widehat{K}(2\widehat{u}^{k+1} - \widehat{u}^k) + (\widehat{p}^{k+1} - \widehat{p}^k).$$

- Compact-form PDHG on $(\widehat{u}, \widehat{p})$:

$$0 \in \begin{bmatrix} I & -\widehat{K}^\top \\ -\widehat{K} & I \end{bmatrix} \left(\begin{bmatrix} \widehat{u}^{k+1} \\ \widehat{p}^{k+1} \end{bmatrix} - \begin{bmatrix} \widehat{u}^{k} \\ \widehat{p}^{k} \end{bmatrix} \right) + \begin{bmatrix} \partial \widehat{G} & \widehat{K}^\top \\ -\widehat{K} & \partial \widehat{F}^* \end{bmatrix} \begin{bmatrix} \widehat{u}^{k+1} \\ \widehat{p}^{k+1} \end{bmatrix}.$$

Preconditioning PDHG

- Here S is primal preconditioner, T is dual preconditioner:

$$\begin{cases} \widehat{u} = S^{1/2}u, \ \widehat{p} = T^{1/2}p, \ \widehat{K} = T^{-1/2}KS^{-1/2}, \\ \widehat{G} = G \circ S^{-1/2}, \ \widehat{F} = F \circ T^{1/2}. \\ \text{Solve: } \max\limits_{\widehat{p}} \min\limits_{\widehat{u}} \langle \widehat{p}, \widehat{K}\widehat{u} \rangle + \widehat{G}(\widehat{u}) - \widehat{F}^*(\widehat{p}). \end{cases}$$

- Compact-form PDHG on $(\widehat{u}, \widehat{p})$:

$$0 \in \begin{bmatrix} I & -\widehat{K}^\top \\ -\widehat{K} & I \end{bmatrix} \left(\begin{bmatrix} \widehat{u}^{k+1} \\ \widehat{p}^{k+1} \end{bmatrix} - \begin{bmatrix} \widehat{u}^{k} \\ \widehat{p}^{k} \end{bmatrix} \right) + \begin{bmatrix} \partial \widehat{G} & \widehat{K}^\top \\ -\widehat{K} & \partial \widehat{F}^* \end{bmatrix} \begin{bmatrix} \widehat{u}^{k+1} \\ \widehat{p}^{k+1} \end{bmatrix}.$$

Proposition

Assume S, T are spd matrices. Then

$$M_{S,T} = \begin{bmatrix} S & -K^\top \\ -K & T \end{bmatrix} \succ 0 \Leftrightarrow \begin{bmatrix} I & -\widehat{K}^\top \\ -\widehat{K} & I \end{bmatrix} \succ 0$$

$$\Leftrightarrow \ \|T^{-1/2}KS^{-1/2}\| < 1.$$

Proof: Argue with *Schur complement*.

Gradient Methods

Proximal Algorithms

Convergence Theory

Acceleration

Summary

Choices of preconditioners

- Scaled PDHG:

$$\begin{cases} 0 \in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k), \\ 0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k). \end{cases}$$

- Expectations on S and T:
  1. S and T shall fulfill $M_{S,T} \succ 0$.
  2. (Scaled) resolvents $(S + \partial G)^{-1}$ and $(T + \partial F^*)^{-1}$ are easy to compute.
  3. $\widehat{K} = T^{-1/2} K S^{-1/2}$ has smaller condition number than K.
     - The theory for why this accelerates convergence is open.
     - Empirical evidences of acceleration are observed.

- Goal: Design S and T that balance (1), (2), (3).

Gradient Methods
Proximal Algorithms
Convergence Theory
Acceleration
Summary

Diagonal preconditioner

- Diagonal preconditioners [Pock/Chambolle, 2011]:

$$S = \mathrm{diag}(\{s_j\}),\ s_j = \sum_i |K_{ij}|^{2-\theta},$$
$$T = \mathrm{diag}(\{t_i\}),\ t_i = \sum_j |K_{ij}|^{\theta},$$

  where $\theta \in [0, 2]$.

- $\widehat{K} = T^{-1/2} K S^{-1/2}$ suggests that S (resp. T) normalizes columns (resp. rows) of K by row (resp. column) sums.

Diagonal preconditioner

- Diagonal preconditioners [Pock/Chambolle, 2011]:

$$S = \operatorname{diag}(\{s_j\}),\ s_j = \sum_i |K_{ij}|^{2-\theta},$$
$$T = \operatorname{diag}(\{t_i\}),\ t_i = \sum_j |K_{ij}|^{\theta},$$

  where $\theta \in [0, 2]$.

- $\widehat{K} = T^{-1/2} K S^{-1/2}$ suggests that S (resp. T) normalizes columns (resp. rows) of K by row (resp. column) sums.

- Convergence is (almost) justified by the following result:

Proposition

Given matrix K, the diagonal preconditioners S and T above satisfy $M_{S,T} \succeq 0$.

Proof: on board.

Diagonal preconditioner

- Diagonal preconditioners [Pock/Chambolle, 2011]:

$$S = \mathrm{diag}(\{s_j\}),\ s_j = \sum_i |K_{ij}|^{2-\theta},$$
$$T = \mathrm{diag}(\{t_i\}),\ t_i = \sum_j |K_{ij}|^{\theta},$$

  where $\theta \in [0, 2]$.

- $\widehat{K} = T^{-1/2} K S^{-1/2}$ suggests that S (resp. T) normalizes columns (resp. rows) of K by row (resp. column) sums.

- Convergence is (almost) justified by the following result:

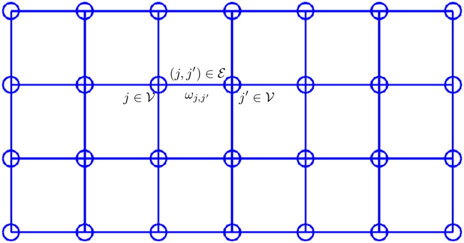
Proposition

Given matrix K, the diagonal preconditioners S and T above satisfy $M_{S,T} \succeq 0$.

Proof: on board.

- Particularly interesting for problems on *weighted graphs*...

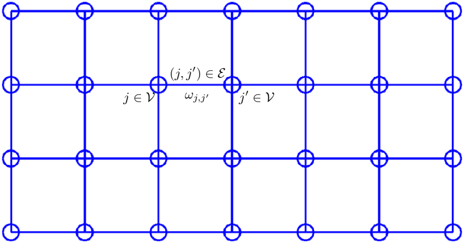
Convex optimization on weighted graphs



- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ be a weighted graph, with $\mathcal{V}$ set of vertices, $\mathcal{E}$ set of edges, $\omega : \mathcal{E} \to \mathbb{R}_+$ weight for edges.
- $\nabla \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{V}|}$ is the *incidence matrix* s.t. for each $(j, j') \in \mathcal{E}$:

$$\nabla_{(j,j'),j} = 1, \ \nabla_{(j,j'),j'} = -1, \ \nabla_{(j,j'),j''} = 0 \ \text{ whenever } j'' \notin \{j, j'\}.$$

Convex optimization on weighted graphs



- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ be a weighted graph, with $\mathcal{V}$ set of vertices, $\mathcal{E}$ set of edges, $\omega : \mathcal{E} \to \mathbb{R}_+$ weight for edges.
- $\nabla \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{V}|}$ is the *incidence matrix* s.t. for each $(j, j') \in \mathcal{E}$:

  $\nabla_{(j,j'),j} = 1, \ \nabla_{(j,j'),j'} = -1, \ \nabla_{(j,j'),j''} = 0$ whenever $j'' \notin \{j, j'\}$.
- Convex optimization on weighted graphs:

$$\min_{u:\mathcal{V}\to\mathbb{R}} F(Ku) + G(u).$$

  where $F : \mathbb{R}^{\mathcal{E}} \to \mathbb{R}$, $G : \mathbb{R}^{\mathcal{V}} \to \mathbb{R}$ are convex functions, and $K = \mathrm{diag}(\omega)\nabla$.

Example: Image segmentation on 2D grid

- Segment images represented on the 2D grid:

$$\min_{u:\mathcal{V}\to\mathbb{R}^L} \underbrace{\sum_{j\in\mathcal{V}} \Big(\delta\{u_j \in \Delta^{L-1}\} + \langle u_j, f_j\rangle\Big)}_{G(u)} + \underbrace{\alpha \sum_{l=1}^{L} \sum_{(j,j')\in\mathcal{E}} \omega_{j,j'} |u_j^l - u_{j'}^l|}_{F(Ku)},$$

- $\mathcal{V}$ contains image pixels; $\mathcal{E}$, ω are model-dependent.
- Pointwise constraint: Δ^{L-1} is the probability simplex in $\mathbb{R}^L$.
- Unary term: $f : \mathcal{V} \to \mathbb{R}^L$ is the pixelwise prediction.
- Pairwise term: $\omega_{j,j'}$ models pairwise similarities, e.g.
  - Edges are forged among spatially neighbored pixels; or
  - Use Gaussian similarity measure: $\omega_{j,j'} = \exp\left(-\frac{|j-j'|^2}{\sigma^2}\right)$.

Empirical study

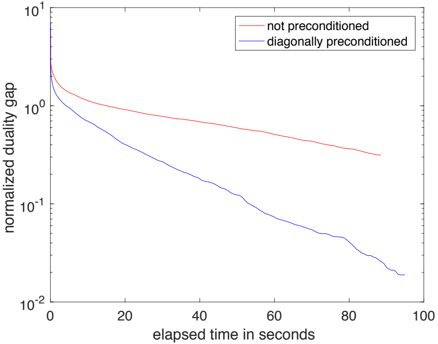
On the image segmentation example, we compare PDHG

$$\begin{cases} 0 \in \partial G(u^{k+1}) + K^\top p^k + S(u^{k+1} - u^k), \\ 0 \in \partial F^*(p^{k+1}) - K(2u^{k+1} - u^k) + T(p^{k+1} - p^k), \end{cases}$$

(i) without preconditioning and (ii) with preconditioning:

(i) $S = sI,\ T = tI,\ s = t = \|K\|$.

(ii) $S = \mathrm{diag}(\{s_j\}),\ T = \mathrm{diag}(\{t_i\}),\ s_j = \sum_i |K_{ij}|,\ t_i = \sum_j |K_{ij}|$.



What you should know from this chapter

- Gradient methods:
  - What is a descent method? (descent direction & step size)
  - How to guarantee convergence? (descent method with line search, majorize-minimize)
- Proximal algorithms:
  - How to derive proximal algorithms (FBS, ADMM, PDHG, DRS) on model problems?
  - When / how to apply a specific proximal algorithm to a specific problem?
  - What is an averaged operator?
  - How to interpret proximal algorithms as customized proximal iterations?
  - How to prove convergence of averaged-operator fixed-point iterations? (under general / special assumptions)
- Acceleration techniques (not for exam):
  - How to accelerate gradient steps in proximal algorithms? (Second-order, multistep)
  - How to precondition PDHG?
  - Some intuitions on why such acceleration techniques work.