Optimization for Machine Learning

(Lecture 2)

Suvrit Sra

Massachusetts Institute of Technology

MPI-IS Tübingen

Machine Learning Summer School, June 2017

Course materials

- My website (Teaching)
- Some references:
  - *Introductory lectures on convex optimization* – Nesterov
  - *Convex optimization* – Boyd & Vandenberghe
  - *Nonlinear programming* – Bertsekas
  - *Convex Analysis* – Rockafellar
  - *Fundamentals of convex analysis* – Urruty, Lemaréchal
  - *Lectures on modern convex optimization* – Nemirovski
  - *Optimization for Machine Learning* – Sra, Nowozin, Wright
  - *NIPS 2016 Optimization Tutorial* – Bach, Sra
- Some related courses:
  - EE227A, Spring 2013, (Sra, UC Berkeley)
  - 10-801, Spring 2014 (Sra, CMU)
  - EE364a,b (Boyd, Stanford)
  - EE236b,c (Vandenberghe, UCLA)
- Venues: NIPS, ICML, UAI, AISTATS, SIOPT, Math. Prog.

Lecture Plan

- Introduction
- Recap of convexity, sets, functions
- Recap of duality, optimality, problems
- First-order optimization algorithms and techniques
- Large-scale optimization (SGD and friends)
- Directions in non-convex optimization

ML Optimization Problems

- **Data**: n observations $(x_i, y_i)_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$
- **Prediction function**: $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **Motivating examples**:
  - **Linear predictions**: $h(x, \theta) = \theta^\top \Phi(x)$ using features $\Phi(x)$
  - **Neural networks**: $h(x, \theta) = \theta_m^\top \sigma(\theta_{m-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$
- Estimating θ parameters is an optimization problem

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^n \ell\big(y_i, h(x_i, \theta)\big) + \quad \lambda \Omega(\theta) \quad = \quad \frac{1}{n} \sum_{i=1}^n f_i(\theta)$$

ML Optimization Problems

- **Data**: n observations $(x_i, y_i)_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$
- **Prediction function**: $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **Motivating examples**:
  - **Linear predictions**: $h(x, \theta) = \theta^\top \Phi(x)$ using features $\Phi(x)$
  - **Neural networks**: $h(x, \theta) = \theta_m^\top \sigma(\theta_{m-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$
- Estimating θ parameters is an optimization problem

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^n \ell\big(y_i, h(x_i, \theta)\big) + \quad \lambda\Omega(\theta) \quad = \quad \frac{1}{n}\sum_{i=1}^n f_i(\theta)$$

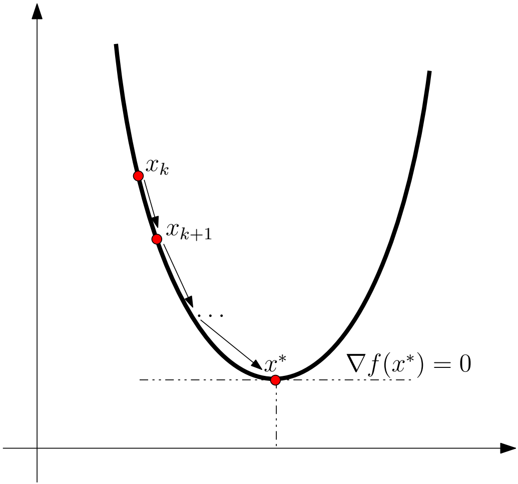
Regression: $y \in \mathbb{R}$; **Quadratic loss**: $\ell\big(y, h(x, \theta)\big) = \frac{1}{2}(y - h(x, \theta))^2$
Classf.: $y \in \{\pm 1\}$; **Logistic loss**: $\ell\big(y, h(x, \theta)\big) = \log(1 + \exp(-yh(x, \theta)))$

Descent methods

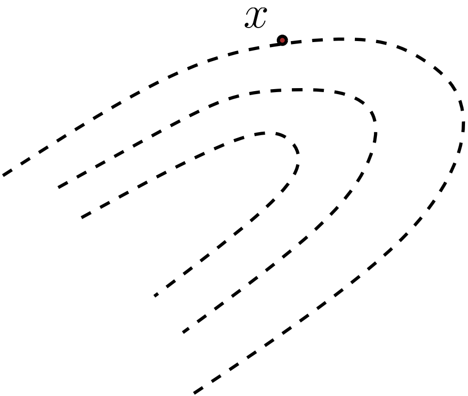
$$\min_x \quad f(x)$$

Descent methods

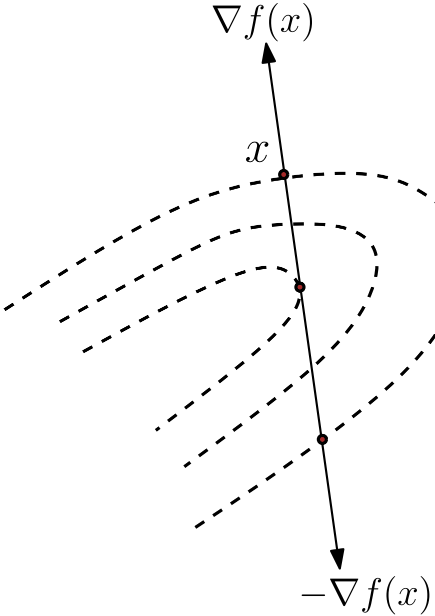
$$\min_x \quad f(x)$$



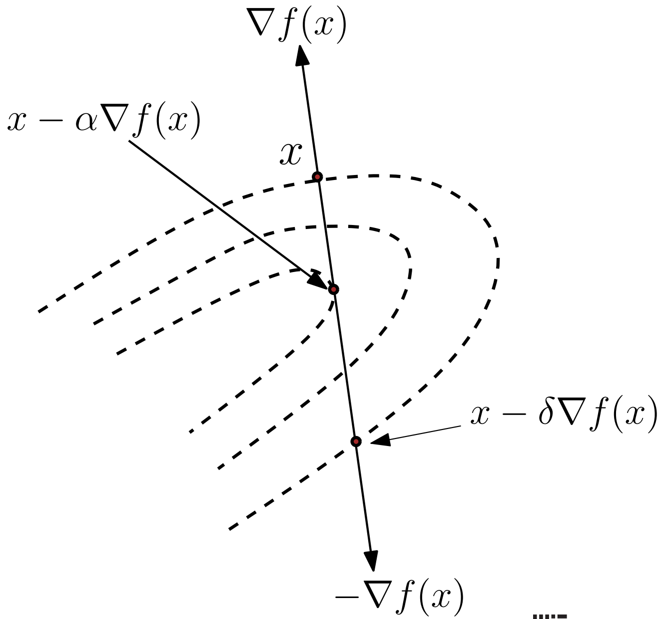
Descent methods



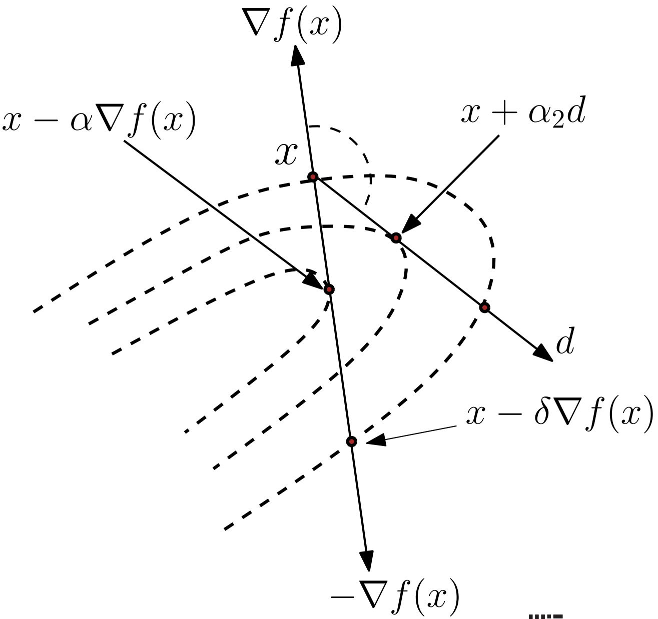
Descent methods



Descent methods



Descent methods



Iterative Algorithm

1. Start with some guess x^0;
2. For each $k = 0, 1, \ldots$
   - "Guess" α_k and d^k
   - $x^{k+1} \leftarrow x^k + \alpha_k d^k$
   - Check when to stop (e.g., if $\nabla f(x^{k+1}) \approx 0$)

(Batch) Gradient methods

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- **stepsize** $\alpha_k \geq 0$, usually ensures $f(x^{k+1}) < f(x^k)$

(Batch) Gradient methods

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- **stepsize** $\alpha_k \geq 0$, usually ensures $f(x^{k+1}) < f(x^k)$
- **Descent direction** d^k satisfies

$$\langle \nabla f(x^k), d^k \rangle < 0$$

(Batch) Gradient methods

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- **stepsize** $\alpha_k \geq 0$, usually ensures $f(x^{k+1}) < f(x^k)$
- **Descent direction** d^k satisfies

$$\langle \nabla f(x^k), d^k \rangle < 0$$

Numerous ways to select α_k and d^k

(Batch) Gradient methods

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- **stepsize** $\alpha_k \geq 0$, usually ensures $f(x^{k+1}) < f(x^k)$
- **Descent direction** d^k satisfies

$$\langle \nabla f(x^k), d^k \rangle < 0$$

Numerous ways to select α_k and d^k

Usually (batch) methods **seek monotonic descent**

$$f(x^{k+1}) < f(x^k)$$

Gradient methods – direction

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- Different choices of direction d^k
  - **Scaled gradient:** $d^k = -D^k \nabla f(x^k)$, $D^k \succ 0$
  - **Newton's method:** $(D^k = [\nabla^2 f(x^k)]^{-1})$
  - **Quasi-Newton:** $D^k \approx [\nabla^2 f(x^k)]^{-1}$
  - **Steepest descent:** $D^k = \boldsymbol{I}$
  - **Diagonally scaled:** D^k diagonal with $D^k_{ii} \approx \left(\frac{\partial^2 f(x^k)}{(\partial x_i)^2}\right)^{-1}$
  - **Discretized Newton:** $D^k = [H(x^k)]^{-1}$, H via finite-diff.

Gradient methods – direction

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots$$

- Different choices of direction d^k
  - **Scaled gradient:** $d^k = -D^k \nabla f(x^k)$, $D^k \succ 0$
  - **Newton's method:** $(D^k = [\nabla^2 f(x^k)]^{-1})$
  - **Quasi-Newton:** $D^k \approx [\nabla^2 f(x^k)]^{-1}$
  - **Steepest descent:** $D^k = \boldsymbol{I}$
  - **Diagonally scaled:** D^k diagonal with $D^k_{ii} \approx \left(\frac{\partial^2 f(x^k)}{(\partial x_i)^2}\right)^{-1}$
  - **Discretized Newton:** $D^k = [H(x^k)]^{-1}$, H via finite-diff.
  - …

Exercise: Verify that $\langle \nabla f(x^k), d^k \rangle < 0$ for above choices

Gradient methods – stepsize

- **Exact:** $\alpha_k := \operatorname{argmin}_{\alpha \geq 0} f(x^k + \alpha d^k)$

Gradient methods – stepsize

- **Exact:** $\alpha_k := \underset{\alpha \geq 0}{\operatorname{argmin}} f(x^k + \alpha d^k)$
- **Limited min:** $\alpha_k = \underset{0 \leq \alpha \leq s}{\operatorname{argmin}} f(x^k + \alpha d^k)$

Gradient methods – stepsize

- **Exact:** $\alpha_k := \operatorname{argmin}_{\alpha \geq 0} f(x^k + \alpha d^k)$
- **Limited min:** $\alpha_k = \operatorname{argmin}_{0 \leq \alpha \leq s} f(x^k + \alpha d^k)$
- **Armijo-rule**. Given **fixed** scalars, s, β, σ with $0 < \beta < 1$ and $0 < \sigma < 1$ (chosen experimentally). Set

$$\alpha_k = \beta^{m_k} s,$$

where we **try** $\beta^m s$ for $m = 0, 1, \ldots$ until **sufficient descent**

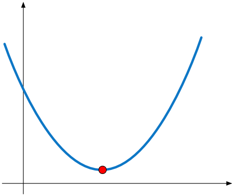
$$f(x^k) - f(x + \beta^m s d^k) \geq -\sigma \beta^m s \langle \nabla f(x^k), d^k \rangle$$

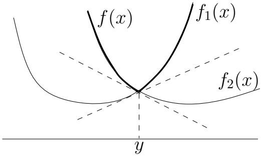
- **Constant:** $\alpha_k = 1/L$ (for suitable value of L)
- **Diminishing:** $\alpha_k \to 0$ but $\sum_k \alpha_k = \infty$.

Convergence

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

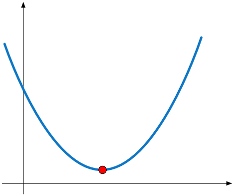


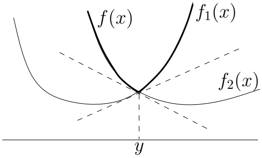


Convergence

Assumption: Lipschitz continuous gradient; denoted $f \in C^1_L$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$





- ♣ Gradient vectors of closeby points are close to each other
- ♣ Objective function has "bounded curvature"
- ♣ Speed at which gradient varies is bounded

Convergence

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

Lemma (Descent). Let $f \in C_L^1$. Then,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|_2^2$$

Theorem. Let $f \in C_L^1$ be convex, and $\{x^k\}$ is sequence generated as above, with $\alpha_k = 1/L$. Then, $f(x^{k+1}) - f(x^*) = O(1/k)$.

Remark: $f \in C_L^1$ is “good” for nonconvex too, except for $f - f^*$.

Strong convexity (faster convergence)

Assumption: Strong convexity; denote $f \in S^1_{L,\mu}$

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2$$

- A twice diff. $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

$$\forall x \in \mathbb{R}^d, \ \text{eigenvalues}\left[\nabla^2(x)\right] \succcurlyeq 0.$$

- A twice diff. $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall x \in \mathbb{R}^d, \ \text{eigenvalues}\left[\nabla^2 f(x)\right] \succcurlyeq \mu.$$

Strong convexity (faster convergence)

Assumption: Strong convexity; denote $f \in S^1_{L,\mu}$

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2$$

- A twice diff. $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

$$\forall x \in \mathbb{R}^d, \ \text{eigenvalues}\left[\nabla^2(x)\right] \geqslant 0.$$

- A twice diff. $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall x \in \mathbb{R}^d, \ \text{eigenvalues}\left[\nabla^2 f(x)\right] \geqslant \mu.$$

Condition number: $\kappa := \frac{L}{\mu} \geq 1$ influences convergence speed.

Setting $\alpha_k = \frac{2}{\mu+L}$ yields linear rate ($\mu > 0$) for gradient descent. That is, $f(x^k) - f(x^*) = O(e^{-k})$.

Strong convexity – linear rate

Theorem. If $f \in S^1_{L,\mu}$, $0 < \alpha < 2/(L+\mu)$, then the gradient method generates a sequence $\{x^k\}$ that satisfies

$$\|x^k - x^*\|_2^2 \le \left(1 - \frac{2\alpha\mu L}{\mu + L}\right)^k \|x^0 - x^*\|_2.$$

Moreover, if $\alpha = 2/(L+\mu)$ then

$$f(x^k) - f^* \le \frac{L}{2}\left(\frac{\kappa - 1}{\kappa + 1}\right)^{2k} \|x^0 - x^*\|_2^2,$$

where $\kappa = L/\mu$ is the condition number.

Gradient methods – lower bounds

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k)$$

Theorem. Lower bound I (Nesterov) For any $x^0 \in \mathbb{R}^n$, and $1 \leq k \leq \frac{1}{2}(n-1)$, there is a smooth f, s.t.

$$f(x^k) - f(x^*) \geq \frac{3L\|x^0 - x^*\|_2^2}{32(k+1)^2}$$

Gradient methods – lower bounds

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k)$$

Theorem. Lower bound I (Nesterov) For any $x^0 \in \mathbb{R}^n$, and $1 \leq k \leq \frac{1}{2}(n-1)$, there is a smooth f, s.t.

$$f(x^k) - f(x^*) \geq \frac{3L\|x^0 - x^*\|_2^2}{32(k+1)^2}$$

Theorem. Lower bound II (Nesterov). For class of smooth, strongly convex, i.e., $S^{\infty}_{L,\mu}$ ($\mu > 0$, $\kappa > 1$)

$$f(x^k) - f(x^*) \geq \frac{\mu}{2}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2k} \|x^0 - x^*\|_2^2.$$

Faster methods*

Optimal gradient methods

♠ We saw efficiency estimates for the gradient method:

$$f \in C_L^1 : \qquad f(x^k) - f^* \leq \frac{2L\|x^0 - x^*\|_2^2}{k+4}$$

$$f \in S_{L,\mu}^1 : \qquad f(x^k) - f^* \leq \tfrac{L}{2}\left(\frac{L-\mu}{L+\mu}\right)^{2k}\|x^0 - x^*\|_2^2.$$

Optimal gradient methods

- We saw efficiency estimates for the gradient method:

$$f \in C_L^1: \qquad f(x^k) - f^* \le \frac{2L\|x^0 - x^*\|_2^2}{k+4}$$

$$f \in S_{L,\mu}^1: \qquad f(x^k) - f^* \le \tfrac{L}{2}\left(\frac{L-\mu}{L+\mu}\right)^{2k}\|x^0 - x^*\|_2^2.$$

- We also saw **lower complexity bounds**

$$f \in C_L^1: \qquad f(x^k) - f(x^*) \ge \frac{3L\|x^0 - x^*\|_2^2}{32(k+1)^2}$$

$$fS_{L,\mu}^\infty: \qquad f(x^k) - f(x^*) \ge \frac{\mu}{2}\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^{2k}\|x^0 - x^*\|_2^2.$$

Optimal gradient methods

- Subgradient method upper and lower bounds

$$f(x^k) - f(x^*) \leq O(1/\sqrt{k})$$
$$f(x^k) - f(x^*) \geq \frac{LD}{2(1+\sqrt{k+1})}.$$

- Composite objective problems: proximal gradient gives same bounds as gradient methods.

Gradient with "momentum"

Polyak's method (aka heavy-ball) for $f \in S^1_{L,\mu}$

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) + \beta_k (x^k - x^{k-1})$$

Gradient with "momentum"

Polyak's method (aka heavy-ball) for $f \in S^1_{L,\mu}$

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) + \beta_k (x^k - x^{k-1})$$

- **Converges** (locally, i.e., for $\|x^0 - x^*\|_2 \leq \epsilon$) as

$$\|x^k - x^*\|_2^2 \leq \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^{2k} \|x^0 - x^*\|_2^2,$$

for $\alpha_k = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta_k = \left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2$

Nesterov's optimal gradient method

$\min_x f(x)$, where $S^1_{L,\mu}$ with $\mu \geq 0$

Nesterov's optimal gradient method

$\min_x f(x)$, where $S^1_{L,\mu}$ with $\mu \geq 0$

1. Choose $x^0 \in \mathbb{R}^n$, $\alpha_0 \in (0, 1)$
2. Let $y^0 \leftarrow x^0$; set $q = \mu/L$

Nesterov's optimal gradient method

$\min_x f(x)$, where $S^1_{L,\mu}$ with $\mu \geq 0$

1. Choose $x^0 \in \mathbb{R}^n$, $\alpha_0 \in (0, 1)$
2. Let $y^0 \leftarrow x^0$; set $q = \mu/L$
3. k-th iteration ($k \geq 0$):
   a). Compute intermediate update

$$x^{k+1} = y^k - \tfrac{1}{L}\nabla f(y^k)$$

Nesterov's optimal gradient method

$\min_x f(x)$, where $S^1_{L,\mu}$ with $\mu \geq 0$

1. Choose $x^0 \in \mathbb{R}^n$, $\alpha_0 \in (0, 1)$
2. Let $y^0 \leftarrow x^0$; set $q = \mu/L$
3. k-th iteration ($k \geq 0$):
   a). Compute intermediate update

$$x^{k+1} = y^k - \tfrac{1}{L}\nabla f(y^k)$$

   b). Compute stepsize α_{k+1} by solving

$$\alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 + q\alpha_{k+1}$$

Nesterov's optimal gradient method

$\min_x f(x)$, where $S^1_{L,\mu}$ with $\mu \geq 0$

1. Choose $x^0 \in \mathbb{R}^n$, $\alpha_0 \in (0,1)$
2. Let $y^0 \leftarrow x^0$; set $q = \mu/L$
3. k-th iteration ($k \geq 0$):
   a). Compute intermediate update

$$x^{k+1} = y^k - \tfrac{1}{L}\nabla f(y^k)$$

   b). Compute stepsize α_{k+1} by solving

$$\alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 + q\alpha_{k+1}$$

   c). Set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$
   d). Update solution estimate

$$y^{k+1} = x^{k+1} + \beta_k(x^{k+1} - x^k)$$

Optimal gradient method – rate

Theorem. Let $\{x^k\}$ be sequence generated by above algorithm. If $\alpha_0 \geq \sqrt{\mu/L}$, then

$$f(x^k) - f(x^*) \leq c_1 \min\left\{\left(1 - \sqrt{\frac{\mu}{L}}\right)^k, \frac{4L}{(2\sqrt{L} + c_2 k)^2}\right\},$$

where constants c_1, c_2 depend on α_0, L, μ.

Strongly convex case – simplification

If $\mu > 0$, select $\alpha_0 = \sqrt{\mu/L}$. The two main steps get simplified:

1. Set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$
2. $y^{k+1} = x^{k+1} + \beta_k(x^{k+1} - x^k)$

$$\alpha_k = \sqrt{\frac{\mu}{L}} \qquad \beta_k = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}, \qquad k \geq 0.$$

Strongly convex case – simplification

If $\mu > 0$, select $\alpha_0 = \sqrt{\mu/L}$. The two main steps get simplified:

1. Set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$
2. $y^{k+1} = x^{k+1} + \beta_k(x^{k+1} - x^k)$

$$\alpha_k = \sqrt{\frac{\mu}{L}} \qquad \beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}, \qquad k \geq 0.$$

Optimal method simplifies to

1. Choose $y^0 = x^0 \in \mathbb{R}^n$
2. k-th iteration ($k \geq 0$):

Strongly convex case – simplification

If $\mu > 0$, select $\alpha_0 = \sqrt{\mu/L}$. The two main steps get simplified:

1. Set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$
2. $y^{k+1} = x^{k+1} + \beta_k(x^{k+1} - x^k)$

$$\alpha_k = \sqrt{\frac{\mu}{L}} \qquad \beta_k = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}, \qquad k \geq 0.$$

Optimal method simplifies to

1. Choose $y^0 = x^0 \in \mathbb{R}^n$
2. k-th iteration ($k \geq 0$):
   a). $x^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$
   b). $y^{k+1} = x^{k+1} + \beta(x^{k+1} - x^k)$

Strongly convex case – simplification

If $\mu > 0$, select $\alpha_0 = \sqrt{\mu/L}$. The two main steps get simplified:

1. Set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$
2. $y^{k+1} = x^{k+1} + \beta_k(x^{k+1} - x^k)$

$$\alpha_k = \sqrt{\frac{\mu}{L}} \qquad \beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}, \qquad k \geq 0.$$

Optimal method simplifies to

1. Choose $y^0 = x^0 \in \mathbb{R}^n$
2. k-th iteration ($k \geq 0$):
   a). $x^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$
   b). $y^{k+1} = x^{k+1} + \beta(x^{k+1} - x^k)$

Notice similarity to Polyak's method!

Subgradient methods

Subgradient method

$$x^{k+1} = x^k - \alpha_k g^k$$

where $g^k \in \partial f(x^k)$ is **any** subgradient

Subgradient method

$$x^{k+1} = x^k - \alpha_k g^k$$

where $g^k \in \partial f(x^k)$ is **any** subgradient

Stepsize $\alpha_k > 0$ must be chosen

Subgradient method

$$x^{k+1} = x^k - \alpha_k g^k$$

where $g^k \in \partial f(x^k)$ is **any** subgradient

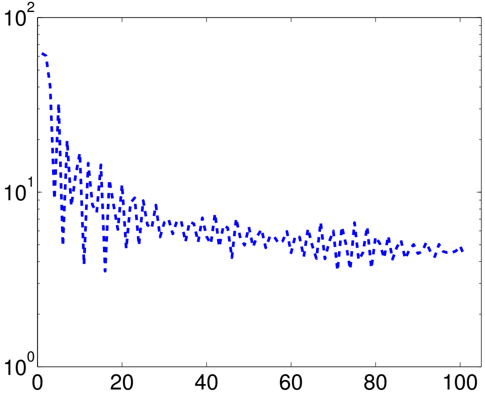
Stepsize $\alpha_k > 0$ must be chosen

- Method generates sequence $\{x^k\}_{k\geq 0}$
- Does this sequence converge to an optimal solution x^*?
- If yes, then how fast?
- What if have constraints: $x \in \mathcal{X}$?

Example: Lasso problem

$$\min \quad \tfrac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1$$

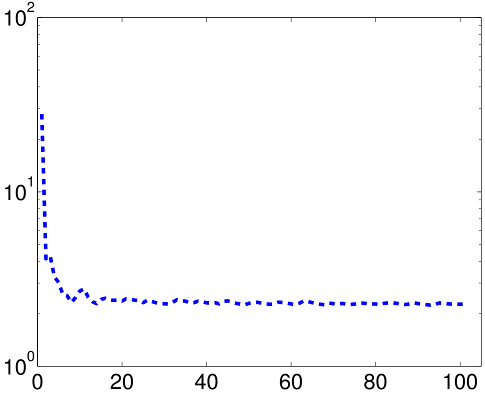
$$x^{k+1} = x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))$$



Example: Lasso problem

$$\min \quad \tfrac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

$$x^{k+1} = x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))$$



(More careful implementation)

Subgradient method – stepsizes

- **Constant** Set $\alpha_k = \alpha > 0$, for $k \geq 0$
- **Scaled constant** $\alpha_k = \alpha / \|g^k\|_2 \quad (\|x^{k+1} - x^k\|_2 = \alpha)$

Subgradient method – stepsizes

- **Constant** Set $\alpha_k = \alpha > 0$, for $k \geq 0$
- **Scaled constant** $\alpha_k = \alpha / \|g^k\|_2$ $(\|x^{k+1} - x^k\|_2 = \alpha)$
- **Square summable but not summable**

$$\sum\nolimits_k \alpha_k^2 < \infty, \qquad \sum\nolimits_k \alpha_k = \infty$$

- **Diminishing scalar**

$$\lim_k \alpha_k = 0, \qquad \sum\nolimits_k \alpha_k = \infty$$

- **Adaptive stepsizes** (not covered)

Not a descent method!
Work with best f^k so far: $f^k_{\min} := \min_{0 \leq i \leq k} f^i$

Exercise

Support vector machines

- Let $\mathcal{D} := \{(x_i, y_i) \mid x_i \in \mathbb{R}^n, y_i \in \{\pm 1\}\}$
- We wish to find $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that

$$\min_{w,b} \quad \tfrac{1}{2}\|w\|_2^2 + C\sum\nolimits_{i=1}^{m} \max[0, 1 - y_i(w^T x_i + b)]$$

- Derive and implement a subgradient method
- Plot evolution of objective function
- Experiment with different values of $C > 0$
- Plot and keep track of $f_{\min}^k := \min_{0 \le t \le k} f(x^t)$

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$
- Subgradient method $x^{k+1} = x^k - \alpha_k g^k$, where $g^k \in \partial|x^k|$.

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$
- Subgradient method $x^{k+1} = x^k - \alpha_k g^k$, where $g^k \in \partial|x^k|$.
- If $x^0 = 1$ and $\alpha_k = \frac{1}{\sqrt{k+1}} + \frac{1}{\sqrt{k+2}}$ (this stepsize is known to be optimal), then $|x^k| = \frac{1}{\sqrt{k+1}}$

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$
- Subgradient method $x^{k+1} = x^k - \alpha_k g^k$, where $g^k \in \partial|x^k|$.
- If $x^0 = 1$ and $\alpha_k = \frac{1}{\sqrt{k+1}} + \frac{1}{\sqrt{k+2}}$ (this stepsize is known to be optimal), then $|x^k| = \frac{1}{\sqrt{k+1}}$
- Thus, $O(\frac{1}{\epsilon^2})$ iterations are needed to obtain ϵ-accuracy.

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$
- Subgradient method $x^{k+1} = x^k - \alpha_k g^k$, where $g^k \in \partial|x^k|$.
- If $x^0 = 1$ and $\alpha_k = \frac{1}{\sqrt{k+1}} + \frac{1}{\sqrt{k+2}}$ (this stepsize is known to be optimal), then $|x^k| = \frac{1}{\sqrt{k+1}}$
- Thus, $O(\frac{1}{\epsilon^2})$ iterations are needed to obtain ϵ-accuracy.
- This behavior typical for the subgradient method which exhibits $O(1/\sqrt{k})$ convergence in general

Nonsmooth convergence rates

- Let $\phi(x) = |x|$ for $x \in \mathbb{R}$
- Subgradient method $x^{k+1} = x^k - \alpha_k g^k$, where $g^k \in \partial|x^k|$.
- If $x^0 = 1$ and $\alpha_k = \frac{1}{\sqrt{k+1}} + \frac{1}{\sqrt{k+2}}$ (this stepsize is known to be optimal), then $|x^k| = \frac{1}{\sqrt{k+1}}$
- Thus, $O(\frac{1}{\epsilon^2})$ iterations are needed to obtain ϵ-accuracy.
- This behavior typical for the subgradient method which exhibits $O(1/\sqrt{k})$ convergence in general

Can we do better in general?

Nonsmooth convergence rates

Theorem. (Nesterov.) Let $\mathcal{B} = \{x \mid \|x - x^0\|_2 \leq D\}$. Assume, $x^* \in \mathcal{B}$. There exists a convex function f in $C_L^0(\mathcal{B})$ (with $L > 0$), such that for $0 \leq k \leq n-1$, the lower-bound

$$f(x^k) - f(x^*) \geq \frac{LD}{2(1+\sqrt{k+1})},$$

holds for **any algorithm** that generates x^k by linearly combining the previous iterates and subgradients.

Exercise: So design problems where we can do better!

Constrained problems

Constrained optimization

$$\min \quad f(x) \qquad \text{s.t.} \quad x \in \mathcal{X}$$

Don't want to be as slow as the subgradient method

Projected subgradient method

$$x^{k+1} = P_{\mathcal{X}}(x^k - \alpha_k g^k)$$

where $g^k \in \partial f(x^k)$ is any subgradient

- **Projection:** closest feasible point

$$P_{\mathcal{X}}(y) = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \|x - y\|^2$$

- Great as long as projection is "easy"

Projected subgradient method

$$x^{k+1} = P_{\mathcal{X}}(x^k - \alpha_k g^k)$$

where $g^k \in \partial f(x^k)$ is any subgradient

- **Projection:** closest feasible point

$$P_{\mathcal{X}}(y) = \operatorname*{argmin}_{x \in \mathcal{X}} \|x - y\|^2$$

- Great as long as projection is "easy"
- Questions we may have:
  - Does it converge?
  - For which stepsizes?
  - How fast?

Examples

$$\min \quad \tfrac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$
$$\text{s.t. } x \in \mathcal{X}$$

- **Nonnegativity** $x \geq 0$
  $P_{\mathcal{X}}(z) = [z]_+$
  Update step: $x^{k+1} = [x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))]_+$

Examples

$$\min \quad \tfrac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$
$$\text{s.t. } x \in \mathcal{X}$$

- **Nonnegativity** $x \geq 0$
  $P_{\mathcal{X}}(z) = [z]_+$
  Update step: $x^{k+1} = [x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))]_+$
- ℓ_∞**-ball** $\|x\|_\infty \leq 1$
  Projection: $\min \|x - z\|^2$ s.t. $x \leq 1$ and $x \geq -1$

Examples

$$\min \quad \tfrac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1$$
$$\text{s.t. } x \in \mathcal{X}$$

- **Nonnegativity** $x \geq 0$
  $P_{\mathcal{X}}(z) = [z]_+$
  Update step: $x^{k+1} = [x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))]_+$
- **ℓ_∞-ball** $\|x\|_\infty \leq 1$
  Projection: $\min \|x-z\|^2$ s.t. $x \leq 1$ and $x \geq -1$
  this is separable, so do it coordinate-wise:
  $P_{\mathcal{X}}(z) = y$ where $y_i = \mathrm{sgn}(z_i)\min\{|z_i|, 1\}$

Examples

$$\min \quad \tfrac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$
$$\text{s.t. } x \in \mathcal{X}$$

- **Nonnegativity** $x \geq 0$
  $P_{\mathcal{X}}(z) = [z]_+$
  Update step: $x^{k+1} = [x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))]_+$
- **ℓ_∞-ball** $\|x\|_\infty \leq 1$
  Projection: $\min \|x - z\|^2$ s.t. $x \leq 1$ and $x \geq -1$
  this is separable, so do it coordinate-wise:
  $P_{\mathcal{X}}(z) = y$ where $y_i = \mathrm{sgn}(z_i)\min\{|z_i|, 1\}$
  Update step:

$$z^{k+1} = x^k - \alpha_k(A^T(Ax^k - b) + \lambda\,\mathrm{sgn}(x^k))$$
$$x_i^{k+1} = \mathrm{sgn}(z_i^{k+1})\min\{|z_i^{k+1}|, 1\}$$

Examples

- **Linear constraints** $Ax = b$ ($A \in \mathbb{R}^{n\times m}$ has rank n)

$$\begin{aligned} P_{\mathcal{X}}(y) &= y - A^\top(AA^\top)^{-1}(Ay - b) \\ &= (I - A^\top(A^\top A)^{-1}A)y + A^\top(AA^\top)^{-1}b \end{aligned}$$

Examples

- **Linear constraints** $Ax = b$ ($A \in \mathbb{R}^{n\times m}$ has rank n)

$$\begin{aligned} P_{\mathcal{X}}(y) &= y - A^\top(AA^\top)^{-1}(Ay - b) \\ &= (I - A^\top(A^\top A)^{-1}A)y + A^\top(AA^\top)^{-1}b \end{aligned}$$

Update step, using $Ax^t = b$:

$$\begin{aligned} x^{t+1} &= P_{\mathcal{X}}(x^t - \alpha_t g^t) \\ &= x^t - \alpha_t(I - A^\top(AA^\top)^{-1}A)g^t \end{aligned}$$

Examples

- **Linear constraints** $Ax = b$ ($A \in \mathbb{R}^{n\times m}$ has rank n)

$$P_{\mathcal{X}}(y) = y - A^\top(AA^\top)^{-1}(Ay - b)$$
$$= (I - A^\top(A^\top A)^{-1}A)y + A^\top(AA^\top)^{-1}b$$

Update step, using $Ax^t = b$:

$$x^{t+1} = P_{\mathcal{X}}(x^t - \alpha_t g^t)$$
$$= x^t - \alpha_t(I - A^\top(AA^\top)^{-1}A)g^t$$

- **Simplex** $x^\top 1 = 1$ and $x \geq 0$
more complex but doable in $O(n)$, similarly ℓ_1-norm ball

Subgradient method – remarks

- Why care?
  - simple
  - low-memory
  - large-scale versions possible

Subgradient method – remarks

- Why care?
  - simple
  - low-memory
  - large-scale versions possible
- Another perspective

$$x^{k+1} = \min_{x \in \mathcal{X}} \langle x, g^k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|^2$$

Mirror Descent

Subgradient method – remarks

- Why care?
  - simple
  - low-memory
  - large-scale versions possible
- Another perspective

$$x^{k+1} = \min_{x \in \mathcal{X}} \langle x, g^k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|^2$$

Mirror Descent

- Improvements using more information (heavy-ball, filtered subgradient, ...)

Subgradient method – remarks

- Why care?
  - simple
  - low-memory
  - large-scale versions possible
- Another perspective

$$x^{k+1} = \min_{x \in \mathcal{X}} \langle x, g^k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|^2$$

Mirror Descent

- Improvements using more information (heavy-ball, filtered subgradient, ...)
- Don't forget the dual
  - may be more amenable to optimization
  - duality gap?

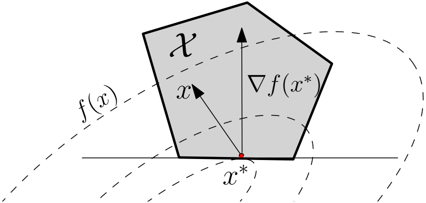
What we did not cover

- Adaptive stepsize tricks
- Space dilation methods, quasi-Newton style subgrads
- Barrier subgradient method
- Sparse subgradient method
- Ellipsoid method, center of gravity, etc. as subgradient methods
- ...

Feasible descent

$$\min \quad f(x) \quad \text{s.t. } x \in \mathcal{X}$$

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0, \qquad \forall x \in \mathcal{X}.$$



Feasible descent

$$x^{k+1} = x^k + \alpha_k d^k$$

Feasible descent

$$x^{k+1} = x^k + \alpha_k d^k$$

- d^k – **feasible direction**, i.e., $x^k + \alpha_k d^k \in \mathcal{X}$

Feasible descent

$$x^{k+1} = x^k + \alpha_k d^k$$

- d^k – **feasible direction**, i.e., $x^k + \alpha_k d^k \in \mathcal{X}$
- d^k must also be **descent direction**, i.e., $\langle \nabla f(x^k), d^k \rangle < 0$
- Stepsize α_k chosen to ensure **feasibility and descent**.

Feasible descent

$$x^{k+1} = x^k + \alpha_k d^k$$

- d^k – **feasible direction**, i.e., $x^k + \alpha_k d^k \in \mathcal{X}$
- d^k must also be **descent direction**, i.e., $\langle \nabla f(x^k), d^k \rangle < 0$
- Stepsize α_k chosen to ensure **feasibility and descent**.

Since $\mathcal{X}$ is convex, all feasible directions are of the form

$$d^k = \gamma(z - x^k), \quad \gamma > 0,$$

where $z \in \mathcal{X}$ is any feasible vector.

Feasible descent

$$x^{k+1} = x^k + \alpha_k d^k$$

- d^k – **feasible direction**, i.e., $x^k + \alpha_k d^k \in \mathcal{X}$
- d^k must also be **descent direction**, i.e., $\langle \nabla f(x^k), d^k \rangle < 0$
- Stepsize α_k chosen to ensure **feasibility and descent**.

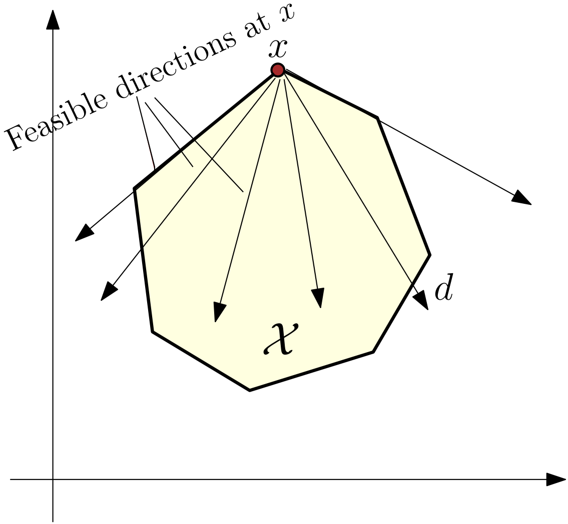
Since $\mathcal{X}$ is convex, all feasible directions are of the form

$$d^k = \gamma(z - x^k), \quad \gamma > 0,$$

where $z \in \mathcal{X}$ is any feasible vector.

$$x^{k+1} = x^k + \alpha_k(z^k - x^k), \quad \alpha_k \in (0, 1]$$

Cone of feasible directions



Frank-Wolfe / conditional gradient method

Optimality: $\langle \nabla f(x^k), z^k - x^k \rangle \geq 0$ for all $z^k \in \mathcal{X}$

Frank-Wolfe / conditional gradient method

Optimality: $\langle \nabla f(x^k), z^k - x^k \rangle \geq 0$ for all $z^k \in \mathcal{X}$
Aim: If not optimal, then generate feasible direction $d^k = z^k - x^k$ that obeys **descent condition** $\langle \nabla f(x^k), d^k \rangle < 0$.

Frank-Wolfe / conditional gradient method

Optimality: $\langle \nabla f(x^k), z^k - x^k \rangle \geq 0$ for all $z^k \in \mathcal{X}$
Aim: If not optimal, then generate feasible direction $d^k = z^k - x^k$ that obeys **descent condition** $\langle \nabla f(x^k), d^k \rangle < 0$.

Frank-Wolfe (Conditional gradient) method

- ▲ Let $z^k \in \operatorname{argmin}_{x \in \mathcal{X}} \langle \nabla f(x^k), x - x^k \rangle$
- ▲ Use different methods to select α_k
- ▲ $x^{k+1} = x^k + \alpha_k (z^k - x^k)$

Frank-Wolfe / conditional gradient method

Optimality: $\langle \nabla f(x^k), z^k - x^k \rangle \geq 0$ for all $z^k \in \mathcal{X}$
Aim: If not optimal, then generate feasible direction $d^k = z^k - x^k$ that obeys **descent condition** $\langle \nabla f(x^k), d^k \rangle < 0$.

Frank-Wolfe (Conditional gradient) method

- ▲ Let $z^k \in \operatorname{argmin}_{x \in \mathcal{X}} \langle \nabla f(x^k), x - x^k \rangle$
- ▲ Use different methods to select α_k
- ▲ $x^{k+1} = x^k + \alpha_k (z^k - x^k)$

- ♠ Practical when solving *linear* problem over $\mathcal{X}$ easy
- ♠ Very popular in machine learning over recent years
- ♠ Refinements, several variants (including nonconvex)

Composite objectives

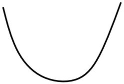
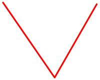
Frequently ML problems take the regularized form

$$\text{minimize}\, f(x) := \ell(x) + r(x)$$

Composite objectives

Frequently ML problems take the regularized form

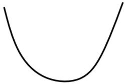
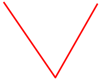
$$\text{minimize}\, f(x) := \ell(x) + r(x)$$

$\ell \in$  $+\ r \in$ 

Composite objectives

Frequently ML problems take the regularized form

$$\text{minimize } f(x) := \ell(x) + r(x)$$

$\ell \in$  $+\ r \in$ 

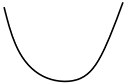
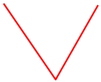
Example: $\ell(x) = \frac{1}{2}\|Ax - b\|^2$ and $r(x) = \lambda\|x\|_1$

Lasso, L1-LS, compressed sensing

Composite objectives

Frequently ML problems take the regularized form

$$\text{minimize}\, f(x) := \ell(x) + r(x)$$

$\ell \in$  $+\ r \in$ 

Example: $\ell(x) = \frac{1}{2}\|Ax - b\|^2$ and $r(x) = \lambda\|x\|_1$

Lasso, L1-LS, compressed sensing

Example: $\ell(x)$: Logistic loss, and $r(x) = \lambda\|x\|_1$

L1-Logistic regression, sparse LR

Composite objective minimization

$$\text{minimize } f(x) := \ell(x) + r(x)$$

subgradient: $x^{k+1} = x^k - \alpha^k g^k, g^k \in \partial f(x^k)$

Composite objective minimization

$$\text{minimize}\, f(x) := \ell(x) + r(x)$$

subgradient: $x^{k+1} = x^k - \alpha^k g^k, g^k \in \partial f(x^k)$

subgradient: converges slowly at rate $O(1/\sqrt{k})$

Composite objective minimization

$$\text{minimize } f(x) := \ell(x) + r(x)$$

subgradient: $x^{k+1} = x^k - \alpha^k g^k,\ g^k \in \partial f(x^k)$

subgradient: converges slowly at rate $O(1/\sqrt{k})$

but: f is *smooth* plus *nonsmooth*

we should **exploit:** smoothness of ℓ for better method!

Proximal Gradient Method

$$\min \quad f(x) \quad x \in \mathcal{X}$$

Projected (sub)gradient

$$x \leftarrow P_{\mathcal{X}}(x - \alpha \nabla f(x))$$

Proximal Gradient Method

$$\min \quad f(x) \quad x \in \mathcal{X}$$

Projected (sub)gradient

$$x \leftarrow P_{\mathcal{X}}(x - \alpha \nabla f(x))$$

$$\min \quad f(x) + h(x)$$

Proximal gradient

$$x \leftarrow \operatorname{prox}_{\alpha h}(x - \alpha \nabla f(x))$$

$\operatorname{prox}_{\alpha h}$ denotes proximity operator for h

Why? If we can compute $\operatorname{prox}_h(x)$ easily, prox-grad converges as fast gradient methods for smooth problems!

Proximity operator

Projection

$$P_{\mathcal{X}}(y) := \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \quad \tfrac{1}{2}\|x - y\|_2^2 + \mathbb{1}_{\mathcal{X}}(x)$$

Proximity operator

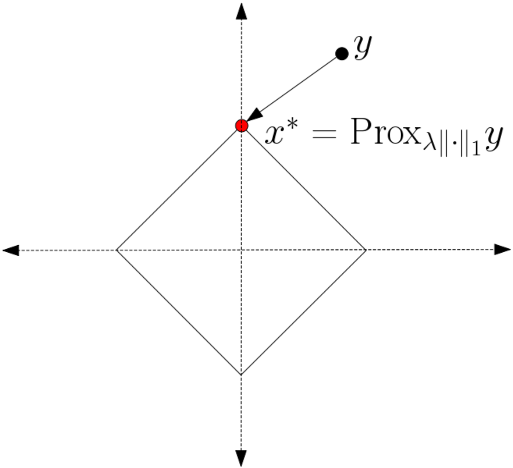
Projection

$$P_{\mathcal{X}}(y) := \operatorname*{argmin}_{x \in \mathbb{R}^n} \quad \tfrac{1}{2}\|x - y\|_2^2 + \mathbb{1}_{\mathcal{X}}(x)$$

Proximity: Replace $\mathbb{1}_{\mathcal{X}}$ by a closed convex function

$$\operatorname{prox}_r(y) := \operatorname*{argmin}_{x \in \mathbb{R}^n} \quad \tfrac{1}{2}\|x - y\|_2^2 + r(x)$$

Proximity operator



ℓ_1-norm ball of radius $\rho(\lambda)$

Proximity operators

Exercise: Let $r(x) = \|x\|_1$. Solve $\text{prox}_{\lambda r}(y)$.

$$\min_{x \in \mathbb{R}^n} \quad \tfrac{1}{2}\|x - y\|_2^2 + \lambda\|x\|_1.$$

Hint 1: The above problem decomposes into n independent subproblems of the form

$$\min_{x \in \mathbb{R}} \quad \tfrac{1}{2}(x - y)^2 + \lambda|x|.$$

Hint 2: Consider the two cases: either $x = 0$ or $x \neq 0$
Exercise: **Moreau decomposition** $y = \text{prox}_h y + \text{prox}_{h^*} y$
(notice analogy to $V = S + S^{\perp}$ in linear algebra)

How to cook-up prox-grad?

Lemma $x^* = \operatorname{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

How to cook-up prox-grad?

Lemma $x^* = \operatorname{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$0 \in \nabla f(x^*) + \partial h(x^*)$$

How to cook-up prox-grad?

Lemma $x^* = \operatorname{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial h(x^*)
\end{aligned}
$$

How to cook-up prox-grad?

Lemma $x^* = \operatorname{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial h(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial h)(x^*)
\end{aligned}
$$

How to cook-up prox-grad?

Lemma $x^* = \mathrm{prox}_{\alpha h}(x^* - \alpha\nabla f(x^*)), \forall \alpha > 0$

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha\nabla f(x^*) + \alpha\partial h(x^*) \\
x^* &\in \alpha\nabla f(x^*) + (I + \alpha\partial h)(x^*) \\
x^* - \alpha\nabla f(x^*) &\in (I + \alpha\partial h)(x^*)
\end{aligned}
$$

How to cook-up prox-grad?

Lemma $x^* = \text{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial h(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial h)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial h)(x^*) \\
x^* &= (I + \alpha \partial h)^{-1}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

How to cook-up prox-grad?

Lemma $x^* = \mathrm{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial h(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial h)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial h)(x^*) \\
x^* &= (I + \alpha \partial h)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \mathrm{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

How to cook-up prox-grad?

Lemma $x^* = \text{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*)), \forall \alpha > 0$

$$\begin{aligned}
0 &\in \nabla f(x^*) + \partial h(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial h(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial h)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial h)(x^*) \\
x^* &= (I + \alpha \partial h)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \text{prox}_{\alpha h}(x^* - \alpha \nabla f(x^*))
\end{aligned}$$

Above fixed-point eqn suggests iteration

$$x_{k+1} = \text{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k))$$

Convergence*

Proximal-gradient works, why?

$$\begin{aligned} x_{k+1} &= \operatorname{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k)) \\ x_{k+1} &= x_k - \alpha_k G_{\alpha_k}(x_k). \end{aligned}$$

Proximal-gradient works, why?

$$\begin{aligned} x_{k+1} &= \operatorname{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k)) \\ x_{k+1} &= x_k - \alpha_k G_{\alpha_k}(x_k). \end{aligned}$$

Gradient mapping: the "gradient-like object"

$$G_\alpha(x) = \frac{1}{\alpha}(x - P_{\alpha h}(x - \alpha \nabla f(x)))$$

Proximal-gradient works, why?

$$
\begin{aligned}
x_{k+1} &= \operatorname{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k)) \\
x_{k+1} &= x_k - \alpha_k G_{\alpha_k}(x_k).
\end{aligned}
$$

Gradient mapping: the "gradient-like object"

$$
G_\alpha(x) = \frac{1}{\alpha}(x - P_{\alpha h}(x - \alpha \nabla f(x)))
$$

- Our lemma shows: $G_\alpha(x) = 0$ if and only if x is optimal
- So G_α analogous to ∇f
- If x locally optimal, then $G_\alpha(x) = 0$ (nonconvex f)

Convergence analysis

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

Convergence analysis

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

- ♣ Gradient vectors of closeby points are close to each other
- ♣ Objective function has "bounded curvature"
- ♣ Speed at which gradient varies is bounded

Convergence analysis

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

- ♣ Gradient vectors of closeby points are close to each other
- ♣ Objective function has "bounded curvature"
- ♣ Speed at which gradient varies is bounded

Lemma (Descent). Let $f \in C_L^1$. Then,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|_2^2$$

Convergence analysis

Assumption: Lipschitz continuous gradient; denoted $f \in C_L^1$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

- ♣ Gradient vectors of closeby points are close to each other
- ♣ Objective function has "bounded curvature"
- ♣ Speed at which gradient varies is bounded

Lemma (Descent). Let $f \in C_L^1$. Then,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|_2^2$$

For convex f, compare with

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

Descent lemma

Proof. Since $f \in C_L^1$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Descent lemma

Proof. Since $f \in C^1_L$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \quad = \quad \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt$$

Descent lemma

Proof. Since $f \in C^1_L$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \\ |f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \right| \end{aligned}$$

Descent lemma

Proof. Since $f \in C^1_L$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$\begin{aligned}
f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \\
|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \right| \\
&\leq \int_0^1 |\langle \nabla f(z_t) - \nabla f(x), y - x \rangle| dt
\end{aligned}$$

Descent lemma

Proof. Since $f \in C^1_L$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$\begin{aligned}
f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \\
|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \right| \\
&\leq \int_0^1 |\langle \nabla f(z_t) - \nabla f(x), y - x \rangle| dt \\
&\leq \int_0^1 \|\nabla f(z_t) - \nabla f(x)\|_2 \cdot \|y - x\|_2 dt
\end{aligned}$$

Descent lemma

Proof. Since $f \in C_L^1$, by Taylor's theorem, for the vector $z_t = x + t(y-x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$\begin{aligned}
f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \\
|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \right| \\
&\leq \int_0^1 |\langle \nabla f(z_t) - \nabla f(x), y - x \rangle| dt \\
&\leq \int_0^1 \|\nabla f(z_t) - \nabla f(x)\|_2 \cdot \|y - x\|_2 dt \\
&\leq L \int_0^1 t \|x - y\|_2^2 dt
\end{aligned}$$

Descent lemma

Proof. Since $f \in C^1_L$, by Taylor's theorem, for the vector $z_t = x + t(y - x)$ we have

$$f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x \rangle dt.$$

Add and subtract $\langle \nabla f(x), y - x \rangle$ on rhs we have

$$\begin{aligned}
f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \\
|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x \rangle dt \right| \\
&\leq \int_0^1 |\langle \nabla f(z_t) - \nabla f(x), y - x \rangle| dt \\
&\leq \int_0^1 \|\nabla f(z_t) - \nabla f(x)\|_2 \cdot \|y - x\|_2 dt \\
&\leq L \int_0^1 t \|x - y\|_2^2 dt \\
&= \tfrac{L}{2} \|x - y\|_2^2.
\end{aligned}$$

Bounds $f(y)$ around x with quadratic functions

Descent lemma – corollary

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|_2^2$$

Let $y = x - \alpha G_\alpha(x)$, then

Descent lemma – corollary

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|_2^2$$

Let $y = x - \alpha G_\alpha(x)$, then

$$f(y) \leq f(x) - \alpha \langle \nabla f(x), G_\alpha(x) \rangle + \frac{\alpha^2 L}{2}\|G_\alpha(x)\|_2^2.$$

Descent lemma – corollary

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|_2^2$$

Let $y = x - \alpha G_\alpha(x)$, then

$$f(y) \leq f(x) - \alpha \langle \nabla f(x), G_\alpha(x) \rangle + \frac{\alpha^2 L}{2}\|G_\alpha(x)\|_2^2.$$

Corollary. So if $0 \leq \alpha \leq 1/L$, we have

$$f(y) \leq f(x) - \alpha \langle \nabla f(x), G_\alpha(x) \rangle + \frac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Descent lemma – corollary

$$f(y) \le f(x) + \langle \nabla f(x),\, y - x\rangle + \tfrac{L}{2}\|y - x\|_2^2$$

Let $y = x - \alpha G_\alpha(x)$, then

$$f(y) \le f(x) - \alpha\langle \nabla f(x),\, G_\alpha(x)\rangle + \frac{\alpha^2 L}{2}\|G_\alpha(x)\|_2^2.$$

Corollary. So if $0 \le \alpha \le 1/L$, we have

$$f(y) \le f(x) - \alpha\langle \nabla f(x),\, G_\alpha(x)\rangle + \frac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Lemma Let $y = x - \alpha G_\alpha(x)$. Then, for any z we have

$$f(y) + h(y) \le f(z) + h(z) + \langle G_\alpha(x),\, x - z\rangle - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Exercise: Prove! (hint: f, h are convex, $G_\alpha(x) - \nabla f(x) \in \partial h(y)$)

Convergence analysis

We've actually shown $x' = x - \alpha G_\alpha(x)$ is a descent method.
Write $\phi = f + h$; plug in $z = x$ to obtain

$$\phi(x') \leq \phi(x) - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Exercise: Why this inequality suffices to show convergence.

Convergence analysis

We've actually shown $x' = x - \alpha G_\alpha(x)$ is a descent method.
Write $\phi = f + h$; plug in $z = x$ to obtain

$$\phi(x') \leq \phi(x) - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Exercise: Why this inequality suffices to show convergence.
Use $z = x^*$ in corollary to obtain progress in terms of iterates:

$$\phi(x') - \phi^* \quad \leq \quad \langle G_\alpha(x),\, x - x^* \rangle - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2$$

Convergence analysis

We've actually shown $x' = x - \alpha G_\alpha(x)$ is a descent method.
Write $\phi = f + h$; plug in $z = x$ to obtain

$$\phi(x') \leq \phi(x) - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Exercise: Why this inequality suffices to show convergence.
Use $z = x^*$ in corollary to obtain progress in terms of iterates:

$$\begin{aligned}
\phi(x') - \phi^* &\leq \langle G_\alpha(x),\, x - x^*\rangle - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2 \\
&= \frac{1}{2\alpha}\left[2\langle \alpha G_\alpha(x),\, x - x^*\rangle - \|\alpha G_\alpha(x)\|_2^2\right] \\
&= \frac{1}{2\alpha}\left[\|x - x^*\|_2^2 - \|x - x^* - \alpha G_\alpha(x)\|_2^2\right]
\end{aligned}$$

Convergence analysis

We've actually shown $x' = x - \alpha G_\alpha(x)$ is a descent method.
Write $\phi = f + h$; plug in $z = x$ to obtain

$$\phi(x') \leq \phi(x) - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2.$$

Exercise: Why this inequality suffices to show convergence.
Use $z = x^*$ in corollary to obtain progress in terms of iterates:

$$\begin{aligned}
\phi(x') - \phi^* &\leq \langle G_\alpha(x),\, x - x^*\rangle - \tfrac{\alpha}{2}\|G_\alpha(x)\|_2^2 \\
&= \frac{1}{2\alpha}\left[2\langle \alpha G_\alpha(x),\, x - x^*\rangle - \|\alpha G_\alpha(x)\|_2^2\right] \\
&= \frac{1}{2\alpha}\left[\|x - x^*\|_2^2 - \|x - x^* - \alpha G_\alpha(x)\|_2^2\right] \\
&= \frac{1}{2\alpha}\left[\|x - x^*\|_2^2 - \|x' - x^*\|_2^2\right].
\end{aligned}$$

Convergence rate

Set $x \leftarrow x_k$, $x' \leftarrow x_{k+1}$, and $\alpha = 1/L$. Then add

Convergence rate

Set $x \leftarrow x_k$, $x' \leftarrow x_{k+1}$, and $\alpha = 1/L$. Then add

$$\sum_{i=1}^{k+1} (\phi(x_i) - \phi^*) \quad \leq \quad \tfrac{L}{2} \sum_{i=1}^{k+1} \left[\|x_i - x^*\|_2^2 - \|x_{i+1} - x^*\|_2^2\right]$$

Convergence rate

Set $x \leftarrow x_k$, $x' \leftarrow x_{k+1}$, and $\alpha = 1/L$. Then add

$$\begin{aligned}\sum_{i=1}^{k+1}(\phi(x_i) - \phi^*) &\leq \frac{L}{2}\sum_{i=1}^{k+1}\left[\|x_i - x^*\|_2^2 - \|x_{i+1} - x^*\|_2^2\right] \\ &= \frac{L}{2}\left[\|x_1 - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2\right]\end{aligned}$$

Convergence rate

Set $x \leftarrow x_k$, $x' \leftarrow x_{k+1}$, and $\alpha = 1/L$. Then add

$$\begin{aligned}
\sum_{i=1}^{k+1} (\phi(x_i) - \phi^*) &\leq \tfrac{L}{2} \sum_{i=1}^{k+1} \left[\|x_i - x^*\|_2^2 - \|x_{i+1} - x^*\|_2^2\right] \\
&= \tfrac{L}{2} \left[\|x_1 - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2\right] \\
&\leq \tfrac{L}{2} \|x_1 - x^*\|_2^2.
\end{aligned}$$

Convergence rate

Set $x \leftarrow x_k$, $x' \leftarrow x_{k+1}$, and $\alpha = 1/L$. Then add

$$\begin{aligned}\sum_{i=1}^{k+1} (\phi(x_i) - \phi^*) &\le \tfrac{L}{2} \sum_{i=1}^{k+1} \left[\|x_i - x^*\|_2^2 - \|x_{i+1} - x^*\|_2^2\right] \\ &= \tfrac{L}{2} \left[\|x_1 - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2\right] \\ &\le \tfrac{L}{2} \|x_1 - x^*\|_2^2.\end{aligned}$$

Since $\phi(x_k)$ is a decreasing sequence, it follows that

$$\phi(x_{k+1}) - \phi^* \le \frac{1}{k+1} \sum_{i=1}^{k+1} (\phi(x_i) - \phi^*) \le \frac{L}{2(k+1)} \|x_1 - x^*\|_2^2.$$

This is the well-known $O(1/k)$ rate.

- But for C_L^1 convex functions, optimal rate is $O(1/k^2)$!

Accelerated Proximal Gradient

$$\min \phi(x) = f(x) + h(x)$$

Let $x^0 = y^0 \in \operatorname{dom} h$. For $k \geq 1$:

$$x^k = \operatorname{prox}_{\alpha_k h}(y^{k-1} - \alpha_k \nabla f(y^{k-1}))$$
$$y^k = x_k + \frac{k-1}{k+2}(x^k - x^{k-1}).$$

Framework due to: Nesterov (1983, 2004); also Beck, Teboulle (2009).

Simplified analysis: Tseng (2008).

- Uses extra "memory" for interpolation
- Same computational cost as ordinary prox-grad
- Convergence rate theoretically optimal

$$\phi(x^k) - \phi^* \leq \frac{2L}{(k+1)^2}\|x^0 - x^*\|_2^2.$$

Proximal methods – cornucopia

- Douglas Rachford splitting
- ADMM (special case of DR on dual)
- Proximal-Dykstra
- Proximal methods for $f_1 + f_2 + \cdots + f_n$
- Peaceman-Rachford
- Proximal quasi-Newton, Newton
- Many other variation...