Unconstrained optimization III
Less-than-gradient methods

Antonio Frangioni

Department of Computer Science
University of Pisa
frangio@di.unipi.it

Computational Mathematics for Learning and Data Analysis
Master in Computer Science – University of Pisa

Outline

Incremental (a.k.a. Stochastic) Gradient

- Motivation: Machine Learning (really?). $I = \{1, \ldots, m\}$ observations, $X = [X^i \subset \mathcal{X}]_{i \in I}$ inputs, $y = [y^i]_{i \in I}$ outputs, want to explain y from X
- Mapping $\Phi : \mathcal{X} \to \mathcal{F}$ from input space to feature space
- Dependence linear in feature space $\Longrightarrow$ fitting

  $$\min \left\{ \sum_{i \in I} l(y^i, \langle \Phi(X^i), w \rangle) : w \in \mathbb{R}^n \right\}$$

  with $l(\cdot, \cdot) =$ loss function (could be l^i)
- Easy case: $\mathcal{F} \subseteq \mathbb{R}^n \Longrightarrow \langle \cdot, \cdot \rangle$ our old friend, $\Phi(X) = A \in \mathbb{R}^{n \times m}$
- Typical loss functions: $l(y, z) = (y - z)^2/2$ for regression, some convex approximation of $l(y, z) = y - sign(z)$ for classification ($y \in \{0, 1\}$)
- Linear least squares: $\min\{ f(w) = \sum_{i \in I} f^i(w) = (y^i - A^i w)^2/2 \}$
- $\nabla f(w) = \sum_{i \in I} \nabla f^i(w) = \sum_{i \in I} -A^i(y^i - A^i w)$ (**check**)
  if $m \gg n$, sum of a very large number of terms
  (not the smart way but bear with me, $l()$ may not be so simple)
- Computing ∇f can be costly already, I'd save to same some

Incremental (a.k.a. Stochastic) Gradient (cont.d)

- Intuition: X^i are i.i.d., "many of them will cancel out" $\Longrightarrow$ a small sample may be enough to compute a close $\approx \nabla f$
- $K \subset I$ "small", $d^i = -\nabla f^K(w) = \sum_{i \in K} \nabla f^i(w)$
- This may not be a descent direction, a different analysis is needed (but heavy ball and ACCG are not descent methods, either)
- How to choose K? What $|K|$ should be?
- Apparently no better way than at random, and $|K| = 1$ often used
- Called incremental gradient, but K random $\Longrightarrow$ stochastic process $\Longrightarrow$ stochastic gradient
- "Normal" iteration using ∇f called "batch", $|K| > 1$ called "mini batch"
- Many variants: heavy ball, perform one batch iteration every m ...
- "Extreme" version: on-line. Observations keep coming (typically fast) and have to be discarded once used
- How can something like this ever converge? (even in a probabilistic sense)

It can easily get worse

- What a true ML-person would really solve is

$$\min \left\{ \sum_{i \in I} l(y^i, \langle \Phi(X^i), w \rangle) + \mu \Omega(w) \ : \ w \in \mathbb{R}^n \right\}$$

  where $\Omega(w)$ regularizer (good theoretical and practical reasons)

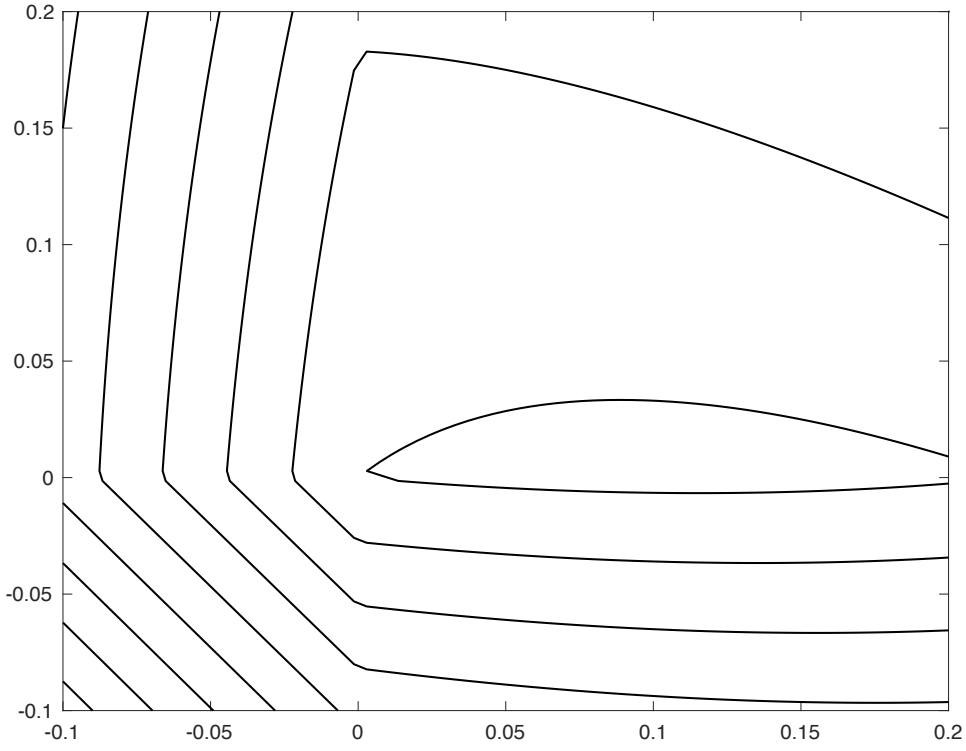
- μ hyper-parameter, to be determined empirically
- Best known regularizer: $\Omega(w) = \| w \|_1$ = Lasso
- Advantages over (say) $\Omega(w) = \| w \|_2^2$: increases sparsity
- Have as many $w_j = 0$ as possible $\approx$ feature selection
- $\| \cdot \|_1$ best convex approximation of $\| \cdot \|_0$ ("true" feature selection)
- But $f(w) = \| w \|_1$ is not differentiable

Exercise: characterize algebraically $\partial[\| y - A \cdot \|^2 + \| \cdot \|_1](w)$

- Never use a smooth method on a nonsmooth function,
  it just does not work

See with your own eyes

- $X^1 = [3, 2]$, $y^1 = 2$, $\mu = 10 \Longrightarrow$
  $f(w_1, w_2) =$
  $(3w_1 + 2w_2 - 2)^2 + 10(|w_1| + |w_2|)$
- Plenty of points where level sets are "kinky" (**check: which?**)
- ∇f does not exist there
- ∂f does, so could take a subgradient
- How chosen?



- There always exists a subgradient "pointing inside the level set" (e.g. min-norm one) $\equiv$ a descent direction (unless at minimum)
- But many others "point outside" $\equiv$ no descent direction
- A descent method choosing them would get $\alpha^i = 0$ and get stuck, something else is needed
- But f is convex, and this can be exploited

Outline

Subgradient methods: basics

- Recall $g \in \partial f(x) \equiv f(y) \geq f(x) + g(\, y - x\,) \;\; \forall\, y \in \mathbb{R}^n$
- We can assume to be able to compute any "random" one of them
- Yet, any $(-)$ subgradient points towards x_*

  $f(x) > f(x_*) \geq f(x) + g(\, x_* - x\,) \Longrightarrow (-g)(\, x_* - x\,) > 0$
- An appropriate step along a $-$ subgradient brings me nearer to x_*

  but what is an appropriate step?
- Anyway, the scheme $x^{i+1} = x^i + \alpha^i d^i$ for $d^i = -g^i / \| g^i \|$ makes sense
- Fundamental relationship (F): $\| x^{i+1} - x_* \|^2 = \| x^i - \alpha^i d^i - x_* \|^2$

  $= \| x^i - x_* \|^2 + 2\alpha^i g^i (\, x_* - x^i\,)/\| g^i \| + (\alpha^i)^2$

  $\leq \| x^i - x_* \|^2 - 2\alpha^i (\, f(x^i) - f(x_*)\,)/\| g^i \| + (\alpha^i)^2$
- $-2(\, f(x^i) - f(x_*)\,)/\| g^i \| < 0 \; + \; \alpha^i \searrow \Longrightarrow (\alpha^i)^2 \searrow 0_+$ fast

  $\Longrightarrow \| x^{i+1} - x_* \|^2 < \| x^i - x_* \|^2$ (knew that already)
- But what is the small enough, large enough value of α^i?

  Can't use the line search to find it
- If we knew $f(x^i) - f(x_*)$ we could estimate it, but we don't

Subgradient methods: stepsizes

- Can define a stepsize that will work no matter what:

  (DSS) $\quad \sum_{i=1}^{\infty} \alpha^i = \infty \ \wedge \ \sum_{i=1}^{\infty} (\alpha^i)^2 < \infty$

  "$\alpha^i \searrow 0$ but not fast enough that the series converges"

  called "diminishing-square summable" stepsize
- (DSS) $\wedge$ (F) $\Longrightarrow \| x^{i+1} - x_* \|^2 \leq \| x^1 - x_* \|^2 + \sum_{k=1}^{i} (\alpha^k)^2 \Longrightarrow$
  $\| x^i - x_* \| < \infty \ \forall i \equiv \{ x^i \}$ bounded $\Longrightarrow$ (a subsequence of) $\{ x^i \} \to \bar{x}$
- $C \subseteq$ *int dom* f bounded $\Longrightarrow \{ g^i \}$ bounded $\equiv \| g^i \| \leq L$ (**why?**)
- Hence $\liminf_{i \to \infty} f(x^i) = f_*$. In fact, $f(x^i) - f_* \geq \varepsilon > 0 \ \ \forall i \Longrightarrow$
  $\| x^{i+1} - x_* \|^2 \leq \| x^1 - x_* \|^2 - \delta \sum_{k=1}^{i} \alpha^k + \sum_{k=1}^{i} (\alpha^k)^2$
  (DSS) $\Longrightarrow$ as $i \to \infty$ the right-hand side $\to -\infty$ ↯
- All in all, (a subsequence of) $\{ x^i \} \to x_*$
- Incredibly robust result: α^i chosen with no knowledge of f at all;
  not even using function values, actually, only subgradients
- Easily extends to $g^i \in \partial_{\varepsilon^i} f(x^i)$, provided that $\{ \varepsilon^i \} \to 0$
- However practical convergence speed is abysmal, cannot use this

Subgradient methods: "better" stepsizes

- "We could estimate α^i if we knew $f(x_*)$, but we don't":

Subgradient methods: "better" stepsizes

- "We could estimate α^i if we knew $f(x_*)$, but we don't": what if we do?
- $-2\alpha^i(\, f(x^i) - f(x_*)\,)/\|\, g^i\,\| + (\alpha^i)^2 < 0 \equiv 2(\, f(x^i) - f(x_*)\,)/\|\, g^i\,\| > \alpha^i$
- Polyak stepsize: (PSS) $\alpha^i = \beta^i(\, f(x^i) - f(x_*)\,)/\|\, g^i\,\|,\ )$
  $\Longrightarrow \|\, x^{i+1} - x_*\,\|^2 < \|\, x^i - x_*\,\|^2 \Longrightarrow \{\, x^i\,\} \to x_*$ ($\Longrightarrow$ bounded)
- Vastly better in practice as far as it can go, which means what?
- $(\Delta^{i+1})^2 := (\, f(x^{i+1}) - f(x_*)\,)^2 \le L^2\|\, x^{i+1} - x_*\,\|^2$ (**why?**)
  $\le L^2\big[\,\|\, x^i - x_*\,\|^2 + \beta^i(\,\beta^i - 2\,)(\, f(x^i) - f(x_*)\,)^2/L^2\,\big]$ (**check**)
  $= L^2\|\, x^i - x_*\,\|^2 + \beta^i(\,\beta^i - 2\,)(\Delta^i)^2$
- $\text{argmax}\{\,\beta(\,\beta - 2\,)\,\} = 1,\ 1(1 - 2\cdot 1) = -1 \Longrightarrow \beta^i = 1$
- Even if $\Delta^i \ge u\|\, x^i - x_*\,\|^2 \Longrightarrow$ one gets $(\Delta^{i+1})^2 \le \Delta^i(\, 1 - \delta\Delta^i\,)$ (**check**)
  even weaker than $\Delta^{i+1} \le \Delta^i(\, 1 - \delta\Delta^i\,) \Longrightarrow O(\, 1\,/\,\varepsilon\,)$
- In fact, efficiency turns out to be $O(\, 1\,/\,\varepsilon^2\,)$ (ouch!!)
- $\varepsilon =$ 1e-3 $\longrightarrow \varepsilon =$ 1e-4 $\Longrightarrow$ 100 times more iterations
- Any accuracy $<$ 1e-4 typically unattainable in practice
- More bad news: $O(\, 1\,/\,\varepsilon^2\,)$ is actually optimal, no ("black-box") method for nondifferentiable f can do better

Target level stepsize

- Even more bad news: f_* is not known (again)
- "If you don't know it estimate it, but be ready to revise your estimate"
- (Vanishing) target level approach:

procedure $x = SGPTL\ (\ f\ ,\ g\ ,\ x\ ,\ i_{max}\ ,\ \beta\ ,\ \delta_0\ ,\ R\ ,\ \rho\ )\ \ \{$
  $r \leftarrow 0;\ \delta \leftarrow \delta_0;\ f_{ref} \leftarrow f_{rec} \leftarrow f(x);\ i \leftarrow 1;$
  while($i < i_{max}$) **do** $\{$
    $g = g(x);\ \alpha = \beta(\ f(x) - (\ f_{ref} - \delta\ )\ )/\|\ g\ \|;\ x \leftarrow x - \alpha g;$
    if($f(x) \leq f_{ref} - \delta/2$) **then** $\{\ f_{ref} \leftarrow f_{rec};\ r \leftarrow 0;\ \}$
      else if($r > R$) **then** $\{\ \delta \leftarrow \delta\rho;\ r \leftarrow 0;\ \}$
      else $r \leftarrow r + \alpha\|\ g\ \|;$
    $f_{rec} \leftarrow \min\{\ f_{rec}\ ,\ f(x)\ \};\ i \leftarrow i + 1;$
  $\}$
$\}$

- f_{ref} = reference value $\approx f_{rec}$ = best value found so far $+\delta$ = threshold defines target level $f_{ref} - \delta \approx f_*$ $(< f(x) \Longrightarrow \alpha > 0)$
- "Good improvement" $\Longrightarrow f_{ref} \searrow \Longrightarrow$ target level $\searrow$
- "Too many steps without improvement" $\Longrightarrow \delta \nearrow \Longrightarrow$ target level $\nearrow$
- (Too) many parameters: $\rho \in (0,1)$, $\beta \in (0,2)$, $\delta_0 > 0$ (??), $R > 0$ (???)

Target level stepsize (cont.d)

- At least, $\{ f^i_{rec} \} \to f_*$
- No reasonable stopping criterion ($\| g^i \| \not\to 0$), just "stop after a while"
- Simpler versions exist, e.g. nonvanishing threshold:
  $$\delta^i \geq \underline{\delta} > 0 \Longrightarrow \{ f^i_{rec} \} \to f_* + \underline{\delta}$$
- Still (too) many parameters
- More complicated versions exist, heuristics to better handle target level
  $\Longrightarrow$ better convergence in practice, but yet more parameters
- In some applications a (potentially good) $\underline{f} \leq f_*$ is known, and this helps
- More in general, one may actually want to reach a known target, not necessarily f_*, and this helps
- Anyway, convergence is slow
- Can it be made any better?

Deflected subgradient

- "Want a better direction? Use a better model!"
- But there is no second-order information (well, there might be, but . . .)
- Yet, deflection is possible: $d^i = \gamma^i g^i + (1 - \gamma^i) d^{i-1}$, $x^{i+1} = x^i - \alpha^i d^i$
- $\approx$ "heavy ball subgradient" (a bit $\neq$, since $x^{i+1} - x^i \neq d^i$)
- Previous proofs won't work, $d^i \notin \partial f(x^i)$

Deflected subgradient

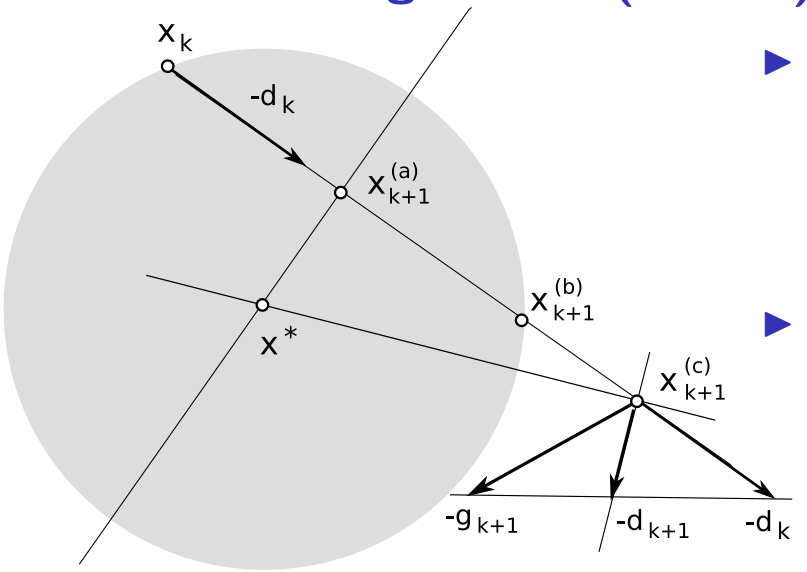
- "Want a better direction? Use a better model!"
- But there is no second-order information (well, there might be, but . . .)
- Yet, deflection is possible: $d^i = \gamma^i g^i + (1 - \gamma^i) d^{i-1}$, $x^{i+1} = x^i - \alpha^i d^i$
- $\approx$ "heavy ball subgradient" (a bit $\neq$, since $x^{i+1} - x^i \neq d^i$)
- Previous proofs won't work, $d^i \notin \partial f(x^i)$
  . . . but $d^i \in \partial_{\varepsilon^i} f(x^i)$ for $\varepsilon^i \geq 0$ that can be easily computed

Exercise: find the formula $\varepsilon^i \longrightarrow \varepsilon^{i+1}$ (hint: use information transport)

Exercise: what conditions on γ^i for this to work? why?

- Prove $\{ \varepsilon^i \} \to 0$, get convergence
- However, this is not free: have to check stepsize and/or deflection
- It depends which one you choose first

Deflected subgradient (cont.d)



- Stepsize-restricted $\equiv$ deflection-first:
  $\alpha^i = \beta^i(f(x^i) - f_*)/\| d^i \| \quad \wedge \quad \beta^i \le \gamma^i$
  "as deflection $\nearrow$, stepsize has to $\searrow$"
- Deflection-restricted $\equiv$ stepsize-first: (DSS)
  $$\wedge \quad \frac{\alpha^{i-1}\| d^{i-1} \|}{(f(x^i) - f_*) + \alpha^{i-1}\| d^{i-1} \|} \le \gamma^i$$
  "as $f(x^i) \to f_*$, deflection $\searrow$"

- In both cases, target level to replace f_*
- $\gamma^i \in \operatorname{argmin}\{ \| \gamma g^i + (1 - \gamma) d^{i-1} \|^2 \ : \ \gamma \in [0,1] \}$
  or a bit smarter variants thereof (using ε^i)

Exercise: explain why the formula makes sense: how would you use ε^i?

- This actually does help in practice
- Can be extended to $g^i \in \partial_{\sigma^i} f(x^i)$ (subgradient inexact to start with)

Primal-Dual Subgradient

- "Primal"? "Dual"? What??
- Hypotheses: f (convex) Lipschitz ($L > 0$), X bounded, diameter $D > 0$

Exercise: are both conditions needed? why?

- $\{ \nu_i \}$: Simple Averages $\equiv \nu_i = 1$, Weighted Averages $\equiv \nu_i = 1 / \| g^i \|$
- $\{ \hat{\omega}^i \}$: $\hat{\omega}^1 = 1$, $\hat{\omega}^{i+1} = \hat{\omega}^i + 1/\hat{\omega}^i$ (clear signs of black magic . . .)
- $\omega^i = L\hat{\omega}^i / \sqrt{2D}$ for SA, $\omega^i = \hat{\omega}^i / \sqrt{2D}$ for WA
- $\Delta^i = \sum_{k=1}^{i} \nu^i$, $\gamma^i = \nu^i / \Delta^i$, $\alpha^i = \Delta^i / \omega^i$ (er . . . what??)

Primal-Dual Subgradient

- "Primal"? "Dual"? What??
- Hypotheses: f (convex) Lipschitz ($L > 0$), X bounded, diameter $D > 0$

Exercise: are both conditions needed? why?

- $\{ \nu_i \}$: Simple Averages $\equiv \nu_i = 1$, Weighted Averages $\equiv \nu_i = 1 \,/\, \| g^i \|$
- $\{ \hat{\omega}^i \}$: $\hat{\omega}^1 = 1$, $\hat{\omega}^{i+1} = \hat{\omega}^i + 1/\hat{\omega}^i$ (clear signs of black magic ...)
- $\omega^i = L\hat{\omega}^i \,/\, \sqrt{2D}$ for SA, $\omega^i = \hat{\omega}^i \,/\, \sqrt{2D}$ for WA
- $\Delta^i = \sum_{k=1}^{i} \nu^i$, $\gamma^i = \nu^i \,/\, \Delta^i$, $\alpha^i = \Delta^i \,/\, \omega^i$ (er ... what??)
  $\equiv d^i = (\sum_{k=1}^{i} \nu^i g^i) \,/\, \Delta^i$ (ah ... this explains all ... ??)
- Based on "usual" analysis of worst-case behaviour, attains "optimal" complexity $O(1 \,/\, \varepsilon^2)$ (oh, wow ...)
- Completely parameter free, very good

Primal-Dual Subgradient

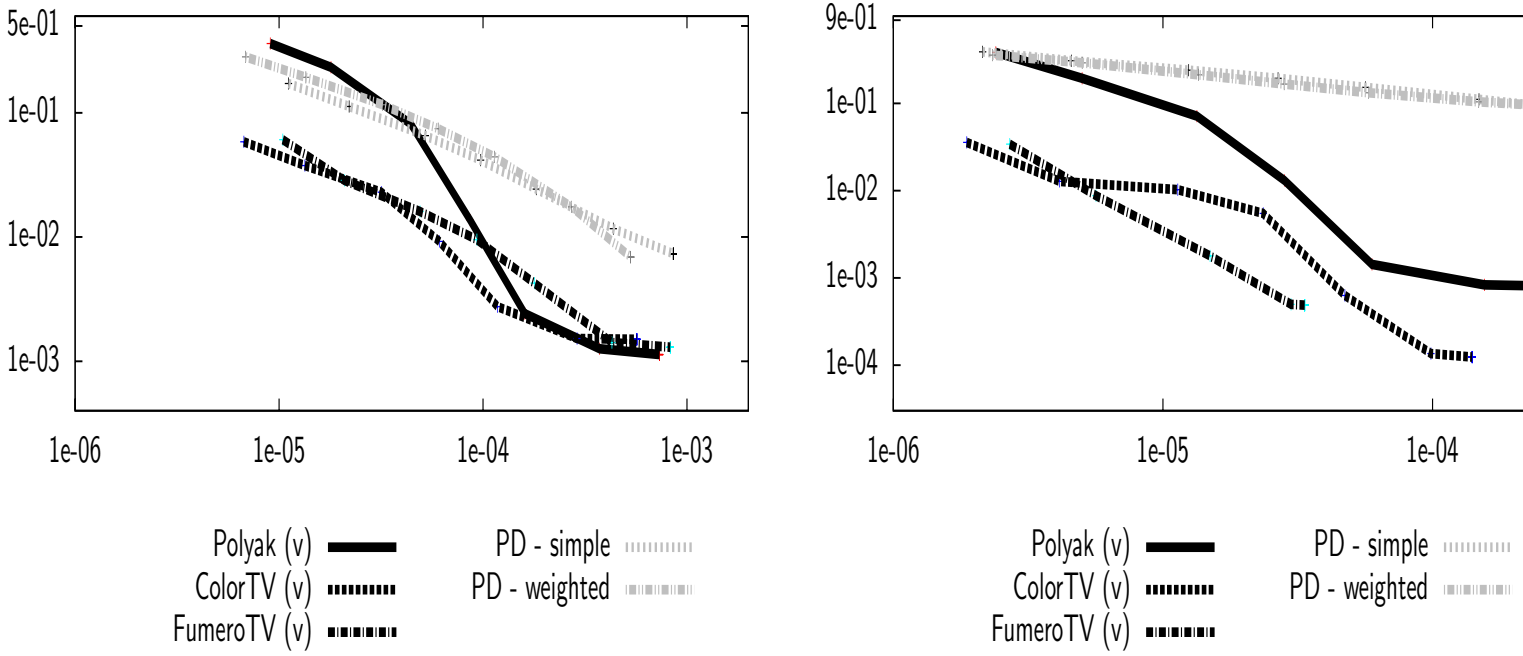
- "Primal"? "Dual"? What??
- Hypotheses: f (convex) Lipschitz ($L > 0$), X bounded, diameter $D > 0$

Exercise: are both conditions needed? why?

- $\{ \nu_i \}$: Simple Averages $\equiv \nu_i = 1$, Weighted Averages $\equiv \nu_i = 1 / \| g^i \|$
- $\{ \hat{\omega}^i \}$: $\hat{\omega}^1 = 1$, $\hat{\omega}^{i+1} = \hat{\omega}^i + 1/\hat{\omega}^i$ (clear signs of black magic ...)
- $\omega^i = L\hat{\omega}^i / \sqrt{2D}$ for SA, $\omega^i = \hat{\omega}^i / \sqrt{2D}$ for WA
- $\Delta^i = \sum_{k=1}^{i} \nu^i$, $\gamma^i = \nu^i / \Delta^i$, $\alpha^i = \Delta^i / \omega^i$ (er ... what??)
  $\equiv d^i = (\sum_{k=1}^{i} \nu^i g^i) / \Delta^i$ (ah ... this explains all ... ??)
- Based on "usual" analysis of worst-case behaviour, attains "optimal" complexity $O(1 / \varepsilon^2)$ (oh, wow ...)
- Completely parameter free, very good ... if it were true, but it isn't:
  L, D unknown in practice, have to use parameters

Primal-Dual Subgradient in practice

- How does this work in practice? Consistently slowish (sounds familiar?) ≈ linear in a doubly-logarithmic chart (ugh!)



- "Constructed to optimize worst-case behaviour" ≡ "Carefully crafted to attain a given speed, typically gets what it is constructed for"
- Faster versions have many more parameters to tune
- Using "exact" estimates of L and D does not necessarily help
- Exploiting information about f_*, even if inaccurate, helps
- No subgradient algorithm ever reliably gets any better than $\varepsilon = 1e{-}4$
- Disclaimer: this but one application, your mileage may vary

Outline

Smoothed gradient methods

- "Want a better direction? Use a better model!"
- You said "there is no second-order information", boss; furthermore, whatever first-order information there is, is crap
- What if I slightly modify f so that it is better?
- Only works with f of a specific kind, i.e.,
  $f(x) = \max\{ x^T A z \,:\, z \in Z \}$, Z convex and bounded
  (will be called a "Lagrangian function" later on ...)
- Actually works for $g(x) = f(x) + h(x)$ with $h \in C^1$ (and Lipschitz)
- Actually works with an extra term "$+\phi(z)$" with ϕ concave
- $I(x) = \operatorname{argmax}\{ x^T A z \,:\, z \in Z \}$, $\partial f(x) = conv(\{ Az \,:\, z \in I(x) \})$

Exercise: you are $\approx$ able to show that: under what conditions?

- $|I(x)| > 1 \approx\Longrightarrow f$ nondifferentiable at x

Exercise: provide a fitting example for the above statement

Smoothed gradient methods (cont'd)

- Dirty trick: $f_\mu(x) = \max\{\, x^T A z - \mu \| z \|_2^2/2 \,:\, z \in Z \,\}$
- $R = \max\{\, \| z \|_2^2/2 \,:\, z \in Z \,\} < \infty$ (**why?**)
  (note: maximizing a convex function, this is hard in general)
- $f_\mu(x) \le f(x) \le f_\mu(x) + \mu R$: as $\mu \searrow 0$, "argmin $\{\, f_\mu(x) \,\} \to x_*$"
- $|I_\mu(x)| = 1 \; \forall x$ (**why?**) $\implies$ f differentiable (**why?**)
- If f is "easy", then f_μ can be "easy": $f(x) = |\, x \,| \implies$

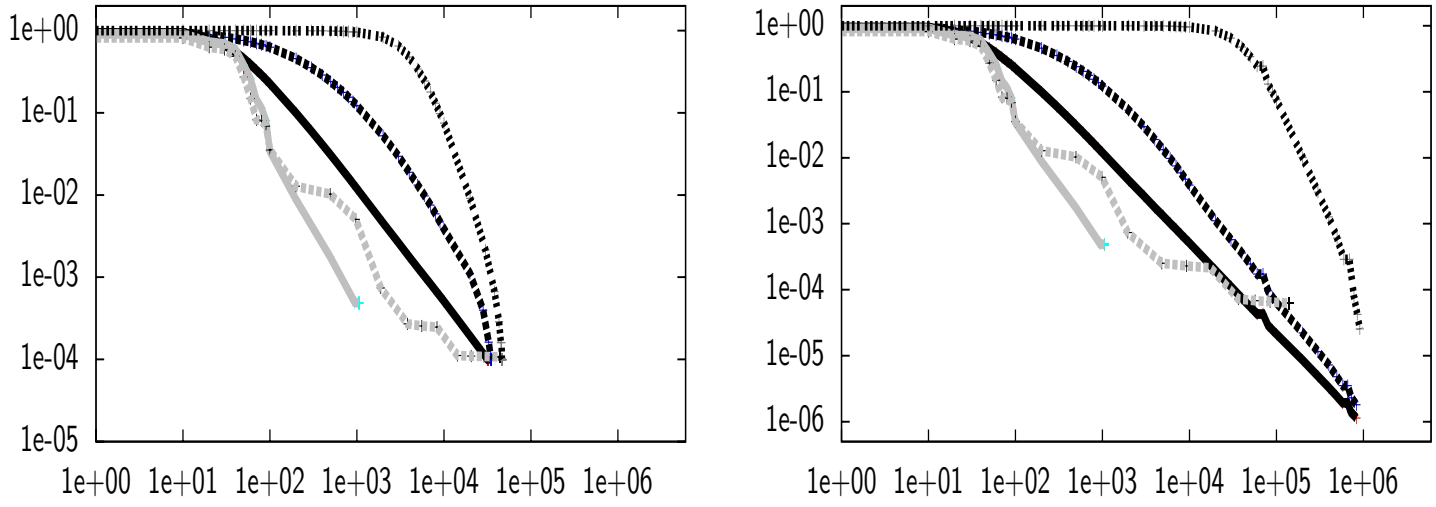
$$f_\mu(x) = \begin{cases} x^2 \,/\, (2\mu) & \text{if} |\, x \,| \le \mu \\ |\, x \,| - \mu \,/\, 2 & \text{if} |\, x \,| \ge \mu \end{cases}$$

Exercise: prove the above formula then draw f_μ: is "smoothed" appropriate?

- ∇f_μ Lipschitz with $L = \| A \|^2 \,/\, \mu$ ("less and less Lipschitz" as $\mu \searrow 0$)
- If $f_* > -\infty$ and $\mu = \varepsilon \,/\, (2R)$, then an appropriate ACCG obtains
  $f(x^i) - f_* \le \varepsilon$ for $i \ge 4 \| A \| \| x_* \| \sqrt{R} \,/\, \varepsilon$
- "Only" $O(\, 1 \,/\, \varepsilon \,)$ instead of $O(\, 1 \,/\, \varepsilon^2 \,)$, "much better"
- Almost parameter-free (have to estimate R)

Smoothed Gradient in practice

- How does this work in practice? Consistently slowish (sounds familiar?)
  ≈ superlinear in a doubly-logarithmic chart after a very long flat leg



- "Constructed to optimize worst-case behaviour" blah blah ...
- Subgradients can be faster but stop at $\varepsilon = 1e{-}4$ or less
  smoothed goes all the way down to $\varepsilon = 1e{-}6$
- But with $\varepsilon = 1e{-}6$ the flat leg is way longer
- ACCG does steps $1 / L_\mu = O(\mu) = O(\varepsilon)$, far too short at start
- Exploiting information about f_* helps (black solid line)

Exercise: how would you exploit information about f_*? (hint: $\varepsilon \longrightarrow \varepsilon^i$)

- Disclaimer: this but one application, your mileage may vary

Outline

The basic idea: cutting-plane model

- "Want a better direction? Use a better model!"
- You said "there is no second-order information", boss; furthermore, whatever first-order information there is, is crap, and my function has not the right form for smoothing (or the max problem is too hard)
- You see, my son, in the convex case first-order information is not so crap: it is globally valid, not only locally
- What if I just collect a bunch of it and use it all?
- $\{ x^i \} \longrightarrow \mathscr{B} = \{ (x^i , f^i = f(x^i) , g^i \in \partial f(x^i)) \} \equiv$ bundle of first-order information
- $f_{\mathscr{B}}(x) = \max\{ f^i + g^i(x - x^i) : (x^i , f^i , g^i) \in \mathscr{B} \} \equiv$ cutting-plane model of f (first-plus-ε-order model)
- $f_{\mathscr{B}}(x) \leq f(x) \; \forall \, x$ (**why?**) $\Longrightarrow \min\{ f_{\mathscr{B}}(x) \} \leq f_* \Longrightarrow$ $x^*_{\mathscr{B}} \in \operatorname{argmin}\{ f_{\mathscr{B}}(x) \} \approx x_*$
- Can use $x^*_{\mathscr{B}}$ as my next iterate (sounds familiar?)

Master problem

- But $f_{\mathscr{B}}$ is ferociously nondifferentiable, boss (**why?**), how do I find $x^*_{\mathscr{B}}$?
- $f_{\mathscr{B}}$ polyhedral function, max of finitely many (few) linear functions
- Dirty trick from Ricerca Operativa:

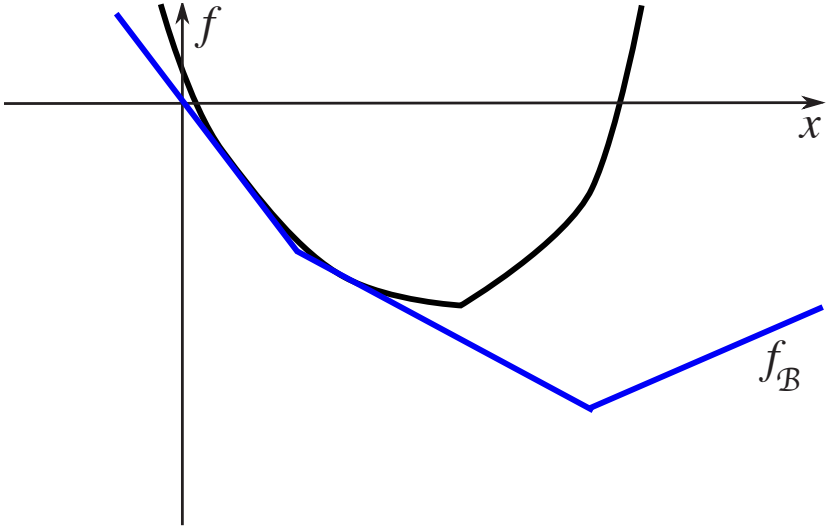
  $$\min\{\, f_{\mathscr{B}}(x) \,\} = \min\{\, v \,:\, v \geq f^i + g^i(\, x - x^i \,) \;\; (\, x^i \,,\, f^i \,,\, g^i \,) \in \mathscr{B} \,\}$$

- This thing has constraints, boss! How do I solve it?
- Everything is linear $\equiv$ a Linear Program
- Somebody will happily solve it for you, my son, quickly if $\mathscr{B}$ not too big
- For instance the Primal or Dual (what?) Simplex method
- This is a large-scale problem if n is large ...

Master problem

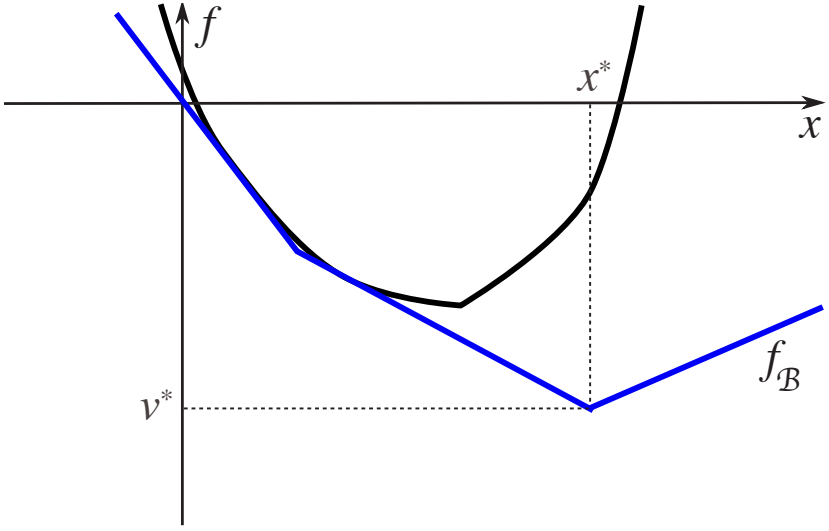
- But $f_{\mathscr{B}}$ is ferociously nondifferentiable, boss (**why?**), how do I find $x^*_{\mathscr{B}}$?
- $f_{\mathscr{B}}$ polyhedral function, max of finitely many (few) linear functions
- Dirty trick from Ricerca Operativa:
  $$\min\{\, f_{\mathscr{B}}(x)\,\} = \min\{\, v \,:\, v \geq f^i + g^i(\,x - x^i\,) \;\; (\,x^i\,,\, f^i\,,\, g^i\,) \in \mathscr{B}\,\}$$
- This thing has constraints, boss! How do I solve it?
- Everything is linear $\equiv$ a Linear Program
- Somebody will happily solve it for you, my son, quickly if $\mathscr{B}$ not too big
- For instance the Primal or Dual (what?) Simplex method
- This is a large-scale problem if n is large ...
  but its dual is smallish if $\mathscr{B}$ is not large (what?)
- Just trust me, it can be considered

The Cutting Plane algorithm



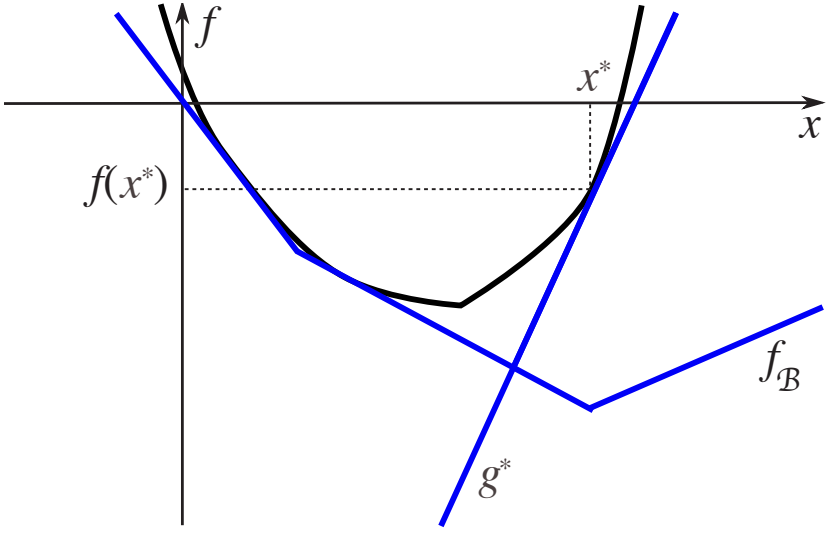
- $\min\{ f_{\mathcal{B}}(x) \}$ master problem

The Cutting Plane algorithm



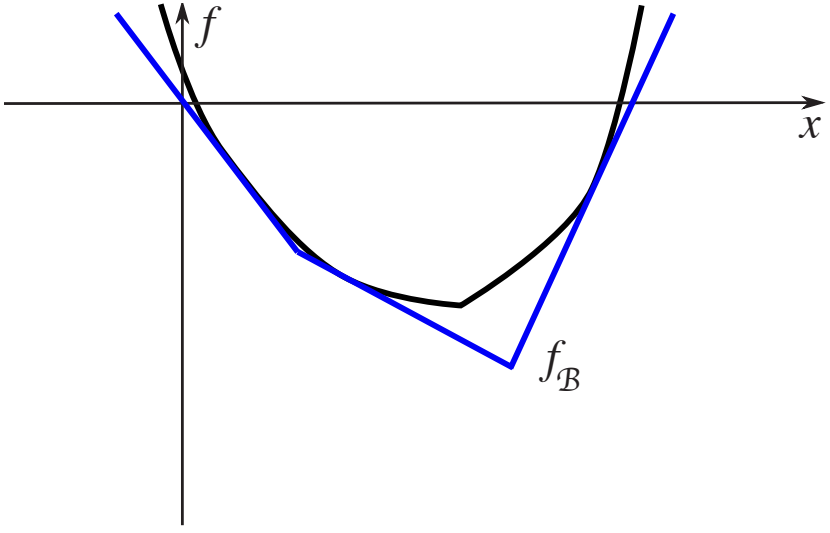
- $\min\{ f_{\mathscr{B}}(x) \}$ master problem
- (x^* , v^*) optimal solutions

The Cutting Plane algorithm



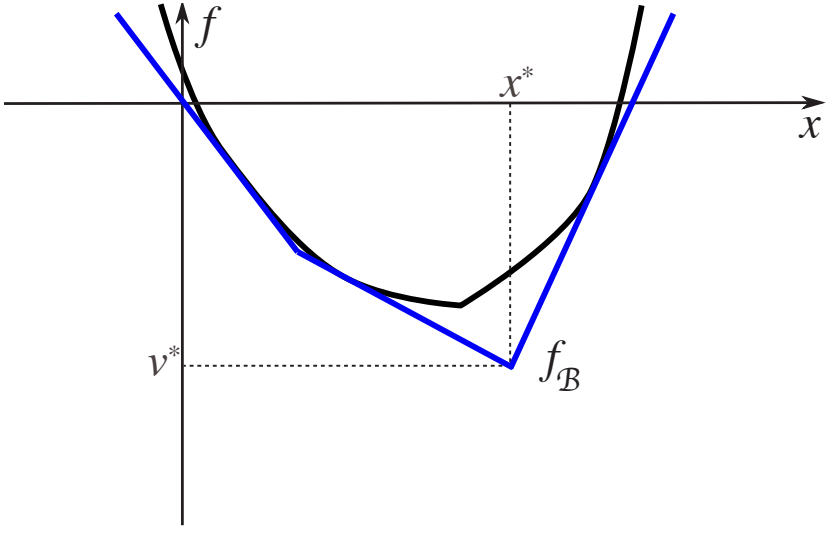
- $\min\{ f_{\mathcal{B}}(x) \}$ master problem
- (x^* , v^*) optimal solutions
- $(x^* , f(x^*) , g^* \in \partial f(x^*))$
- if $f(x^*) \leq v^*$ then x^* optimal (**why?**)

The Cutting Plane algorithm



- $\min\{ f_{\mathcal{B}}(x) \}$ master problem
- (x^* , v^*) optimal solutions
- $(x^* , f(x^*) , g^* \in \partial f(x^*))$
- if $f(x^*) \leq v^*$ then x^* optimal (**why?**)
- otherwise $\mathcal{B} \leftarrow \mathcal{B} \cup (x^* , f(x^*) , g^*)$

The Cutting Plane algorithm



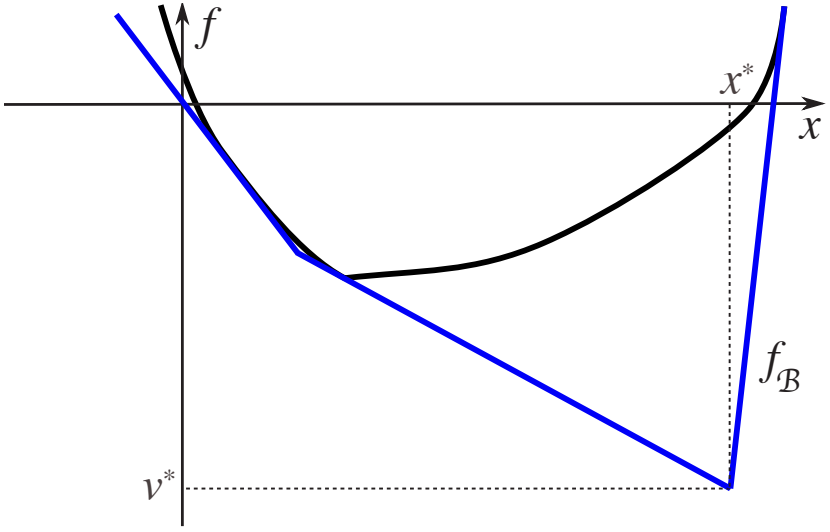
- $\min\{ f_{\mathscr{B}}(x) \}$ master problem
- (x^* , v^*) optimal solutions
- $(x^* , f(x^*) , g^* \in \partial f(x^*))$
- if $f(x^*) \leq v^*$ then x^* optimal (**why?**)
- otherwise $\mathscr{B} \leftarrow \mathscr{B} \cup (x^* , f(x^*) , g^*)$
- Hopefully, $x^*_{\mathscr{B}} \to x_*$, $v^*_{\mathscr{B}} \to f_*$

- Actually globally convergent, which is not surprising:
  f convex $\Longrightarrow f(x) = \max\{ f(y) + g(x - y) : y \in \mathbb{R}^n \ , \ g \in \partial f(y) \}$
- Practical stopping criterion, unlike any subgradient-stuff

Exercise: prove in at least two ways that if the algorithm stops then $x^*_{\mathscr{B}} = x_*$

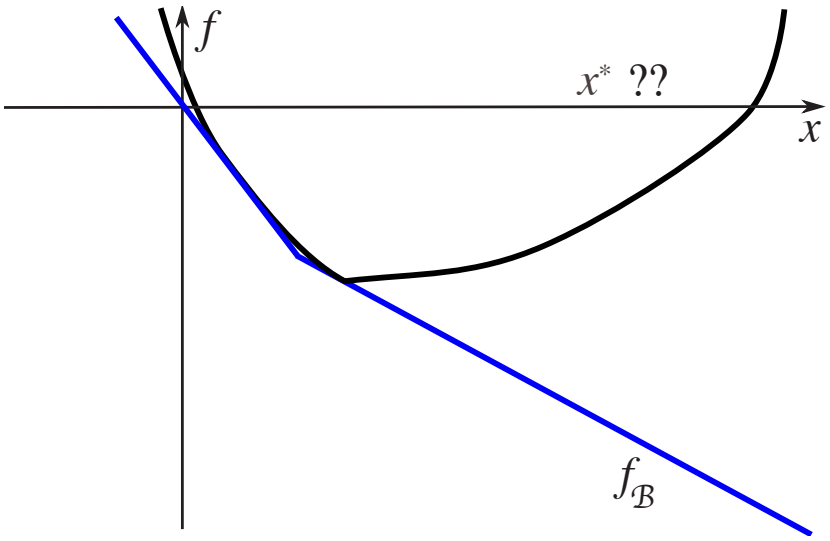
- But $|\mathscr{B}|$ may have to $\to \infty$
- Practical convergence speed is horrible
- A host of other problems

Why the Cutting Plane algorithm is so bad



- $x^*_{\mathcal{B}}$ may be very far from x_*

Why the Cutting Plane algorithm is so bad



- $x^*_{\mathscr{B}}$ may be very far from x_*
  ... up to infinitely far $\Longrightarrow$
  some dirty trick when $\mathscr{B}$ is small
- Iterates have no locality property:
  $\| x^{i+1} - x^i \|$ can be very large and
  does not go "smoothly" to 0
- Forget "fast convergence in the tail"

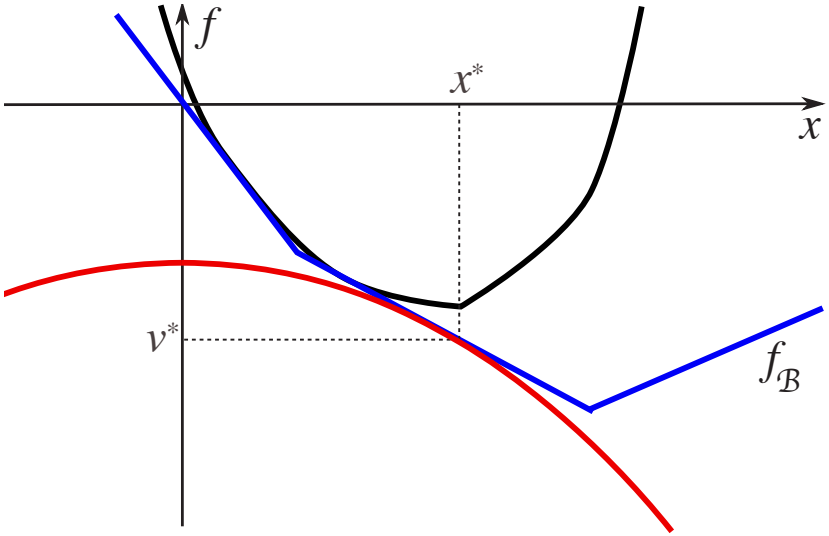
- The algorithm is unstable ($\Longrightarrow$ hard to analyze)
- $\approx$ unavoidable: linear functions have no curvature (really?),
  you need very many linear functions to make a quadratic one
- Many iterations $\Longrightarrow |\mathscr{B}| \nearrow \Longrightarrow$ the master problem grows costly
- Pruning $\mathscr{B}$ possible but not easy, no a-priori bound on maximum $|\mathscr{B}|$

Exercise: how would you discard elements from $\mathscr{B}$ retaining convergence?

- All in all, looks better than subgradient but impractical as it is

Bundle methods

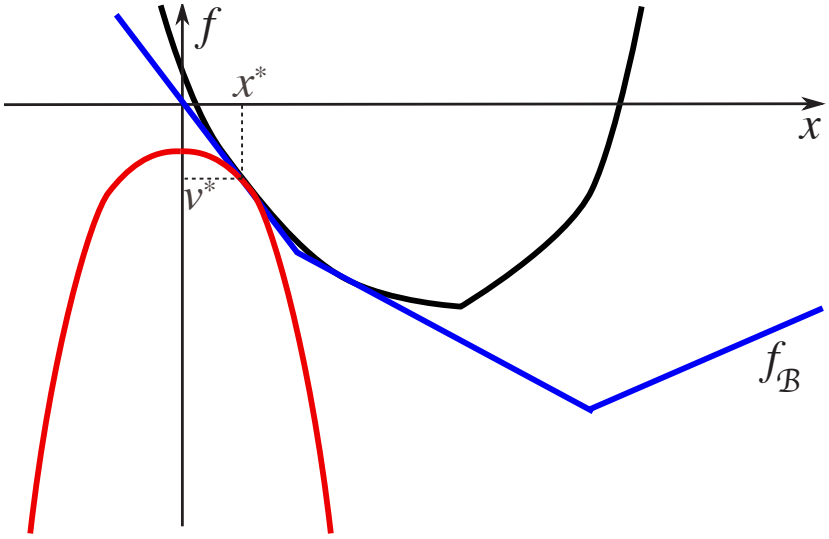
- "If something is unstable, then stabilize it" (a.k.a. "regularize")



- $\bar{x}$ stability center ($\approx$ best x^i so far)
- μ stability parameter, "how far from $\bar{x}$ I can trust $f_{\mathscr{B}}$" (?? who knows ??)
- Stabilized master problem: $\min\{ f_{\mathscr{B}}(x) + \mu \| x - \bar{x} \|_2^2/2 \}$
- Keeps $x^*_{\mathscr{B}}$ "close" to $\bar{x}$

Bundle methods

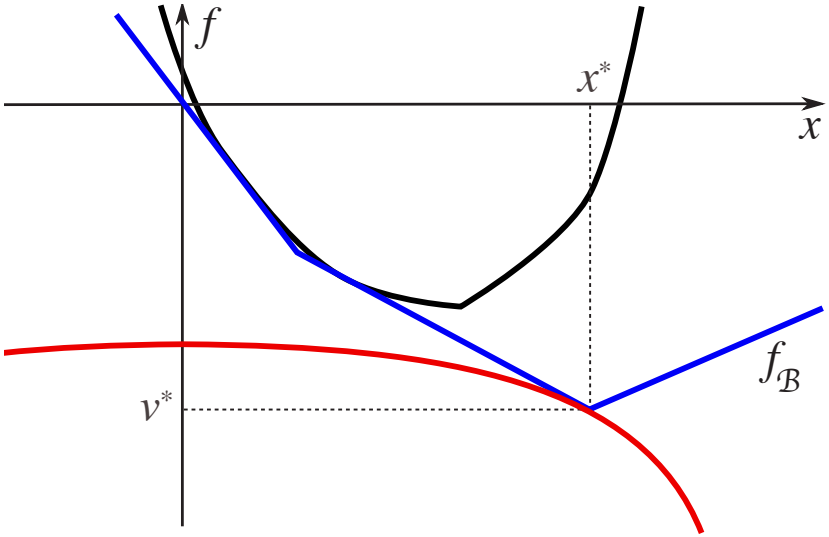
- "If something is unstable, then stabilize it" (a.k.a. "regularize")



- $\bar{x}$ stability center ($\approx$ best x^i so far)
- μ stability parameter, "how far from $\bar{x}$ I can trust $f_{\mathcal{B}}$" (?? who knows ??)
- Stabilized master problem:
  $\min\{ f_{\mathcal{B}}(x) + \mu \| x - \bar{x} \|_2^2/2 \}$
- Keeps $x^*_{\mathcal{B}}$ "close" to $\bar{x}$
  perhaps too close (μ too large)

Bundle methods

- "If something is unstable, then stabilize it" (a.k.a. "regularize")

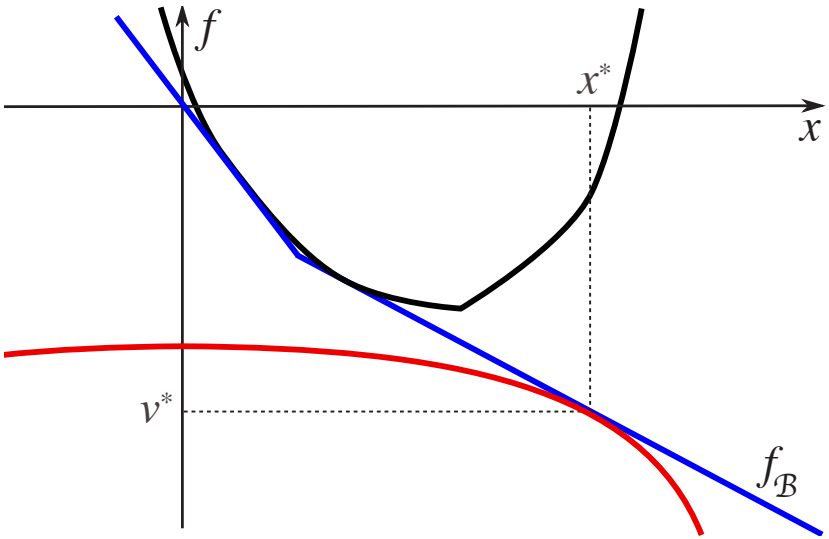


  - $\bar{x}$ stability center ($\approx$ best x^i so far)
  - μ stability parameter, "how far from $\bar{x}$ I can trust $f_{\mathscr{B}}$" (?? who knows ??)
  - Stabilized master problem: $\min\{ f_{\mathscr{B}}(x) + \mu \| x - \bar{x} \|_2^2/2 \}$
  - Keeps $x^*_{\mathscr{B}}$ "close" to $\bar{x}$ perhaps too close (μ too large)

- μ too small $\equiv$ un-stabilized cutting plane

Bundle methods

- "If something is unstable, then stabilize it" (a.k.a. "regularize")



  - $\bar{x}$ stability center ($\approx$ best x^i so far)
  - μ stability parameter, "how far from $\bar{x}$ I can trust $f_{\mathscr{B}}$" (?? who knows ??)
  - Stabilized master problem: $\min\{ f_{\mathscr{B}}(x) + \mu \| x - \bar{x} \|_2^2/2 \}$
  - Keeps $x^*_{\mathscr{B}}$ "close" to $\bar{x}$ perhaps too close (μ too large)

- μ too small $\equiv$ un-stabilized cutting plane

  except always bounded below (**why?**) $\Longrightarrow$ $x^*_{\mathscr{B}}$ always well-defined

Exercise: explain why the curious upside-down parabola graphically finds x^*

- Enforces stability $\approx$ trust region ($\nabla^2 f \not\succeq 0$); in fact other versions $\exists$, among which trust region, this one being proximal
- Graft "poorman's Hessian" μI onto $f_{\mathscr{B}}$ $\Longrightarrow$ "poorman's Newton"
- No longer a LP, but still OK: smallish convex quadratic program
- But how do I manage $\bar{x}$ and μ?

Analysis of Bundle methods

procedure $\bar{x} = PBM\,(\,f\,,\,g\,,\,\bar{x}\,,\,m_1\,,\,\varepsilon\,)\;\;\{$
  choose μ; $\mathscr{B} \leftarrow \{\,(\,\bar{x}\,,\,f(\bar{x})\,,\,g(\bar{x})\,)\,\}$;
  while (**true**) **do** $\{$
    $x^* \leftarrow \operatorname{argmin}\,\{\,f_{\mathscr{B}}(x) + \mu\|\,x - \bar{x}\,\|_2^2/2\,\}$;
    if($\mu\|\,x^* - \bar{x}\,\|_2 \le \varepsilon$) **then break**;
    if($f(x^*) \le f(\bar{x}) + m_1(\,f_{\mathscr{B}}(x^*) - f(\bar{x})\;$) **then** $\bar{x} \leftarrow x^*$; possibly decrease μ;
                                                else possibly increase μ;
    $\mathscr{B} \leftarrow \mathscr{B} \cup (\,x^*\,,\,f(x^*)\,,\,g(x^*)\,)$; $\}$
$\}$

- $\bar{x} \leftarrow x^*$ called a Serious Step (SS), $\bar{x}$ unchanged a Null Step (NS)
- How to increase/decrease μ? Heuristics $\equiv$ parameters, parameters, ...
- $-\mu(\,x^* - \bar{x}\,) \in \partial f_{\mathscr{B}}(x^*)$ (**check**. hint: $0 \in \partial[\,f_{\mathscr{B}}(\cdot) + \mu\|\;\cdot - \bar{x}\,\|_2^2\,](x^*)$)
  $\Longrightarrow f_{\mathscr{B}}(x^*) - f(\bar{x}) \le -\mu\|\,x^* - \bar{x}\,\|_2^2$ (**check**: use $f_{\mathscr{B}}(\bar{x}) = f(\bar{x})$ **why?**)
- ∞ SS made $\Longrightarrow$ either $f(\bar{x}) \to -\infty$ or $\|\,x^* - \bar{x}\,\| \to 0$

Exercise: an hypothesis is missing: find which and prove it

Exercise: this is clearly an Armijo-type rule, explain exactly why

Analysis of Bundle methods (cont'd)

- Complicated part: $|\mathrm{SS}| < \infty \Longrightarrow \infty$ consecutive NS $\Longrightarrow \| x^* - \bar{x} \| \to 0$
- Intuitively clear: $\bar{x}$ fixed and $|\mathscr{B}| \nearrow \Longrightarrow$ "$f_{\mathscr{B}} \to f$" around $\bar{x}$
- Proof easy with the dual master problem, we don't know it (yet), we skip
- $\varepsilon > 0 \Longrightarrow$ the algorithm finitely stops and $\bar{x}$ is $\approx$ optimal

Exercise: check what "$\approx$ optimal" means here (hint: consider $\| x^* - \bar{x} \| = 0$)

- Easy to see which (x^i , f^i , g^i) can be eliminated from $\mathscr{B}$
  if you know what the dual master problem looks like, which we don't
- Can "compress $\mathscr{B}$" down to $|\mathscr{B}| = 2$ using the dual master problem
- Trade-off: $|\mathscr{B}| \nearrow \Longrightarrow$ iterations $\searrow$ but master problem cost $\nearrow$
- $|\mathscr{B}| \approx 2 \Longrightarrow$ Bundle $\approx$ subgradient (but working stopping criterion)
- You have to pay a "fat" $\mathscr{B}$ for a decent convergence rate
- It actually pays to make $\mathscr{B}$ as fat as you can with dirty tricks

Bundle methods in practice

- Dirty trick I: $f = f^1 + f^2 + \ldots f^k \Longrightarrow f_{\mathscr{B}} = f^1_{\mathscr{B}} + f^2_{\mathscr{B}} + \ldots f^k_{\mathscr{B}}$
  "the sum of the models is much better than the model of the sum"
- Dirty trick II: $f = f^1 + f^2$ with f^2 "easy" (say, quadratic) $\Longrightarrow$
  put f^2 in the maser problem instead of $f^2_{\mathscr{B}}$
- Yet other dirty tricks (special forms of $f_{\mathscr{B}}$ for specific f)
- One example ($\varepsilon =$ 1e−6 relative, $i_{max} = 5000$ for SG)

SG			$\mathscr{B}$ "very fat"		$\mathscr{B}$ "slim"	
time	iter	gap	time	iter	time	iter
5.39	4738	1e-4	16.34	30	10.44	8084
26.16	4903	2e-4	188.33	32	82.67	9830
17.45	4051	1e-4	16.77	26	38.18	5920
52.98	3093	1e-4	61.28	25	84.30	3761
10.74	3580	1e-4	3.69	48	33.20	8985
180.41	4900	1e-4	76.22	25	168.72	4174

- It may pay to invest on a "very fat" $\mathscr{B}$, but care is required
- At least it knows when to stop
- Your mileage may vary

Outline

Wrap up

- Lack of continuous derivatives is un-good for optimization
- Lack of derivatives is double-plus-un-good for optimization, we don't even talk about it (although ML could use it, e.g. hyperparameters tuning)
- Nonsmooth algorithms are in general sadly slow
- Forget getting anything better than $\varepsilon = \texttt{1e-4}$
  unless you are prepared to fight hard
- Good news: `1e-4` (or less) can be OK for ML
- You can also fight hard and get something better
- Either you cheat on the function, or you work with a fat model
- All approaches nontrivial
- We have heard "Dual" quite too many times so far: what's that?
- Duality is about constraints (well, not exactly, but . . .), so
  high time that we move to constrained optimization