Convex Optimization Theory and Applications

Topic 19 - Subgradient Methods and ADMM

Li Li

Department of Automation
Tsinghua University

Fall, 2009-2021.

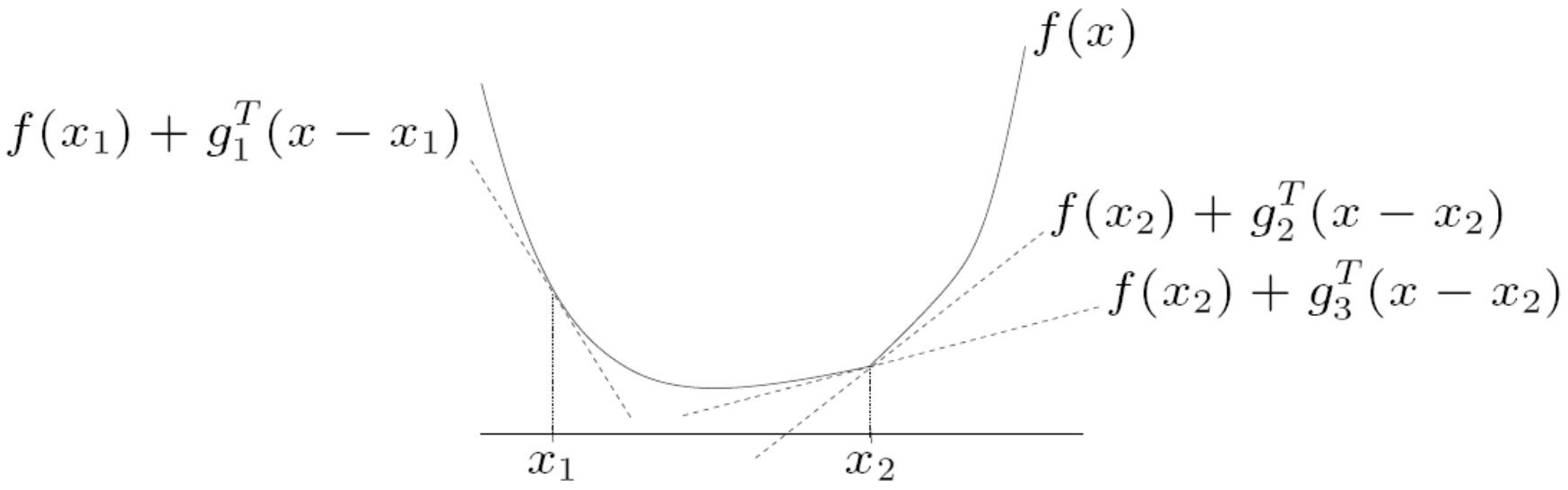
19.0. Outline

19.1. Subgradient

19.1.1 Definition

g is a subgradient of f (not necessarily convex) at x if

$f(y) \geq f(x) + g^T(y - x)$ for all y



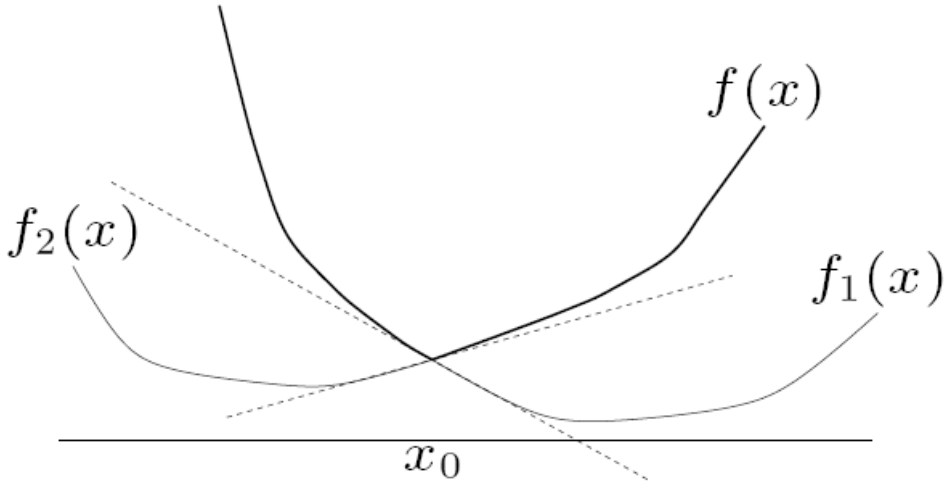
$(g, -1)$ supports epi f at $(x, f(x))$

19.1. Subgradient

19.1.1 Definition

$f(x)+g^T(y-x)$ is a global (affine) underestimator of f

If f is convex and differentiable, $\nabla f(x)$ is a subgradient of f at x



19.1. Subgradient

19.1.1 Definition

$f = \max\{f_1, f_2\}$, with f_1 , f_2 are convex and differentiable, we have the subgradient as

$f_1(x_0) > f_2(x_0)$: unique subgradient $g = \nabla f_1(x_0)$

$f_2(x_0) > f_1(x_0)$: unique subgradient $g = \nabla f_2(x_0)$

$f_1(x_0) = f_2(x_0)$: subgradients form a line segment $[\nabla f_1(x_0), \nabla f_2(x_0)]$

Why unique?

19.1. Subgradient

19.1.1 Definition

Subgradient method is important, if f is not differentiable

set of all subgradients of f at x is called subdifferential of f at x, denoted as $\partial f(x)$ which is a closed convex set (can be empty)

If f is convex, $\partial f(x)$ is nonempty (why?)

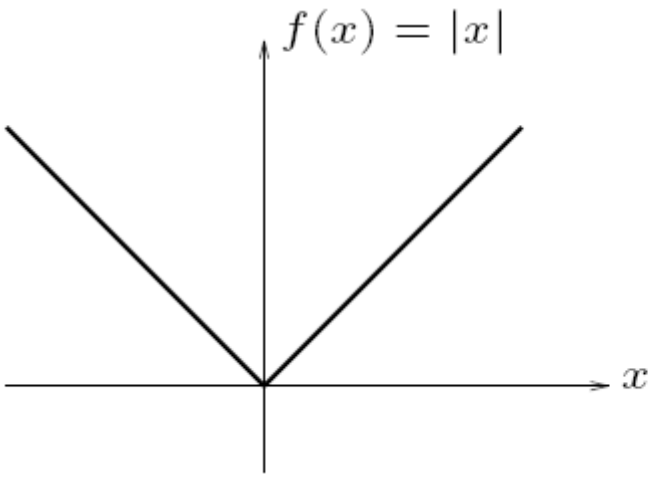
$\partial f(x) = \{\nabla f(x)\}$, if f is differentiable at x

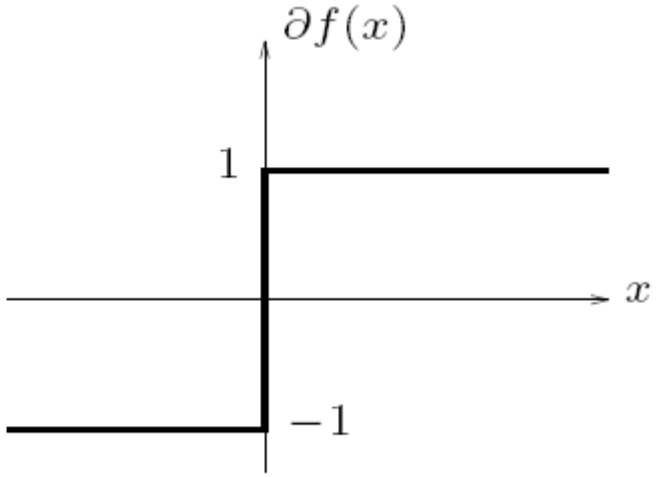
If $\partial f(x) = \{g\}$, then f is differentiable at x and $g = \nabla f(x)$

19.1. Subgradient

19.1.1 Definition

For example, consider $f(x) = |x|$





19.1. Subgradient

19.1.1 Definition

weak subgradient calculus: finding one subgradient $g \in \partial f(x)$

strong subgradient calculus: finding the whole subdifferential $\partial f(x)$, i.e., all subgradient of f at x

many algorithms for nondifferentiable convex optimization require only one subgradient at each step, so weak calculus suffices. But some algorithms, optimality conditions, etc., need whole subdifferential

Usually if we know $f(x)$, we can compute a $g \in \partial f(x)$

19.1. Subgradient

19.1.1 Definition

scaling: $\partial(\alpha f) = \alpha \partial f$, if $\alpha > 0$

addition: $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$ (RHS is the addition of sets)

affine transformation of variables: if $g(x) = f(Ax + b)$, then $\partial g(x) = A^T \partial f(Ax + b)$

finite pointwise maximum: if $f = \max_{i=1,\cdots,m} f_i$, then

$$\partial f(x) = Co \bigcup \{\partial f_i(x) \mid f_i(x) = f(x)\} \qquad (19.1)$$

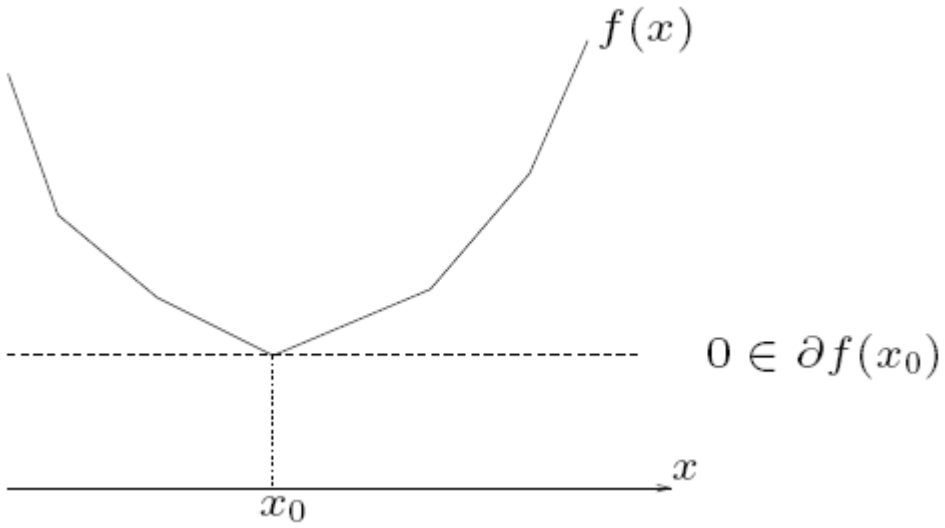
i.e., convex hull of union of subdifferentials of 'active' functions at x

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

Optimality conditions for unconstrained problem: for f convex, differentiable, we have

$$f(x^*) = \inf_x f(x) \quad \Leftrightarrow \quad 0 = \nabla f(x^*) \tag{19.2}$$



19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

generalization to nondifferentiable convex f :

$$f(x^*) = \inf_x f(x) \quad \Leftrightarrow \quad 0 \in \partial f(x^*) \tag{19.3}$$

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

generalization to nondifferentiable convex f :

$$f(x^*) = \inf_x f(x) \quad \Leftrightarrow \quad 0 \in \partial f(x^*) \tag{19.4}$$

Brieft proof: x^* is optimal if and only if $f(x) \geq f(x^*)$ for all x, or equivalently

$$f(x) \geq f(x^*) + 0^T (x - x^*) \text{ for all } x \tag{19.5}$$

Thus, x^* is optimal if and only if $0 \in \partial f(x^*)$

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

For the piecewise linear optimization $f(x) = \max_{i=1,\cdots,m} (a_i^T x + b_i)$,

the subdifferential is $\partial f(x) = Co\{a_i \mid a_i^T x + b_i = f(x)\}$.

Thus, x^* minimizes f if and only if there is a λ such that

$$\lambda \geq 0, \quad 1^T \lambda = 1, \quad \sum_{i=1}^{m} \lambda_i a_i = 0$$

where $\lambda_i = 0$, if $a_i^T x^* + b_i < f(x^*)$

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

Another interpretation is the KKT conditions for the epigraph

minimize t

subject to $a_i^T x + b_i \le t, \quad i = 1, \cdots, m$

The dual should be

maximize $b^T \lambda$

subject to $\lambda \ge 0, \quad A^T \lambda = 0, \quad 1^T \lambda = 1$

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

Consider convex programming problem

minimize $f_0(x)$

subject to $C: f_i(x) \le 0, \quad i = 1, \cdots, m$

Optimality conditions for this unconstrained problem is

Theorem 2.2: a vector $x^* \in C$ minimizes f over the set C if and only if there exists a subgradient $d \in \partial f(x^*)$ such that

$$d^T(x - x^*) \ge 0 \quad \text{for all} \quad x \in C \qquad (19.6)$$

19.1. Subgradient

19.1.2 Optimality Conditions via Subgradients

Or we have the strict feasibility

x^* is primal optimal (λ^* is dual optimal) iff

1. $f_i(x^*) \le 0, \quad i = 1, \cdots, m$
2. $\lambda_i^* \ge 0, \quad i = 1, \cdots, m$
3. $0 \in \partial f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \partial f_i(x^*)$
4. $\lambda_i^* f_i(x^*) = 0, \; i = 1, \cdots, m$

which generalizes KKT for nondifferentiable f_i

19.2. Subgradient Method

subgradient method is simple algorithm to minimize nondifferentiable convex function f

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

- $x^{(k)}$ is the kth iterate
- $g^{(k)}$ is **any** subgradient of f at $x^{(k)}$
- $\alpha_k > 0$ is the kth step size

not a descent method, so we keep track of best point so far

$$f_{\text{best}}^{(k)} = \min_{i=1,\ldots,k} f(x^{(i)})$$

19.2. Subgradient Method

step sizes are fixed ahead of time

- *constant step size:* $\alpha_k = \alpha$ (constant)
- *constant step length:* $\alpha_k = \gamma / \|g^{(k)}\|_2$ (so $\|x^{(k+1)} - x^{(k)}\|_2 = \gamma$)
- *square summable but not summable:* step sizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

- *nonsummable diminishing:* step sizes satisfy

$$\lim_{k \to \infty} \alpha_k = 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

19.2. Subgradient Method

Some assumptions are stronger than needed, just to simplify proofs

- $f^\star = \inf_x f(x) > -\infty$, with $f(x^\star) = f^\star$
- $\|g\|_2 \leq G$ for all $g \in \partial f$ (equivalent to Lipschitz condition on f)
- $\|x^{(1)} - x^\star\|_2 \leq R$

19.2. Subgradient Method

Some convergence results

define $\bar{f} = \lim_{k\to\infty} f_{\text{best}}^{(k)}$

- *constant step size:* $\bar{f} - f^\star \leq G^2\alpha/2$, *i.e.*,
  converges to $G^2\alpha/2$-suboptimal
  (converges to $f^\star$ if f differentiable, α small enough)

- *constant step length:* $\bar{f} - f^\star \leq G\gamma/2$, *i.e.*,
  converges to $G\gamma/2$-suboptimal

- *diminishing step size rule:* $\bar{f} = f^\star$, *i.e.*, **converges**

19.2. Subgradient Method

Convergence proof

key quantity: *Euclidean distance to the optimal set*, not the function value

let $x^\star$ be any minimizer of f

$$\begin{aligned}
\|x^{(k+1)} - x^\star\|_2^2 &= \|x^{(k)} - \alpha_k g^{(k)} - x^\star\|_2^2 \\
&= \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k g^{(k)T}(x^{(k)} - x^\star) + \alpha_k^2 \|g^{(k)}\|_2^2 \\
&\leq \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k (f(x^{(k)}) - f^\star) + \alpha_k^2 \|g^{(k)}\|_2^2
\end{aligned}$$

using $f^\star = f(x^\star) \geq f(x^{(k)}) + g^{(k)T}(x^\star - x^{(k)})$

19.2. Subgradient Method

apply recursively to get

$$\begin{aligned}\|x^{(k+1)} - x^\star\|_2^2 &\leq \|x^{(1)} - x^\star\|_2^2 - 2\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) + \sum_{i=1}^{k} \alpha_i^2 \|g^{(i)}\|_2^2 \\ &\leq R^2 - 2\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) + G^2 \sum_{i=1}^{k} \alpha_i^2\end{aligned}$$

now we use

$$\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) \geq (f_{\text{best}}^{(k)} - f^\star) \left(\sum_{i=1}^{k} \alpha_i \right)$$

to get

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i}.$$

19.2. Subgradient Method

constant step size: for $\alpha_k = \alpha$ we get

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 k \alpha^2}{2k\alpha}$$

righthand side converges to $G^2\alpha/2$ as $k \to \infty$

constant step length: for $\alpha_k = \gamma / \|g^{(k)}\|_2$ we get

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + \sum_{i=1}^k \alpha_i^2 \|g^{(i)}\|_2^2}{2\sum_{i=1}^k \alpha_i} \leq \frac{R^2 + \gamma^2 k}{2\gamma k/G},$$

righthand side converges to $G\gamma/2$ as $k \to \infty$

19.2. Subgradient Method

square summable but not summable step sizes:
suppose step sizes satisfy

$$\sum_{i=1}^{\infty} \alpha_k^2 < \infty, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

then

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i}$$

as $k \to \infty$, numerator converges to a finite number, denominator converges to ∞, so $f_{\text{best}}^{(k)} \to f^\star$

19.2. Subgradient Method

Stop criterion

- terminating when $\dfrac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i} \leq \epsilon$ is really, really, slow
- optimal choice of α_i to achieve $\dfrac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i} \leq \epsilon$ for smallest k:

$$\alpha_i = (R/G)/\sqrt{k}, \quad i = 1, \ldots, k$$

number of steps required: $k = (RG/\epsilon)^2$

- the truth: there really isn't a good stopping criterion for the subgradient method . . .

19.2. Subgradient Method

Example: Piecewise linear minimization

$$\text{minimize} \quad f(x) = \max_{i=1,\ldots,m}(a_i^T x + b_i)$$

to find a subgradient of f: find index j for which

$$a_j^T x + b_j = \max_{i=1,\ldots,m}(a_i^T x + b_i)$$

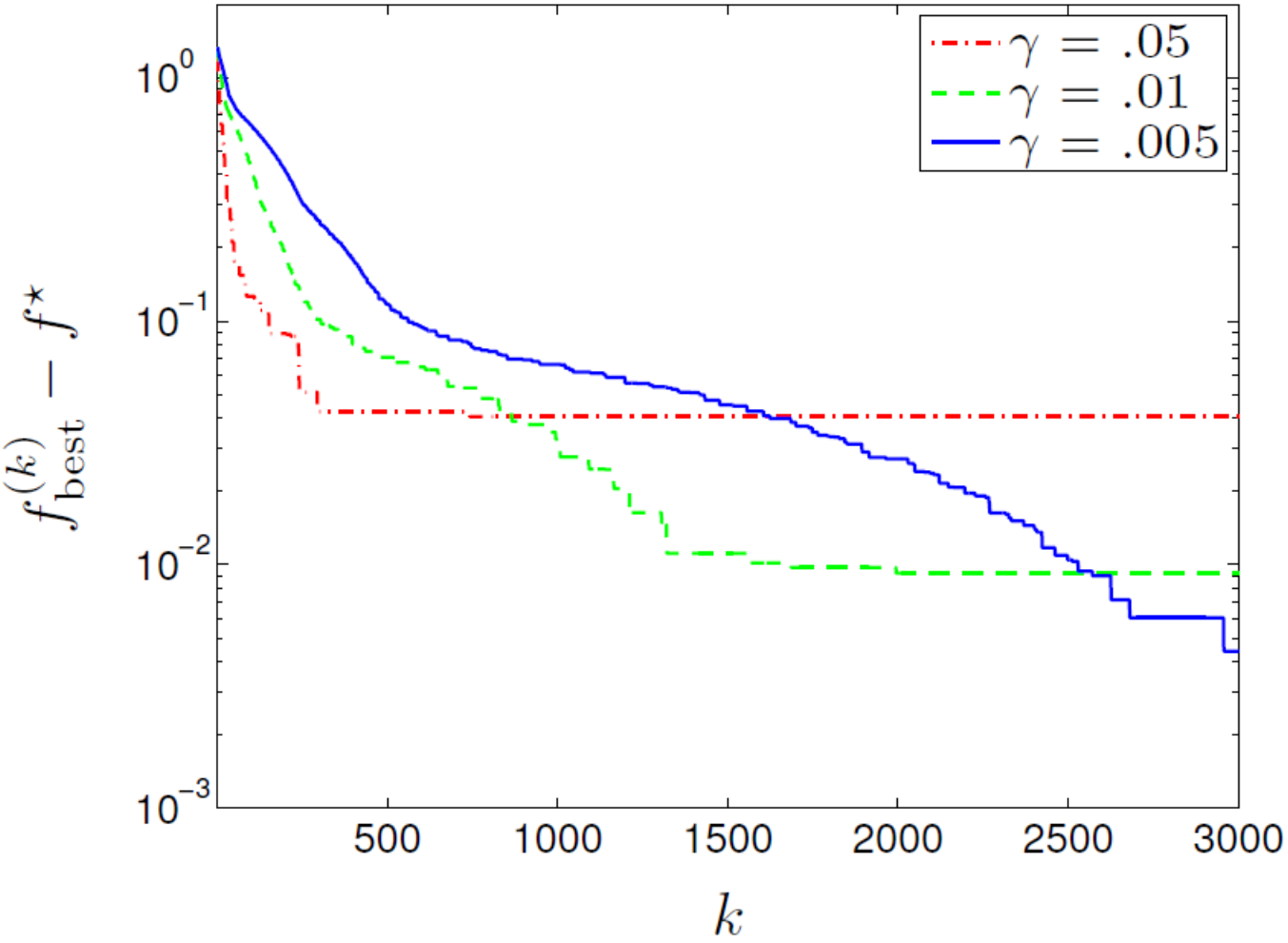
and take $g = a_j$

subgradient method: $x^{(k+1)} = x^{(k)} - \alpha_k a_j$

19.2. Subgradient Method

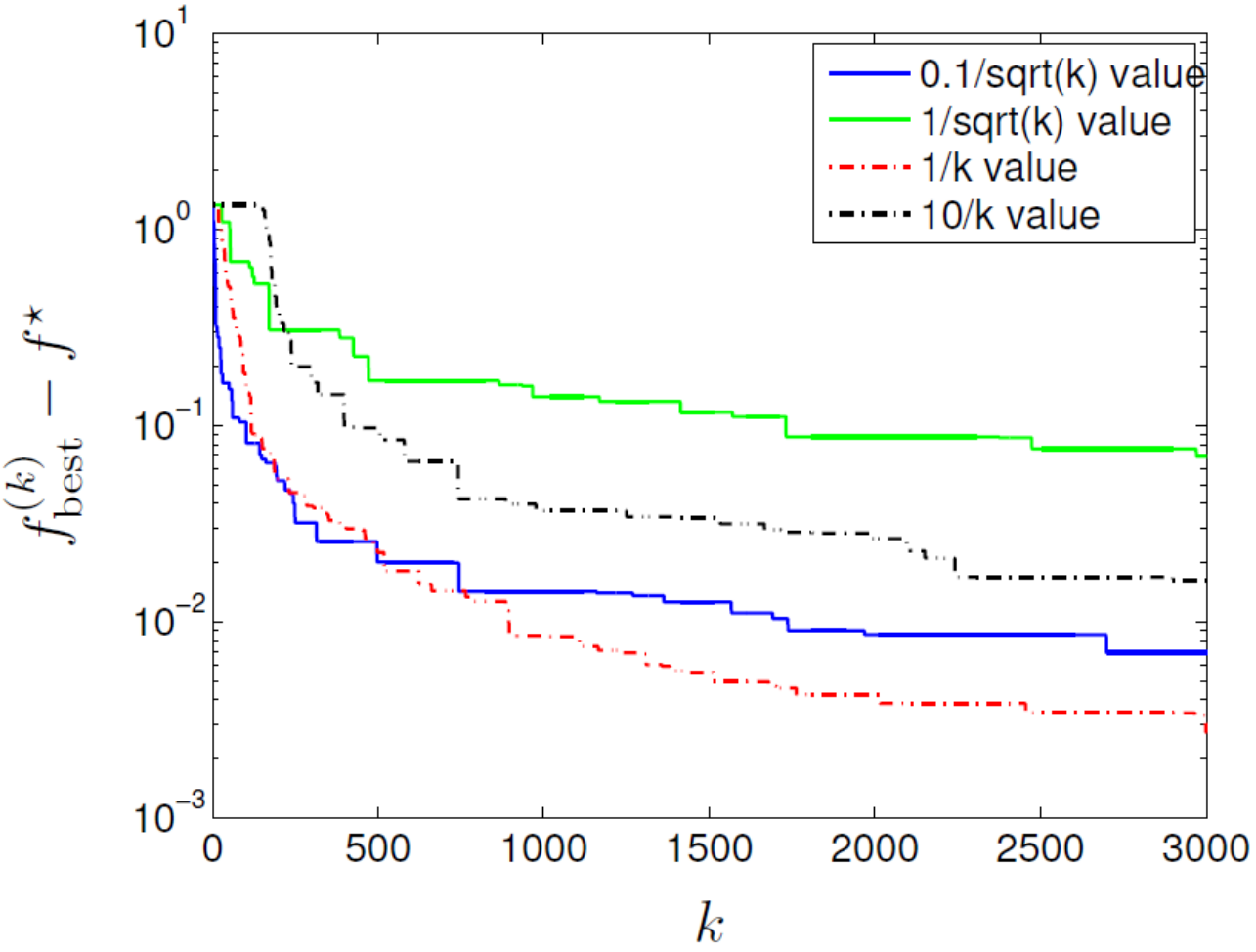
problem instance with $n = 20$ variables, $m = 100$ terms, $f^\star \approx 1.1$

$f_{\text{best}}^{(k)} - f^\star$, constant step length $\gamma = 0.05, 0.01, 0.005$



19.2. Subgradient Method

diminishing step rules $\alpha_k = 0.1/\sqrt{k}$ and $\alpha_k = 1/\sqrt{k}$, square summable step size rules $\alpha_k = 1/k$ and $\alpha_k = 10/k$



19.2. Subgradient Method

Optimal step size when f^* is known

- choice due to Polyak:

$$\alpha_k = \frac{f(x^{(k)}) - f^\star}{\|g^{(k)}\|_2^2}$$

(can also use when optimal value is estimated)

- motivation: start with basic inequality

$$\|x^{(k+1)} - x^\star\|_2^2 \leq \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k(f(x^{(k)}) - f^\star) + \alpha_k^2\|g^{(k)}\|_2^2$$

and choose α_k to minimize righthand side

19.2. Subgradient Method

- yields

$$\|x^{(k+1)} - x^\star\|_2^2 \leq \|x^{(k)} - x^\star\|_2^2 - \frac{(f(x^{(k)}) - f^\star)^2}{\|g^{(k)}\|_2^2}$$

  (in particular, $\|x^{(k)} - x^\star\|_2$ decreases each step)

- applying recursively,

$$\sum_{i=1}^{k} \frac{(f(x^{(i)}) - f^\star)^2}{\|g^{(i)}\|_2^2} \leq R^2$$

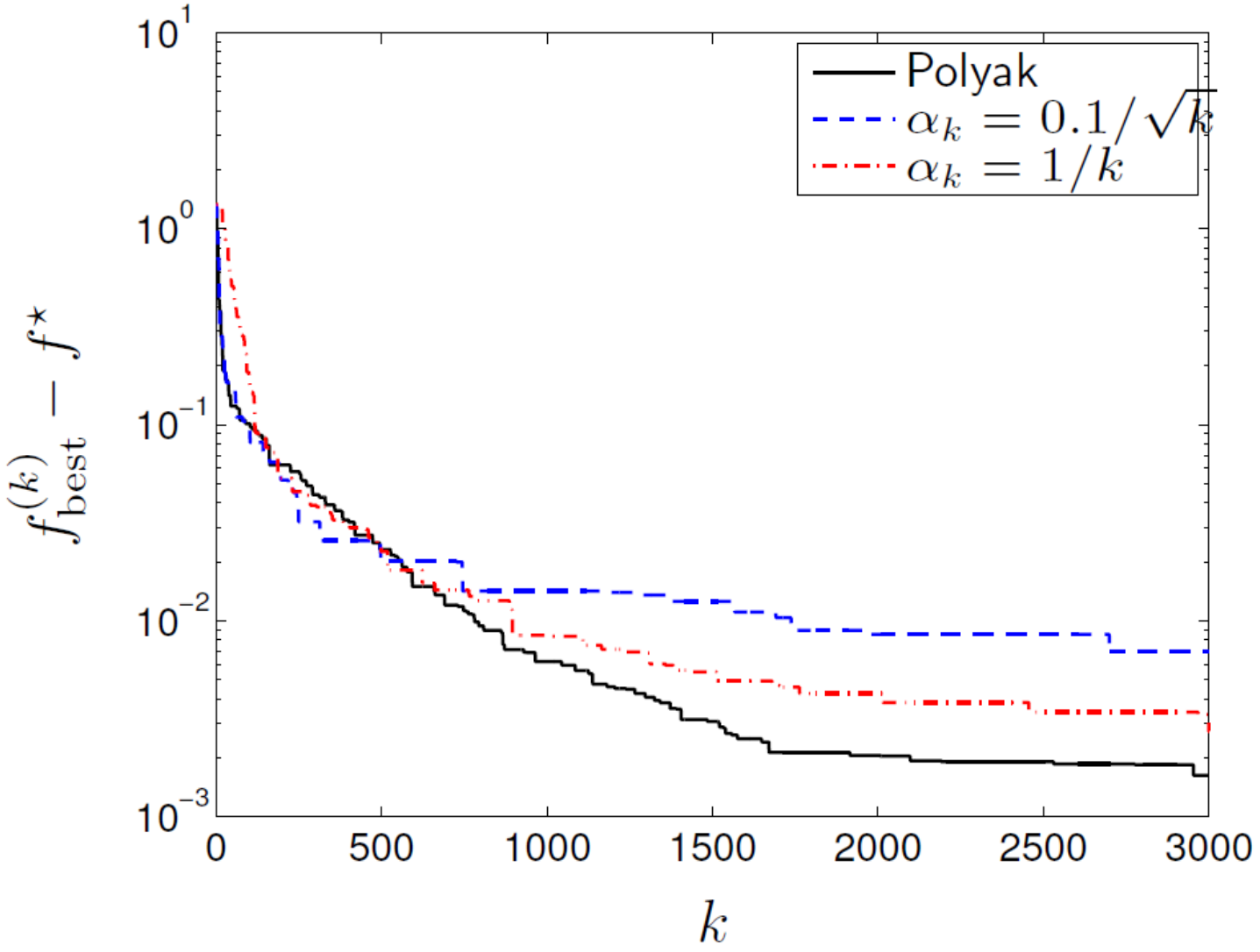
  and so

$$\sum_{i=1}^{k} (f(x^{(i)}) - f^\star)^2 \leq R^2 G^2$$

  which proves $f(x^{(k)}) \to f^\star$

19.2. Subgradient Method

PWL example with Polyak's step size, $\alpha_k = 0.1/\sqrt{k}$, $\alpha_k = 1/k$



19.2. Subgradient Method

Finding a point in the intersection of convex sets

$C = C_1 \cap \cdots C_m$ is nonempty, $C_1, \ldots, C_m \subseteq \mathbf{R}^n$ closed and convex

find a point in C by minimizing

$$f(x) = \max\{\mathbf{dist}(x, C_1), \ldots, \mathbf{dist}(x, C_m)\}$$

with $\mathbf{dist}(x, C_j) = f(x)$, a subgradient of f is

$$g = \nabla\, \mathbf{dist}(x, C_j) = \frac{x - P_{C_j}(x)}{\|x - P_{C_j}(x)\|_2}$$

19.2. Subgradient Method

subgradient update with optimal step size:

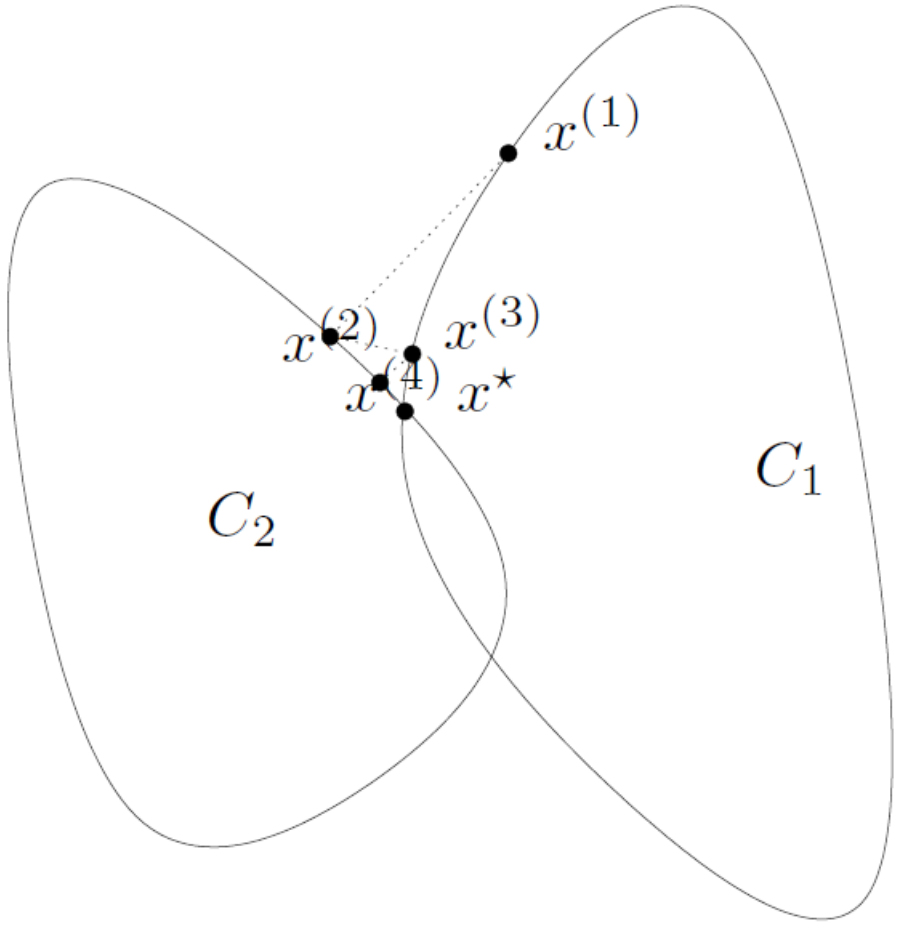
$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \alpha_k g^{(k)} \\ &= x^{(k)} - f(x^{(k)}) \frac{x - P_{C_j}(x)}{\|x - P_{C_j}(x)\|_2} \\ &= P_{C_j}(x^{(k)}) \end{aligned}$$

- a version of the famous *alternating projections* algorithm
- at each step, project the current point onto the farthest set
- for $m = 2$ sets, projections alternate onto one set, then the other
- convergence: $\mathbf{dist}(x^{(k)}, C) \to 0$ as $k \to \infty$

19.2. Subgradient Method

Alternating projections

first few iterations:



. . . $x^{(k)}$ eventually converges to a point $x^\star \in C_1 \cap C_2$

19.3. Subgradient for Constrained Problems

19.3.1 Projected Subgradient for Primary

solves constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C}, \end{array}$$

where $f : \mathbf{R}^n \to \mathbf{R}$, $\mathcal{C} \subseteq \mathbf{R}^n$ are convex

projected subgradient method is given by

$$x^{(k+1)} = \Pi(x^{(k)} - \alpha_k g^{(k)}),$$

Π is (Euclidean) projection on $\mathcal{C}$, and $g^{(k)} \in \partial f(x^{(k)})$

19.3. Subgradient for Constrained Problems

19.3.1 Projected Subgradient for Primary

same convergence results:

- for constant step size, converges to neighborhood of optimal (for f differentiable and h small enough, converges)

- for diminishing nonsummable step sizes, converges

key idea: projection does not increase distance to $x^\star$

19.3. Subgradient for Constrained Problems

19.3.1 Projected Subgradient for Primary

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

projection of z onto $\{x \mid Ax = b\}$ is

$$\begin{aligned} \Pi(z) &= z - A^T(AA^T)^{-1}(Az - b) \\ &= (I - A^T(AA^T)^{-1}A)z + A^T(AA^T)^{-1}b \end{aligned}$$

projected subgradient update is (using $Ax^{(k)} = b$)

$$\begin{aligned} x^{(k+1)} &= \Pi(x^{(k)} - \alpha_k g^{(k)}) \\ &= x^{(k)} - \alpha_k (I - A^T(AA^T)^{-1}A)g^{(k)} \\ &= x^{(k)} - \alpha_k \Pi_{\mathcal{N}(A)}(g^{(k)}) \end{aligned}$$

19.3. Subgradient for Constrained Problems

19.3.1 Projected Subgradient for Primary

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array}$$

subgradient of objective is $g = \mathbf{sign}(x)$

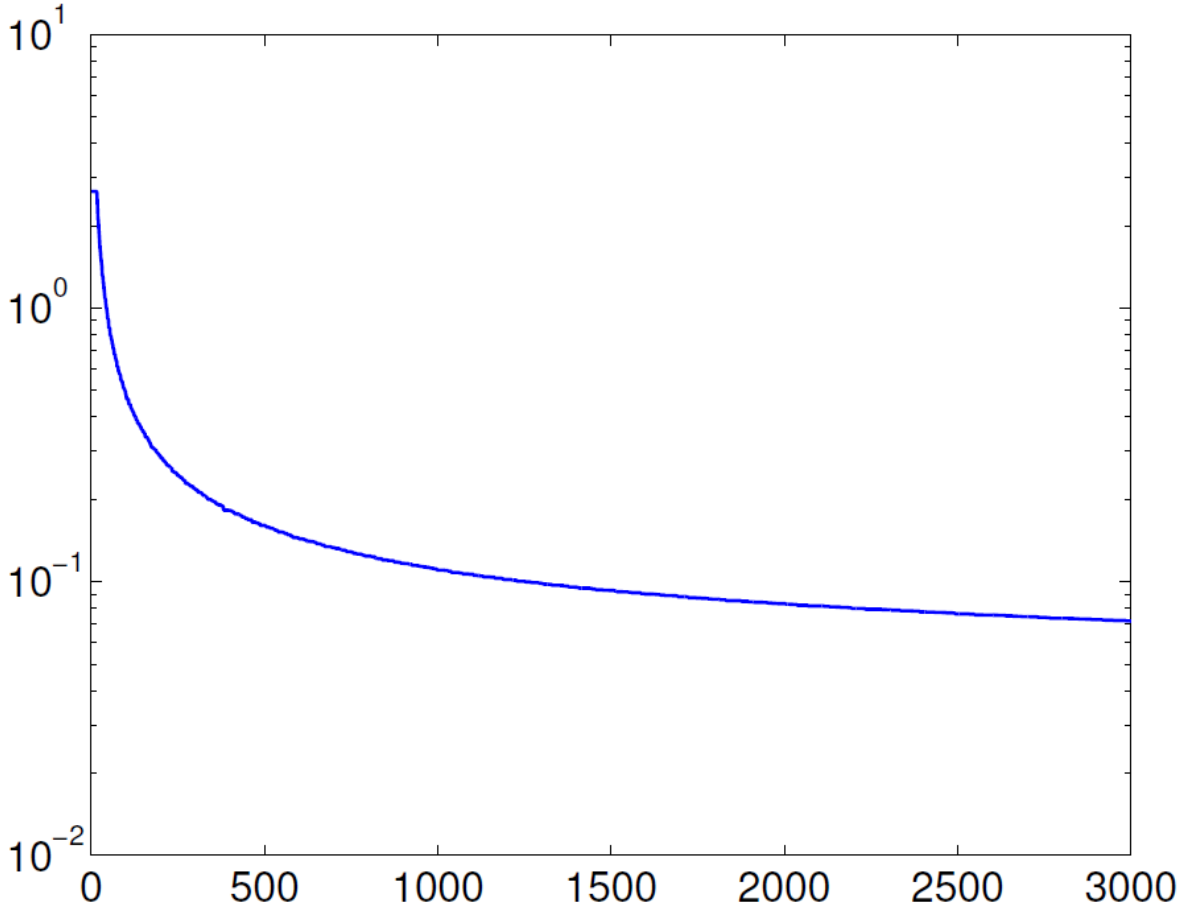
projected subgradient update is

$$x^{(k+1)} = x^{(k)} - \alpha_k (I - A^T (AA^T)^{-1} A)\, \mathbf{sign}(x^{(k)})$$

19.3. Subgradient for Constrained Problems

19.3.1 Projected Subgradient for Primary

problem instance with $n = 1000$, $m = 50$, step size $\alpha_k = 0.1/k$, $f^\star \approx 3.2$



19.3. Subgradient for Constrained Problems

19.3.2 Projected Subgradient for Dual

(convex) primal:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

solve dual problem

$$\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

via projected subgradient method:

$$\lambda^{(k+1)} = \left(\lambda^{(k)} - \alpha_k h\right)_+, \qquad h \in \partial(-g)(\lambda^{(k)})$$

19.3. Subgradient for Constrained Problems

19.3.2 Projected Subgradient for Dual

assume f_0 is strictly convex, and denote, for $\lambda \succeq 0$,

$$x^*(\lambda) = \operatorname*{argmin}_z \left(f_0(z) + \lambda_1 f_1(z) + \cdots + \lambda_m f_m(z)\right)$$

so $g(\lambda) = f_0(x^*(\lambda)) + \lambda_1 f_1(x^*(\lambda)) + \cdots + \lambda_m f_m(x^*(\lambda))$

a subgradient of $-g$ at λ is given by $h_i = -f_i(x^*(\lambda))$

projected subgradient method for dual:

$$x^{(k)} = x^*(\lambda^{(k)}), \qquad \lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)})\right)_+$$

19.3. Subgradient for Constrained Problems

19.3.2 Projected Subgradient for Dual

- primal iterates $x^{(k)}$ are not feasible, but become feasible in limit (sometimes can find feasible, suboptimal $\tilde{x}^{(k)}$ from $x^{(k)}$)
- dual function values $g(\lambda^{(k)})$ converge to $f^\star = f_0(x^\star)$

interpretation:

- λ_i is price for 'resource' $f_i(x)$
- price update $\lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)})\right)_+$
  - increase price λ_i if resource i is over-utilized ($i.e.$, $f_i(x) > 0$)
  - decrease price λ_i if resource i is under-utilized ($i.e.$, $f_i(x) < 0$)
  - but never let prices get negative

19.3. Subgradient for Constrained Problems

19.3.2 Projected Subgradient for Dual

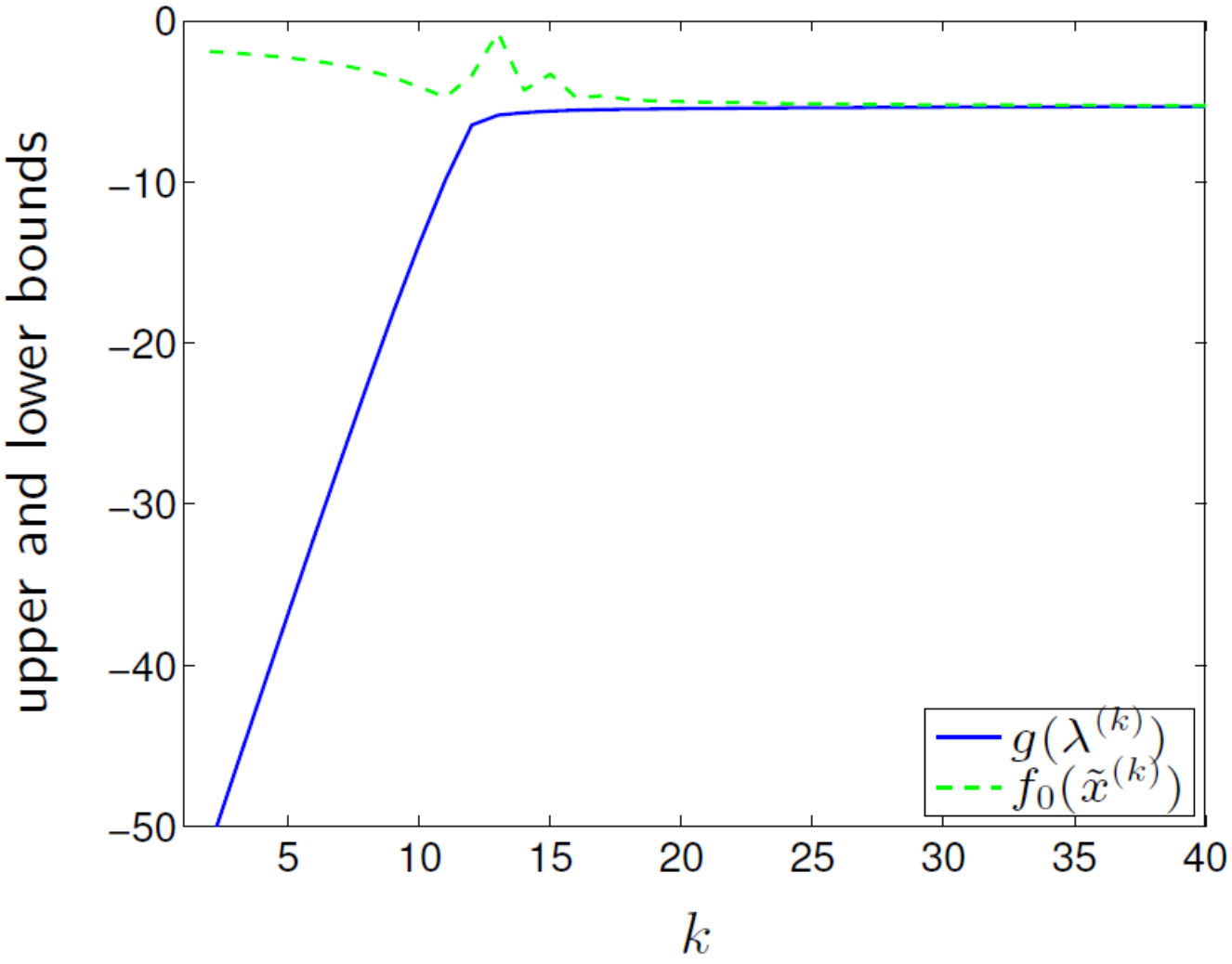
minimize strictly convex quadratic $(P \succ 0)$ over unit box:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x - q^T x \\ \text{subject to} & x_i^2 \leq 1, \quad i = 1, \ldots, n \end{array}$$

- $L(x, \lambda) = (1/2)x^T(P + \mathbf{diag}(2\lambda))x - q^T x - \mathbf{1}^T\lambda$
- $x^*(\lambda) = (P + \mathbf{diag}(2\lambda))^{-1}q$
- projected subgradient for dual:

$$x^{(k)} = (P + \mathbf{diag}(2\lambda^{(k)}))^{-1}q, \quad \lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k((x_i^{(k)})^2 - 1)\right)_+$$

problem instance with $n = 50$, fixed step size $\alpha = 0.1$, $f^\star \approx -5.3$; $\tilde{x}^{(k)}$ is a nearby feasible point for $x^{(k)}$



19.3. Subgradient for Constrained Problems

19.3.3 Subgradient Method for Constrained Optimization

solves constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where $f_i : \mathbf{R}^n \to \mathbf{R}$ are convex

same update $x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$, but we have

$$g^{(k)} \in \begin{cases} \partial f_0(x) & f_i(x) \leq 0, \quad i = 1, \ldots, m, \\ \partial f_j(x) & f_j(x) > 0 \end{cases}$$

define $f_{\text{best}}^{(k)} = \min\{f_0(x^{(i)}) \mid x^{(i)} \text{ feasible}, \ i = 1, \ldots, k\}$

19.3. Subgradient for Constrained Problems

19.3.3 Subgradient Method for Constrained Optimization

assumptions:

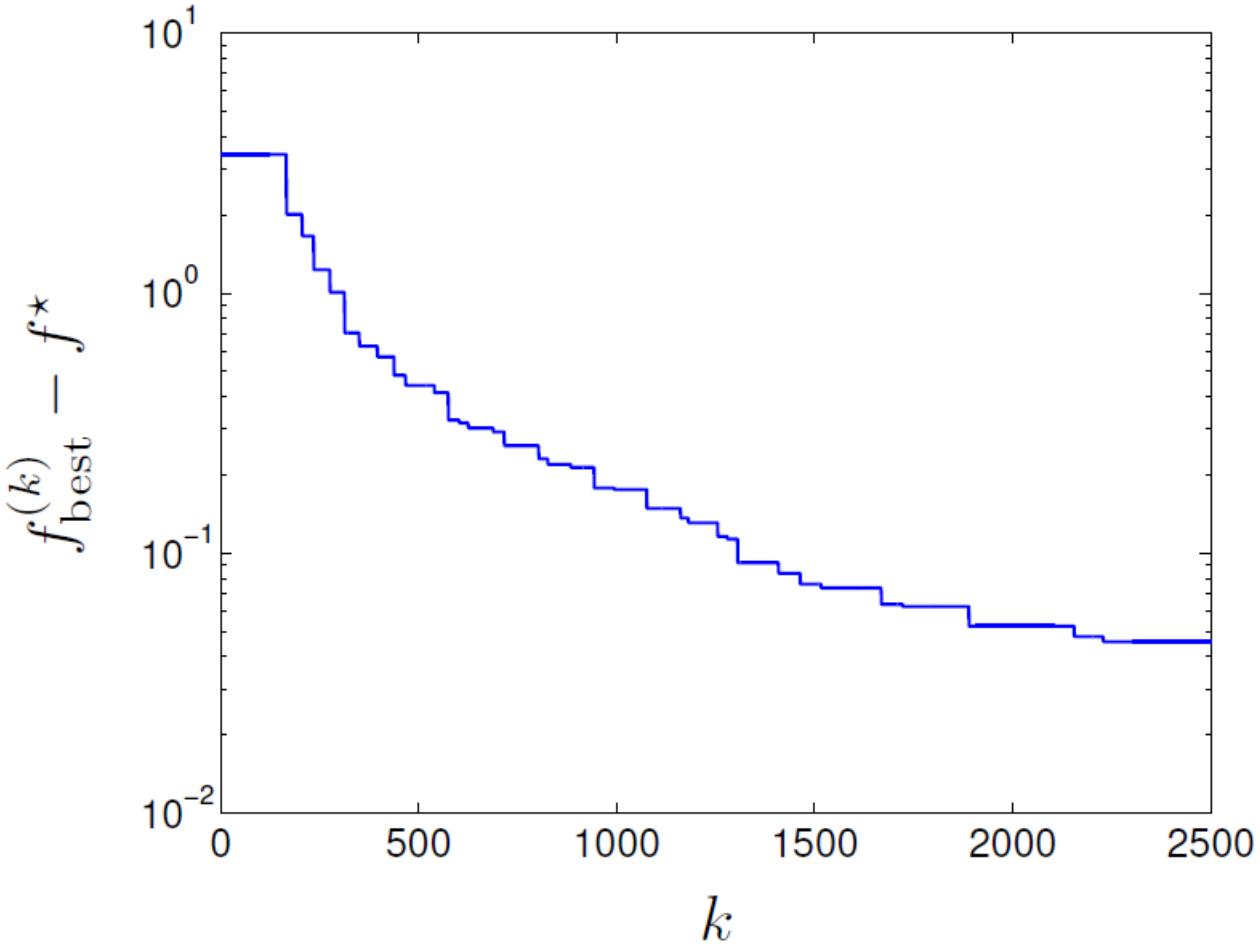
- there exists an optimal $x^\star$; Slater's condition holds
- $\|g^{(k)}\|_2 \leq G$; $\|x^{(1)} - x^\star\|_2 \leq R$

typical result: for $\alpha_k > 0$, $\alpha_k \to 0$, $\sum_{i=1}^{\infty} \alpha_i = \infty$, we have $f_{\text{best}}^{(k)} \to f^\star$

19.3. Subgradient for Constrained Problems

19.3.3 Subgradient Method for Constrained Optimization

LP with $n = 20$ variables, $m = 200$ inequalities, $f^\star \approx -3.4$;
$\alpha_k = 1/k$ for optimality step, Polyak's step size for feasibility step



19.4. Primal-Dual Subgradient Methods

- convex equality constrained problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

with variable x and optimal value $p^\star$

- we will work instead with (equivalent) *augmented problem*

$$\begin{array}{ll} \text{minimize} & f(x) + (\rho/2)\|Ax - b\|_2^2 \\ \text{subject to} & Ax = b \end{array}$$

where $\rho > 0$

19.4. Primal-Dual Subgradient Methods

- **augmented Lagrangian** is

$$L(x,\nu) = f(x) + \nu^T(Ax-b) + (\rho/2)\|Ax-b\|_2^2$$

- (x,ν) primal-dual optimal if and only if

$$\begin{aligned} 0 &\in \partial_x L(x,\nu) = \partial f(x) + A^T\nu + \rho A^T(Ax-b) \\ 0 &= -\nabla_\nu L(x,\nu) = b - Ax \end{aligned}$$

- same as $0 \in T(x,\nu)$, with $z=(x,\nu)$ and $T(x,\nu) = \begin{bmatrix} \partial_x L(x,\nu) \\ -\nabla_\nu L(x,\nu) \end{bmatrix}$

- T is a **monotone operator** (much more on this later)

19.4. Primal-Dual Subgradient Methods

- primal-dual subgradient method is

$$z^{(k+1)} = z^{(k)} - \alpha_k T^{(k)}$$

  where $T^{(k)} \in T(z^{(k)})$ and α_k is step length

- more explicitly:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \alpha_k (g^{(k)} + A^T \nu^{(k)} + \rho A^T (Ax^{(k)} - b)) \\ \nu^{(k+1)} &= \nu^{(k)} + \alpha_k (Ax^{(k)} - b) \end{aligned}$$

  where $g^{(k)} \in \partial f(x^{(k)})$

19.4. Primal-Dual Subgradient Methods

Convergence results

with step size $\alpha_k = \gamma_k / \|T^{(k)}\|_2$,

$$\gamma_k > 0, \quad \sum_k \gamma_k = \infty, \quad \sum_k \gamma_k^2 < \infty$$

we get convergence:

$$f(x^{(k)}) \to p^\star, \qquad Ax^{(k)} - b \to 0$$

19.4. Primal-Dual Subgradient Methods

- convex inequality constrained problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \qquad i = 1, \ldots, m \end{array}$$

with variable x, optimal value $p^\star$

- (equivalent) augmented problem

$$\begin{array}{ll} \text{minimize} & f_0(x) + (\rho/2)\|F(x)\|_2^2 \\ \text{subject to} & F(x) \preceq 0 \end{array}$$

where $F(x) = (f_1(x)_+, \ldots, f_m(x)_+)$, $\rho > 0$

19.4. Primal-Dual Subgradient Methods

- augmented Lagrangian is

$$L(x,\lambda) = f_0(x) + \lambda^T F(x) + (\rho/2)\|F(x)\|_2^2$$

- (x,λ) primal-dual optimal if and only if

$$\begin{aligned} 0 &\in \partial_x L(x,\lambda) = \partial f_0(x) + \sum_{i=1}^{m} (\lambda_i + \rho f_i(x)_+)\partial f_i(x)_+ \\ 0 &= -\nabla_\lambda L(x,\lambda) = -F(x) \end{aligned}$$

19.4. Primal-Dual Subgradient Methods

- define $z = (x, \nu)$ and

$$T(x, \lambda) = \begin{bmatrix} \partial_x L(x, \lambda) \\ -\nabla_\lambda L(x, \lambda) \end{bmatrix}$$

(T is the KKT operator for the problem, and is monotone)

- primal-dual subgradient method is

$$z^{(k+1)} = z^{(k)} - \alpha_k T^{(k)}$$

where $T^{(k)} \in T(z^{(k)})$ and α_k is step length

19.4. Primal-Dual Subgradient Methods

- more explicitly:

$$
\begin{aligned}
x^{(k+1)} &= x^{(k)} - \alpha_k \left(g_0^{(k)} + \sum_{i=1}^{m} (\lambda_i^{(k)} + \rho f_i(x^{(k)})_+) g_i^{(k)} \right) \\
\lambda_i^{(k+1)} &= \lambda_i^{(k)} + \alpha_k f_i(x^{(k)})_+, \quad i = 1, \ldots, m
\end{aligned}
$$

  where $g_0^{(k)} \in \partial f_0(x^{(k)})$, $g_i^{(k)} \in \partial f_i(x^{(k)})_+$, $i = 1, \ldots, m$

- note that $\lambda_i^{(k)}$ can only increase with k

19.4. Primal-Dual Subgradient Methods

Convergence results

with step size $\alpha_k = \gamma_k / \|T^{(k)}\|_2$,

$$\gamma_k > 0, \quad \sum_k \gamma_k = \infty, \quad \sum_k \gamma_k^2 < \infty$$

we get convergence:

$$f_0(x^{(k)}) \to p^\star, \qquad f_i(x^{(k)})_+ \to 0, \quad i = 1, \ldots, m$$

19.4. Primal-Dual Subgradient Methods

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \end{array}$$

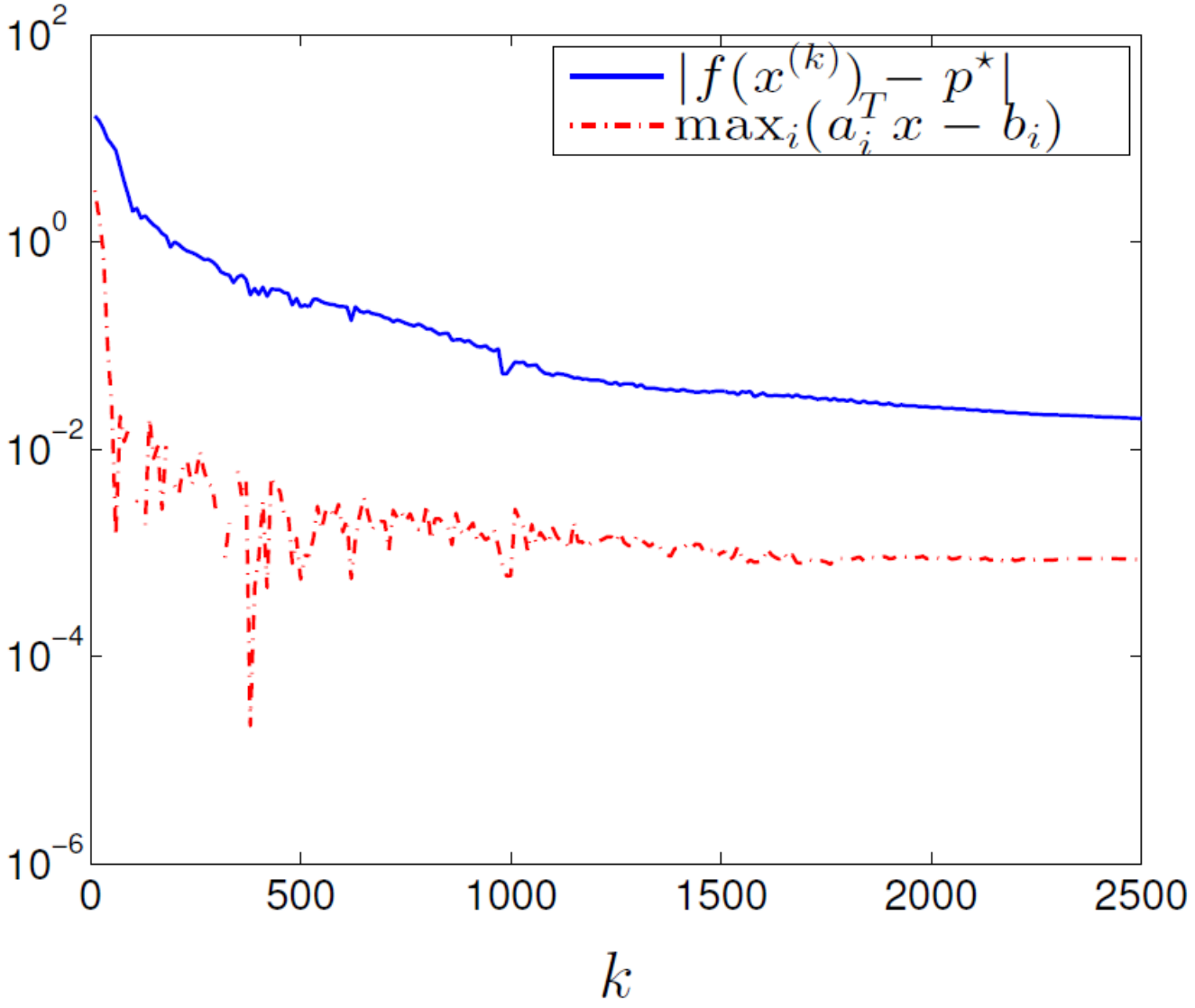
primal-dual subgradient update is

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \alpha_k \left(c + A^T M^{(k)} (\lambda^{(k)} + \rho (Ax^{(k)} - b)_+) \right) \\ \lambda^{(k+1)} &= \lambda^{(k)} + \alpha_k (Ax^{(k)} - b)_+ \end{aligned}$$

where $M^{(k)}$ is a diagonal matrix

$$M_{ii}^{(k)} = \begin{cases} 1 & a_i^T x^{(k)} > b_i \\ 0 & a_i^T x^{(k)} \leq b_i \end{cases}$$

problem instance with $n = 20$, $m = 200$, $p^\star \approx -3.4$
step size $\alpha_k = 1/(k\|T^{(k)}\|_2)$



19.5. ADMM

复习：对偶问题

- convex equality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$
- dual function: $g(y) = \inf_x L(x, y)$
- dual problem: maximize $g(y)$
- recover $x^\star = \operatorname{argmin}_x L(x, y^\star)$

19.5. ADMM

复习：对偶上升算法 dual ascent

- gradient method for dual problem: $y^{k+1} = y^k + \alpha^k \nabla g(y^k)$
- $\nabla g(y^k) = A\tilde{x} - b$, where $\tilde{x} = \operatorname{argmin}_x L(x, y^k)$
- dual ascent method is

$$\begin{array}{lll} x^{k+1} & := & \operatorname{argmin}_x L(x, y^k) \qquad // \ x\text{-minimization} \\ y^{k+1} & := & y^k + \alpha^k (Ax^{k+1} - b) \qquad // \ \text{dual update} \end{array}$$

- works, with lots of strong assumptions

19.5. ADMM

复习：对偶分解 dual decomposition

- suppose f is separable:

$$f(x) = f_1(x_1) + \cdots + f_N(x_N), \quad x = (x_1, \ldots, x_N)$$

- then L is separable in x: $L(x, y) = L_1(x_1, y) + \cdots + L_N(x_N, y) - y^T b$,

$$L_i(x_i, y) = f_i(x_i) + y^T A_i x_i$$

- x-minimization in dual ascent splits into N separate minimizations

$$x_i^{k+1} := \operatorname*{argmin}_{x_i} L_i(x_i, y^k)$$

which can be carried out in parallel

19.5. ADMM

复习：对偶分解 dual decomposition

- dual decomposition (Everett, Dantzig, Wolfe, Benders 1960–65)

$$x_i^{k+1} := \operatorname{argmin}_{x_i} L_i(x_i, y^k), \quad i = 1, \ldots, N$$
$$y^{k+1} := y^k + \alpha^k (\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b)$$

- scatter y^k; update x_i in parallel; gather $A_i x_i^{k+1}$
- solve a large problem
  - by iteratively solving subproblems (in parallel)
  - dual variable update provides coordination
- works, with lots of assumptions; often slow

19.5. ADMM

复习：Method of multipliers

- a method to robustify dual ascent
- use **augmented Lagrangian** (Hestenes, Powell 1969), $\rho > 0$

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2$$

- method of multipliers (Hestenes, Powell; analysis in Bertsekas 1982)

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} - b) \end{aligned}$$

(note specific dual update step length ρ)

19.5. ADMM

复习：Method of multipliers

- optimality conditions (for differentiable f):

$$Ax^\star - b = 0, \qquad \nabla f(x^\star) + A^T y^\star = 0$$

  (primal and dual feasibility)

- since x^{k+1} minimizes $L_\rho(x, y^k)$

$$\begin{aligned} 0 &= \nabla_x L_\rho(x^{k+1}, y^k) \\ &= \nabla_x f(x^{k+1}) + A^T \left(y^k + \rho(Ax^{k+1} - b)\right) \\ &= \nabla_x f(x^{k+1}) + A^T y^{k+1} \end{aligned}$$

- dual update $y^{k+1} = y^k + \rho(x^{k+1} - b)$ makes (x^{k+1}, y^{k+1}) *dual feasible*
- *primal feasibility* achieved in limit: $Ax^{k+1} - b \to 0$

19.5. ADMM

复习：Method of multipliers

(compared to dual decomposition)

- *good news*: converges under much more relaxed conditions (f can be nondifferentiable, take on value $+\infty$, . . .)
- *bad news*: quadratic penalty destroys splitting of the x-update, so can't do decomposition

19.5. ADMM

Method of multipliers

(compared to dual decomposition)

- *good news*: converges under much more relaxed conditions (f can be nondifferentiable, take on value $+\infty$, ...)
- *bad news*: quadratic penalty destroys splitting of the x-update, so can't do decomposition

Alternating direction method of multipliers (ADMM)

- a method
  - with good robustness of method of multipliers
  - which can support decomposition
- "robust dual decomposition" or "decomposable method of multipliers"
- proposed by Gabay, Mercier, Glowinski, Marrocco in 1976

19.5. ADMM

- ADMM problem form (with f, g convex)

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

  - two sets of variables, with separable objective

- $L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$

- ADMM:

$$\begin{array}{lcll} x^{k+1} & := & \text{argmin}_x \, L_\rho(x, z^k, y^k) & \textit{// } x\textit{-minimization} \\ z^{k+1} & := & \text{argmin}_z \, L_\rho(x^{k+1}, z, y^k) & \textit{// } z\textit{-minimization} \\ y^{k+1} & := & y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) & \textit{// dual update} \end{array}$$

- if we minimized over x and z jointly, reduces to method of multipliers
- instead, we do one pass of a Gauss-Seidel method
- we get splitting since we minimize over x with z fixed, and vice versa

- optimality conditions (for differentiable case):
  - primal feasibility: $Ax + Bz - c = 0$
  - dual feasibility: $\nabla f(x) + A^T y = 0, \quad \nabla g(z) + B^T y = 0$
- since z^{k+1} minimizes $L_\rho(x^{k+1}, z, y^k)$ we have

$$\begin{aligned} 0 &= \nabla g(z^{k+1}) + B^T y^k + \rho B^T (Ax^{k+1} + Bz^{k+1} - c) \\ &= \nabla g(z^{k+1}) + B^T y^{k+1} \end{aligned}$$

- so with ADMM dual variable update, $(x^{k+1}, z^{k+1}, y^{k+1})$ satisfies second dual feasibility condition
- primal and first dual feasibility are achieved as $k \to \infty$

19.6. References

[1] Mert Pilanci, EE364b - Convex Optimization II, http://web.stanford.edu/class/ee364b/lectures.html

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, "Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1-122, 2011.

[3] https://zhuanlan.zhihu.com/p/86826985