Unconstrained optimization II
More-than-gradient methods

Antonio Frangioni

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

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

Outline

Different descent directions

- So far the analysis has hinged on $d^i = -\nabla f(x^i)/\| \nabla f(x^i) \| \ldots$

Different descent directions

- So far the analysis has hinged on $d^i = -\nabla f(x^i)/\| \nabla f(x^i) \| \ldots$
  or has it?
- Crucial arguments in the convergence proofs:
  "the derivative promises a significant descent along direction d^i" +
  "you can get a non-vanishing fraction of the promised descent" (LS)
  $\Longrightarrow$ the algorithm converges
- "Significant descent" $\equiv$ non-vanishing fraction of $\| \nabla f(x^i) \|$, because
  $\varphi'(0) = \frac{\partial f}{\partial d^i}(x^i) = \langle\, d^i\,,\, \nabla f(x^i)\,\rangle = -\langle\, \nabla f(x^i)\,,\, \nabla f(x^i)\,\rangle = -\| \nabla f(x^i) \|$
  and $\| \nabla f(x^i) \|$ has to go $\to 0$
- But what if $d^i =$ "$-\nabla f(x^i)/\| \nabla f(x^i) \|$ rotated by 45 degrees"?
- $\varphi'(0) = \langle\, d^i\,,\, \nabla f(x^i)\,\rangle = \| \nabla f(x^i) \| cos(\pi/4) \equiv$ just a different constant
- You can expect the convergence proofs to carry over, provided the angle is not too small
- And in fact they do

Convergence of general descent methods

- Descent direction $\equiv \frac{\partial f}{\partial d^i}(x^i) < 0 \equiv \langle\, d^i\,,\, \nabla f(x^i)\,\rangle < 0 \equiv cos(\theta^i) \geq 0$
  "d^i points roughly in the same direction as $-\nabla f(x^i)$"
- There is a whole half space of descent directions $\Longrightarrow$ a lot of flexibility
- "d^i must not be almost orthogonal to $\nabla f(x^i)$" $\equiv cos(\theta^i)$ "too small"
- One general result (Zoutendijk): $f \in C^1$, ∇f Lipschitz, f bounded below
  (A) $\cap$ (W') $\Longrightarrow \sum_{i=1}^{\infty} cos^2(\,\theta^i) \| \nabla f(x^i) \|^2 < \infty$ (er . . . what??)
- $cos(\,\theta^i)$ bounded away from 0 $\Longrightarrow \| \nabla f(x^i) \| \to 0$
  (if every element in a series is bounded away from 0, the sum is ∞)
- Proof not too difficult, a bit technical, let's do without
- Gradient method is just the obvious case, $cos^2(\,\theta^i) = 1$
- Note: a (≥ 0) sequence has to $\searrow 0$ quite fast for the series to be $< \infty$
  i.e., $\sum_{i=1}^{\infty} a_i < \infty \Longrightarrow \{\, a_i \,\} \to 0$ but vice-versa not true
- $cos^2(\,\theta^i) \searrow 0$ also possible, provided "not too fast"

Exercise: find an $\{\, a_i\}$ providing the counter-example

Outline

Newton's method

- Want a better direction? Use a better model!
- Next better model to linear ($\equiv$ gradient): quadratic
- Assume $\nabla^2 f(x^i) \succ 0$: $\exists$ minimum of second-order model
$$f(x^i) + \nabla f(x^i)(\, x - x^i \,) + \tfrac{1}{2}(x - x^i)^T \nabla^2 f(x^i)(x - x^i)$$
- Newton's direction: $d^i \leftarrow -[\, \nabla^2 f(x^i) \,]^{-1} \nabla f(x^i)$ (just $\mathbb{R}^n$ version)
- No problem with the step here, $\alpha^i = 1$ (the minimum $\exists$)
$\Longrightarrow$ Newton's method: just do step $\alpha^i = 1$ along d^i
- Nonlinear equation interpretation: want to solve $\nabla f(x) = 0$, write $\nabla f(x) \approx \nabla f(x^i) + \nabla^2 f(x^i)(\, x - x^i \,)$ and solve linear equation instead
- $f \in C^3$, $\nabla f(x_*) = 0 \wedge \nabla^2 f(x_*) \succ 0 \Longrightarrow \exists\, \mathcal{B}(x_*, r)$ s.t. $x^1 \in \mathcal{B} \Longrightarrow$ $\{\, x^i \,\} \to x_*$ quadratically (basically same proof as for $n = 1$)
- Again, convergence only local (albeit fast)
- However, d^i is of descent if $\nabla^2 f(x^i) \succ 0 \Longrightarrow [\, \nabla^2 f(x^i) \,]^{-1} \succ 0$:
$$d^i \nabla f(x^i) = -\nabla f(x^i)^T [\, \nabla^2 f(x^i) \,]^{-1} \nabla f(x^i) < 0$$
- But is it enough of descent?

Interpretation: Newton = Gradient + Space dilation

- Interesting interpretation: Newton = Gradient in a $\neq$ space
- Relevant object: $Q \succeq 0 \Longrightarrow Q = RR$, $R = Q^{1/2}$ square root of Q
- R $\exists$ and is symmetric, e.g. $Q = H\Lambda H^T \Longrightarrow R = H\sqrt{\Lambda}H^T$ (**check**)
- $f(x) = \frac{1}{2}x^T Qx + qx$, $\nabla f(x) = Qx + q$, $d = -Q^{-1}q$
- $y = Rx \equiv x = R^{-1}y$, $\mathbb{R}^n \to \mathbb{R}^n$ because R nonsingluar (**why?**)
- $f(y) = \frac{1}{2}y^T Iy + qR^{-1}y$, $\nabla f(y) = y + R^{-1}q$, $d = -R^{-1}q$
- Translate direction from y-space to x-space $\Longrightarrow d = -Q^{-1}q$
- $\lambda^1(I) = \lambda^n(I) \Longrightarrow$ gradient terminates in one iteration in y-space
- Indeed, Newton terminates in one iteration in x-space (doh!)
- In general, "Newton = Gradient in a space where $\nabla^2 f(x^i)$ looks like I"
- One should expect it to converge fast, and in fact it does

Global convergence of Newton's method

- Global convergence if $cos(\theta^i) = d^i \nabla f(x^i) / (\| d^i \| \| \nabla f(x^i) \|) \leq \delta < 0$
- "Usual" assumption $uI \preceq \nabla^2 f \preceq LI$ (for the moment)
- Two technical steps:
  - $\nabla^2 f(x^i) d^i = -\nabla f(x^i) \Longrightarrow d^i \nabla f(x^i) = -(d^i)^T \nabla^2 f(x^i) d^i \leq -\lambda^n \| d^i \|^2$
  - $\| \nabla f(x^i) \| = \| \nabla^2 f(x^i) d^i \| \leq \| \nabla^2 f(x^i) \| \| d^i \| = \lambda^1 \| d^i \|$

  $\Longrightarrow cos(\theta^i) \leq -\lambda^n / \lambda^1 \leq -u/L$ (**check**) $\Longrightarrow$ global convergence
- Convergence is also fast: whenever $\{ x^i \} \to x_*$ ($\Longrightarrow \nabla f(x_*) = 0$) with $\nabla^2 f(x_*) \succ 0$, $\{ f(x^i) \} \to f(x_*)$ superlinearly (can be proven ...)
- Usual stuff about if $\{ x^i \}$ converges (compactness ...)
- Minor tweak: requires $m_1 \leq 1/2$ (one way to see it: $m_1 > 1/2$ "too steep", cuts away minimum e.g. when f quadratic)

Global convergence of Newton's method (cont.d)

- Even better: for some iteration onwards, $\alpha^i = 1$ always satisfy (A)

$$\begin{aligned} f(x^i + d^i) &= f(x^i) + d^i \nabla f(x^i) + \tfrac{1}{2}(d^i)[\nabla^2 f(x^i)]d^i + R(\| d^i \|) \\ &= f(x^i) + \nabla f(x^i)^T [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) \\ &\qquad -\tfrac{1}{2} \nabla f(x^i)^T [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) + R(\| d^i \|) \\ &= f(x^i) - \tfrac{1}{2} \nabla f(x^i)^T [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) + R(\| d^i \|) \end{aligned}$$

- Convergence $\Longrightarrow \| \nabla f(x^i) \| \to 0 \Longrightarrow \| d^i \| \to 0$ (**why?**)
- $R(\| d^i \|) \in O(\| \nabla f(x^i) \|^3)$ (**why?**) $\Longrightarrow$ eventually R is negligible
  $\Longrightarrow$ eventually (A) holds with $m_1 < 1/2$ (again)
- Backtracking search with $\bar{\alpha} = 1$ particularly well-suited here:
  $\exists\, k$ s.t. $\alpha^i = 1\ \forall\, i \geq k$
- No backtracking after a point $\Longrightarrow$ quadratic convergence in the end

Newton's method for general functions

- Does it only work for strongly convex f and Lipschitz ∇f? Nope
- Where did we use "d^i defined out of $\nabla^2 f(x^i)$"? Nowhere
- $d^i \leftarrow -H^i \nabla f(x^i)+$ (A) $\cap$ (W') $\Longrightarrow$ global convergence if $uI \preceq H^i \preceq LI$
- Local quadratic and global superlinear convergence obviously require $H^i = \nabla^2 f(x^i)$

Newton's method for general functions

- Does it only work for strongly convex f and Lipschitz ∇f? Nope
- Where did we use "d^i defined out of $\nabla^2 f(x^i)$"? Nowhere
- $d^i \leftarrow -H^i \nabla f(x^i)$+ (A) $\cap$ (W') $\Longrightarrow$ global convergence if $uI \preceq H^i \preceq LI$
- Local quadratic and global superlinear convergence obviously require $H^i = \nabla^2 f(x^i)$ ... or do they? Actually, $H^i \approx \nabla^2 f(x^i)$ suffices
- f not convex $\equiv \nabla^2 f \not\succ 0$ can't take $H^i = [\nabla^2 f(x^i)]^{-1}$: hack it
- Simple approach: choose "smallest" ε^i s.t. $H^i = \nabla^2 f(x^i) + \varepsilon^i I \succ 0$
- Issue: there is no such thing, any $\varepsilon^i > -\lambda^n$ works ($\lambda^n < 0$)
- Numerical issues: $0 < \lambda^n \leq$ 1e-16 $\Longrightarrow \nabla^2 f(x^i) \succ 0$ "mathematically but not computationally" (1e-16 very optimistic, 1e-12 or larger)
- Algorithmic issues: if $\lambda^n(\nabla^2 f(x^i) + \varepsilon I)$ "very small", then "axes of quadratic model very elongated" $\Longrightarrow$ "x^{i+1} very far from x^i", not the way one expects using a local model
- Simple form: $\varepsilon = \max\{\, 0\,,\, \delta - \lambda^n \,\}$ for appropriately chosen smallish δ (1e-8? 1e-4? 1e-12? hard to say in general)
- Solves $\min\{\, \| H - \nabla^2 f(x^i) \|\,,\, H \succeq \delta I \,\}$ ... "$\succeq$" constraints??

Newton's method: Hessian modifications

- The above holds for $\| \cdot \|_2$, with other norms it is $\neq$
- Solution of $\min\{ \| H - \nabla^2 f(x^i) \|_F , H \succeq \delta I \}$ is:
  - compute spectral decomposition $\nabla^2 f(x^i) = H \Lambda H^T$
  - $H^i = H \bar{\Lambda} H^T$ with $\bar{\gamma}^i = \max\{ \lambda^i , \delta \}$
- Solving problems with $\succeq$ constraints $\equiv$ computing eigenvalues/vectors
- In both cases, $\{ x^i \} \to x_*$ with $\nabla^2 f(x_*) \succeq \delta I \Longrightarrow \varepsilon^i = 0 \equiv$ $H^i = \nabla^2 f(x^i)$ eventually $\Longrightarrow$ quadratic convergence in the tail
- Superlinear convergence is also relatively easy to obtain, like in

  $$\lim_{i \to \infty} \| (H^i - \nabla^2 f(x^i)) d^i \| / \| d^i \| = 0$$

  i.e., "H^i looks like $\nabla^2 f(x^i)$ along d^i" (but don't care elsewhere)
- In both cases, $O(n^3)$; say, compute λ^n + Cholesky factorization $H^i = L^i (L^i)^T$, L^i triangular (fastest and more stable way)
- Can use nifty tricks, like modify Cholesky factorization to compute L^i before H^i (diagonal $< 0 \Longrightarrow$ modify it)
- Whatever you do, $O(n^3)$ too much for large-scale (large n)

A different approach: Trust Region (sketch)

- $\nabla^2 f(x^i) \not\succ 0 \Longrightarrow \exists$ negative curvature directions
- That's directions where f eventually decrease, aren't they interesting when minimizing f?
- Instead of raping $\nabla^2 f$ to pretend they don't exist, let's exploit them
- But how? Second-order model has no minimum?
- That's if $x \in \mathbb{R}^n$, but it does if restricted to a conpact set
- $x^{i+1} \in \operatorname{argmin}\{ \nabla f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T [\nabla^2 f(x^i)](x - x^i) : x \in \mathcal{T}^i \}$
  $\mathcal{T} =$ trust region, $\subset \mathbb{R}^n$ where "the model can be trusted"
- However, this is a constrained optimization problem
- Even worse: it is $\mathcal{NP}$-hard even for simple $\mathcal{T}$ like $\mathcal{B}_1(x^i, r)$ or $\mathcal{B}_\infty(x^i, r)$

A different approach: Trust Region (sketch)

- $\nabla^2 f(x^i) \not\succ 0 \Longrightarrow \exists$ negative curvature directions
- That's directions where f eventually decrease, aren't they interesting when minimizing f?
- Instead of raping $\nabla^2 f$ to pretend they don't exist, let's exploit them
- But how? Second-order model has no minimum?
- That's if $x \in \mathbb{R}^n$, but it does if restricted to a conpact set
- $x^{i+1} \in \operatorname{argmin}\{ \nabla f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T [\nabla^2 f(x^i)](x - x^i) : x \in \mathcal{T}^i \}$
  $\mathcal{T}$ = trust region, $\subset \mathbb{R}^n$ where "the model can be trusted"
- However, this is a constrained optimization problem
- Even worse: it is $\mathcal{NP}$-hard even for simple $\mathcal{T}$ like $\mathcal{B}_1(x^i, r)$ or $\mathcal{B}_\infty(x^i, r)$
  ... but not for $\mathcal{B}_2(x^i, r) = \{ x \in \mathbb{R}^n : \| x - x^i \|_2 \leq 1 \}$
- This is an optimization problem with quadratic constraints
- "Round balls are simpler than kinky balls"

Trust Region (sketch, cont.d)

- Of course, can use any $H^i \approx \nabla^2 f(x^i)$, not necessarily $\succ 0$
- x^{i+1} optimal $\equiv \exists\, \lambda^* \geq 0$ s.t. $[\, H^i + \lambda^* I \,] x^{i+1} = -\nabla f(x^i) \wedge H^i + \lambda^* I \succeq 0$ $\wedge\, \lambda^* (\, r - \| x^{i+1} \| \,) = 0$ (that'll be Karush-Khun-Tucker for you, sir ...)
- $\| x^{i+1} \| < r \Longrightarrow \lambda^* = 0 \Longrightarrow$ normal Newton step ($\mathcal{T}$ has no effect)
- $\lambda^* > 0 \Longrightarrow$ like in line search with $\varepsilon^i = \lambda^*$
- Plenty of smart ways to find λ^*, x^{i+1} or approximate them (just as well), typically have to do with eigenvalue/vectors computation
- Line Search $\approx$ Trust Region, except:
  - Line Search: first d^i, then α^i
  - Trust Region: first r ($\approx \alpha^i$), then d^i

  Nonetheless, countless religion wars (as in Atari vs. Intellevision, Spectrum vs. C64, PC vs. Mac, PS vs. XBox, Android vs. iPhone, ...)
- No time to develop details, so we follow the sacred path of Line Search
- In both religions, plenty of ways to use $H^i \approx \nabla^2 f(x^i)$ to reduce the cost, let's concentrate on that

Outline

Quasi-Newton methods

- Derivation of Quasi-Newton methods
- $m^i(x) = \nabla f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T H^i (x - x^i)$, $x^{i+1} = x^i + \alpha^i d^i$
- Having computed $\nabla f(x^{i+1})$, want to update model to
  $m^{i+1}(x) = \nabla f(x^{i+1})(x - x^{i+1}) + \frac{1}{2}(x - x^{i+1})^T H^{i+1}(x - x^{i+1})$
- How to choose H^{i+1}? Let's see what we would like to have:
  i) $H^{i+1} \succ 0$ ii) $\nabla m^{i+1}(x^i) = \nabla f(x^i)$ iii) $\| H^{i+1} - H^i \|$ "small"
- ii) $\equiv H^{i+1}(x^{i+1} - x^i) = \nabla f(x^{i+1}) - \nabla f(x^i)$ "secant equation" (**check**)

Exercise: figure out why the name (hint: with $n = 1$, picture f')

- Notation: $s^i = x^{i+1} - x^i = \alpha^i d^i$, $y^i = \nabla f(x^{i+1}) - \nabla f(x^i) \Longrightarrow$
  ii) $\equiv$ (S) $H^{i+1} s^i = y^i$
- i) $\Longrightarrow s^i y^i = (s^i)^T H^{i+1} s^i = \| s^i \|^2 > 0$ "curvature condition" (C)
  depends on the data, (C) does not hold $\Longrightarrow$ i) $\cap$ ii) $= \emptyset$
- (W') $\Longrightarrow$ (C): $\varphi'(\alpha^i) = \nabla f(x^{i+1}) d^i \geq m_3 \varphi'(0) = m_3 \nabla f(x^i) d^i$
  $\Longrightarrow (\nabla f(x^{i+1}) - \nabla f(x^i)) d^i \geq (m_3 - 1)\varphi'(0) > 0$

Quasi-Newton methods: DFP

- Hence $\min\{ \| H - H^i \| : (\mathrm{S}) \ , \ H \succeq 0 \}$ has a solution (again "$\succeq \ldots$")
- With proper choice of "$\| \cdot \|$", it turns out to be ($\rho^i = 1/y^i s^i > 0$)

  (DFP) $\quad H^{i+1} = (I - \rho^i y^i (s^i)^T) H^i (I - \rho^i s^i (y^i)^T) + \rho^i y^i (y^i)^T$

  Davidon-Fletcher-Powell formula
- However, what one really needs is $B^{i+1} = [H^{i+1}]^{-1}$
- Collecting terms, (DFP) $\equiv$ rank-two correction of H^i, can use

  (SMW) $\quad [A + ab^T]^{-1} = A^{-1} - A^{-1} ab^T A^{-1} / (1 - b^T A^{-1} a)$

  Sherman-Morrison-Woodbury formula for inverse of rank-one correction
- Using (SMW), update directly for B^i:

  (DFP) $\quad B^{i+1} = B^i + \rho^i s^i (s^i)^T - B^i y^i (y^i)^T B^i / (y^i)^T B^i y^i$
- $O(n^2)$ iteration cost: just matrix-vector products, no inverse
- This is learning $\nabla^2 f$ out of samples of ∇f as you go (learning2optimize)
- Quite efficient, but can do better

Quasi-Newton methods: BFGS

- Write (S) in terms of B^{i+1} rather than H^{i+1}: $s^i = B^{i+1} y^i$
- Problem becomes $\min\{ \| B - B^i \| : (\mathrm{S}) \ , \ B \succeq 0 \}$
- Everything is symmetric, just $B \leftrightarrow H$ and $s \leftrightarrow y$
- Broyden-Fletcher-Goldfarb-Shanno formulæ:

$$\text{(BFGS)} \quad H^{i+1} = H^i + \rho^i y^i (y^i)^T - H^i s^i (s^i)^T H^i \,/\, (s^i)^T H^i s^i$$

$$\text{(BFGS)} \quad B^{i+1} = (\, I - \rho^i s^i (y^i)^T \,) B^i (\, I - \rho^i y^i (s^i)^T \,) + \rho^i s^i (s^i)^T$$

$$= B^i + \rho^i [\, (\, 1 + \rho^i (y^i)^T B^i y^i \,) s^i (s^i)^T - (\, B^i y^i (s^i)^T + s^i (y^i)^T B^i \,) \,]$$

- Broyden family: $\beta H^{i+1}_{\mathrm{DFP}} + (1-\beta) H^{i+1}_{\mathrm{BFGS}}$: $\succeq 0$ if $\beta \in [0,1]$, satisfies (S), even more flexible (nice theoretical results about how to choose β ...)
- Very good compromise between iteration cost and convergence speed
- Nontrivial issue: B^1? σI, σ?. Finite difference $\approx [\, \nabla^2 f(x^1) \,]^{-1}$? ...

Exercise: work out the details: what is a "finite difference $\approx [\, \nabla^2 f(x^1) \,]^{-1}$", how you compute it, how much does this cost

"Poorman's" quasi-Newton: limited-memory BFGS

- For very large n even $O(n^2)$ is too much (folklore says $\leq O(n^{3/2})$)
- Quasi-newton generates $O(n)$ new information per iteration
- Obvious idea: only keep/use $k \ll n$ of the generated information
- Limited-memory BFGS (L-BFGS):

  (BFGS) $\quad B^{i+1} = (V^i)^T B^i V + \rho^i s^i (s^i)^T \quad$ with $V^k = I - \rho^i y^i (s^i)^T$

  just unfold the loop of last k iterations:

  (BFGS) $\quad B^{i+1} = (V^i V^{i-1} \dots V^{i-k})^T B^{i-k} (V^i V^{i-1} \dots V^{i-k}) +$
  $\rho^{i-k+1} (V^i \dots V^{i-k+1})^T s^{i-k+1} (s^{i-k+1})^T (V^i \dots V^{i-k+1}) +$
  $\rho^{i-k+2} (V^i \dots V^{i-k+2})^T s^{i-k+2} (s^{i-k+2})^T (V^i \dots V^{i-k+2}) +$
  $\dots + \rho^i s^i (s^i)^T$
- Memory/time cost per iteration is $O(kn)$, convergence worsens as $k \searrow$
- k small $\approx$ gradient, k large $\approx$ Newton, can find good trade-off
- Funny tidbit: can choose B^{i-k} arbitrarily anew at each i

  (of course need be sparse, e.g. $B^{i-k} = \gamma^i I$ with $\gamma^i = s^i y^{i-1} / \| y^{i-1} \|^2$)

Exercise: work out the details of how to compute d^i in $O(kn)$

Outline

Conjugate gradient method: quadratic functions

- Gradient method + exact LS $\Longrightarrow \langle\, d^{i+1}\,,\, d^{i}\,\rangle = 0 \equiv d^{i+1} \perp d^{i}$
- Good for one step, but not more: $d^{i+1} \not\perp d^{i-1} \Longrightarrow$ zig-zags
- Would be nice if $d^1 \perp d^2 \perp \ldots d^i = 0 \Longrightarrow$ no zig-zags
- Actually ($\approx$) possible with quadratic f with simple recurrence formula
- Actual form is $(d^i)^T Q\, d^{i-1} = 0$, i.e., d^i and d^{i-1} conjugate w.r.t. Q (C)
- Can't make it with $d^i = -\nabla f(x^i)$, have to modify it
- Very simple modification: $-\nabla f(x^i)$ is deflected using d^{i-1}, i.e.
  $$d^i = -\nabla f(x^i) + \beta^i\, d^{i-1} \qquad \text{(with } d^0 = 0\text{)}$$
- β^i giving (C) immediate: $\beta^i = (\,\nabla f(x^i)^T Q\, d^{i-1}\,) \,/\, (\,(d^{i-1})^T Q\, d^{i-1}\,)$
- Optimal step also trivial: $\alpha^i = -(\,\nabla f(x^i)^T\, d^i\,) \,/\, (\,(d^i)^T Q\, d^i\,)$
- Alternative formulæ can be devised, e.g. $\beta^i = \|\,\nabla f(x^i)\,\|^2 \,/\, \|\,\nabla f(x^{i-1})\,\|^2$ (useful later), $\alpha^i = \|\,\nabla f(x^i)\,\|^2 \,/\, (\,(d^i)^T Q\, d^i\,)$

Conjugate gradient method: quadratic functions (cont.d)

- Basic version of the method (can be streamlined somewhat)

```
procedure x = CGQ ( Q , q , x , ε )  {
  d⁻ ← 0;
  while( ‖∇f(x)‖ > ε ) do  {
    if( d⁻ = 0 ) then d ← −∇f(x);
      else { β = ( ∇f(x)ᵀ Q d⁻ ) / ( (d⁻)ᵀ Q d⁻ ); d ← −∇f(x) + β d⁻; }
    α ← ( ∇f(x)ᵀ d )/( dᵀ Q d ); x ← x + α d; d⁻ ← d;
  }
}
```

- $\nabla f(x) = 0 \equiv Qx = -q$ in at most n iterations (exact arithmetic)
- Proof uses exact line search and $d^1 = -\nabla f(x^1)$ (won't work otherwise)
- Can take much less than n iterations (clustered eigenvalues . . .), especially if properly preconditioned
- Mostly useful for solving linear systems, so details @Federico

Conjugate gradient method: nonlinear functions

- Basic Fletcher-Reeves version with Armijo-Wolfe line search

```
procedure x = CGQ ( Q , q , x , ε )  {
  d⁻ ← 0; ∇f⁻ = 0;
  while( ‖∇f(x)‖ > ε ) do  {
    if( d⁻ = 0 ) then d ← −∇f(x);
      else { β = ‖∇f(x^i)‖² / ‖∇f⁻‖²; d ← −∇f(x) + βd⁻; }
    α ← AWLS( f(x + αd) ); x ← x + αd; d⁻ ← d; ∇f⁻ ← ∇f(x);
  }
}
```

- f quadratic + exact line search $\Longrightarrow$ quadratic CG
- Many $\neq \beta$-formulæ, all $\equiv$ for quadratic f, not so here
  1. Polak-Ribière: $\beta^i = [\nabla f(x^i)^T(\nabla f(x^i) - \nabla f(x^{i-1}))] / \|\nabla f(x^{i-1})\|^2$
  2. Hestenes-Stiefel: $\beta^i = [\nabla f(x^i)^T(\nabla f(x^i) - \nabla f(x^{i-1}))] / [(\nabla f(x^i) - \nabla f(x^{i-1}))^T d^{i-1}]$
  3. Dai-Yuan: $\beta^i = \|\nabla f(x^i)\|^2 / [(\nabla f(x^i) - \nabla f(x^{i-1}))^T d^{i-1}]$
  4. ...
- Convergence mostly uses standard descent arguments with quirks

Conjugate gradient method: convergence and efficiency

- Convergence not entirely trivial, depends a lot on β-formula
- F-R requires $m_1 < m_2 < 1/2$ for (A) $\cap$ (W') to work
- (A) $\cap$ (W') $\not\Longrightarrow$ d^i of P-R is of descent, unless $\beta^i_{PR} = \max\{\, \beta^i \,,\, 0 \,\}$
- The above is a restart: from time to time, take "plain" $-\nabla f$
- Turns out restarts are a good idea, especially for F-R: $\| \nabla f(x^i) \| \ll \| d^i \|$
  $\Longleftrightarrow$ $\cos(\, \theta^i \,) \approx 0 \equiv \nabla f(x^i) \approx\perp d^i \Longrightarrow x^{i+1} \approx x^i \Longrightarrow \cos(\, \theta^{i+1} \,) \approx 0$
  $\Longrightarrow$ one bad step leads to many bad steps, restarting cures this
- Typical restart after n steps, not very nice when n is large (or small)
- P-R does not need that, but does not converge for some f
- Variants surprisingly $\neq$ in practice; P-R/D-Y often better but varies a lot
- Efficiency n-step quadratic: $\| x^{i+n} - x_* \| \leq R \| x^i - x_* \|^2$
  n CG steps $\approx$ 1 Newton step (makes sense, "exactly solve for $\nabla^2 f(x)$")
- Not very nice when n is large
- Interesting relationships with quasi-Newton methods, hybrid versions
- All in all: powerful approach, but not easy to manage

Outline

"Poorman's deflection" I: Heavy Ball Gradient

- Basic idea: use previous direction while computing current one
- Simple form "just do it": $x^{i+1} \leftarrow x^i - \alpha^i \nabla f(x^i) + \beta^i (\, x^i - x^{i-1} \,)$
- x^i "heavy", $\nabla f(x^i)$ "force" steering the trajectory ($\beta =$ "momentum")
- Not a descent algorithm, $f(x^i)$ may $\nearrow \searrow \nearrow \searrow \Longrightarrow$ specific analysis
- Clear results in strongly convex case $(\lambda^n = u)\, I \preceq \nabla f^2(x) \preceq (\lambda^1 = L)\, I$
- $\lambda^n > 0 \Longrightarrow$ can estimate best possible values of α^i and β^i (constant):

$$\alpha = 4 \,/\, \left(\sqrt{\lambda^1} + \sqrt{\lambda^n} \right)^2 \quad , \quad \beta = \max \left\{ \left| 1 - \sqrt{\alpha \lambda^n} \right| , \left| 1 - \sqrt{\alpha \lambda^1} \right| \right\}^2$$

- With these choices

$$\| \, x^{i+1} - x_* \, \| \leq \left(\frac{\sqrt{\lambda^1} - \sqrt{\lambda^n}}{\sqrt{\lambda^1} + \sqrt{\lambda^1}} \right) \| \, x^i - x_* \, \|$$

  a lot of algebra to minimize on α and β ... and a bit of black magic
- Again, $\{ \, x^i \, \} \rightarrow x_*$ (no surprise: strongly convex $\Longrightarrow S(f, v)$ compact)
- $\sqrt{\lambda^1}$ much smaller than $\lambda^1 \Longrightarrow$ this R much better than gradient's
- $\lambda^1 = 1000 \lambda^n \Longrightarrow$ this $R \approx 0.938$, gradient $R \approx 0.996$
- May seem small, but $0.996^{100} = 0.6698$, $0.938^{100} = 0.0016$!

Heavy Ball Gradient (cont'd)

- Different convergence rate shows in practice
- Issue: λ^1 and λ^n unknown, so have to find right α^i and β^i
- Convergence with non-strongly-convex f a lot murkier
- Even more so with non convex f
- Very recent result: convergence in the nonconvex case for
$$\beta \in (0, 1] \quad , \quad \alpha \in (0, 2(1-\beta)/\lambda^1)$$
- Good news: free to choose β
- Bad news: $2/L$ already rather small in practice, $\searrow 0$ as $\beta \to 1$
- And, again, λ^1 unknown
- Can work well in practice, but nontrivial

"Poorman's deflection" II: Accelerated Gradient

procedure $y = ACCG\,(\,f\,,\,\nabla f\,,\,x\,,\,\varepsilon\,)\;\;\{$
$x_- \leftarrow x;\ \gamma \leftarrow 1;$
do $\{$
$\gamma_- \leftarrow \gamma;\ \gamma \leftarrow (\,\sqrt{4\gamma^2 + \gamma^4} - \gamma^2\,)/2;\ \beta \leftarrow \gamma(\,1\,/\,\gamma_- - 1\,);$
$y \leftarrow x + \beta(\,x - x_-\,);\ g \leftarrow \nabla f(y);\ x_- \leftarrow x;\ x \leftarrow y - (\,1\,/\,L\,)g;$
$\}$ **while**($\|g\| > \varepsilon$);
$\}$

"Poorman's deflection" II: Accelerated Gradient

procedure $y = ACCG\,(\,f\,,\,\nabla f\,,\,x\,,\,\varepsilon\,)\;\;\{$
$\quad x_- \leftarrow x;\; \gamma \leftarrow 1;$
$\quad$ **do** $\;\{$
$\quad \gamma_- \leftarrow \gamma;\; \gamma \leftarrow (\,\sqrt{4\gamma^2 + \gamma^4} - \gamma^2\,)/2;\; \beta \leftarrow \gamma(\,1\,/\,\gamma_- \,-\, 1\,);$
$\quad y \leftarrow x + \beta(\,x \,-\, x_-\,);\; g \leftarrow \nabla f(y);\; x_- \leftarrow x;\; x \leftarrow y - (\,1\,/\,L\,)g;$
$\quad \}$ **while**$(\;\|g\| > \varepsilon\;);$
$\}$

- . . . er . . . what?? That's a lot of black magic!

"Poorman's deflection" II: Accelerated Gradient

procedure $y = ACCG\,(\,f\,,\,\nabla f\,,\,x\,,\,\varepsilon\,)\;\;\{$
$x_- \leftarrow x;\ \gamma \leftarrow 1;$
do $\{$
$\gamma_- \leftarrow \gamma;\ \gamma \leftarrow (\,\sqrt{4\gamma^2 + \gamma^4} - \gamma^2\,)/2;\ \beta \leftarrow \gamma(\,1\,/\,\gamma_- - 1\,);$
$y \leftarrow x + \beta(\,x - x_-\,);\ g \leftarrow \nabla f(y);\ x_- \leftarrow x;\ x \leftarrow y - (\,1\,/\,L\,)g;$
$\}$ **while**($\|g\| > \varepsilon$);
$\}$

- . . . er . . . what?? That's a lot of black magic!
- Like ballstep, but ∇f computed after momentum but before descent
- Where does all these weird formulæ come from?
- Actually, there is a method in your madness (if f is convex)
- Turns out, this is not at all a gradient-like method, but something (almost) entirely $\neq$
- Let's shed some light in the darkness of black magic

Analysis of Accelerated Gradient (sketch)

- Behind the scenes there is a sequence $\{ \psi^i \}$ of models of f, starting from
$$\psi^0(x) = f(x^1) + L\| x - x^1 \|^2/2 \geq f(x)$$
- At each iteration the model includes new information:
$$\psi^i(x) = (\, 1 \,-\, \gamma_{i-1} \,)\psi^{i-1}(x) + \gamma_{i-1}(\, f(y^i) + \nabla f(y^i)(\, x \,-\, y^i \,)\,)$$
- The model incrementally gets a better and better upper estimator:
$$\psi^i(x) \leq (\, 1 - \sigma^i \,) f(x) + \sigma^i \psi^0(x) \quad \text{with} \quad \sigma^i = O(\, 1 \,/\, i^2 \,) \searrow 0$$
- x^i is carefully constructed so that $f(x^i) \leq \min\{\, \psi^i(x) \,:\, x \in \mathbb{R}^n \,\}$
- This implies $|\, f(x^i) - f_* \,| \leq \sigma^i (\, f(x^i) - f_* \,) \searrow 0$ as $O(\, 1 \,/\, i^2 \,)$
- All choices carefully tailored to make everything work
- Convexity baked in the analysis from the roots
- Heavily model-based approach, where the model is "order $1 + \varepsilon$"
- Interesting fact: no first-order algorithm can do better than
$$|\, f(x^i) - f_* \,| = 3L\| x^1 - x_* \|^2 \,/\, 32(\, i + 1 \,)^2 \implies \text{ACCG optimal}$$
(another bit of black magic . . .)
- Counter-example is a simple quadratic function (not strongly convex)

Accelerated Gradient in practice

- Issue: L unknown $\Longrightarrow$ use line search
- Backtraking gives $\alpha^i \geq 1/2L$, hence things still work
- For bound to work $\alpha^i \leq \alpha^{i-1}$: easy, just start backtracking from α^i
- Algorithm non-monotone, like heavy ball, but can be made monotone

Exercise: discuss how to make any non-monotone algorithm monotone (doh!)

- No guaranteed linear convergence for $\nabla^2 f \succeq uI$, but can be modified to
- Universal gradient method: automatically adapts to f to always achieve best complexity
- Practical behaviour: consistently slowish
- Carefully crafted to attain a given convergence speed, typically gets what it is constructed for
- Typical of algorithms constructed to optimize worst-case behaviour
- Speed reasonable (in its class), but less guarantees in worst-case may mean more speed on average (but also catastrophic failures)

Outline

Wrap up

- You can go (much) faster than gradient
- Thanks goodness, because gradient is very slow already with $\nabla^2 f \succeq uI$, all the more so otherwise
- But there is only so much you can do with first-order methods
- Convergence at best linear ($\nabla^2 f \succeq uI$), possibly much worse
- Choice of model most often the deciding factor
- Second-order methods have vastly better convergence ($\nearrow$ quadratic)
- But $\nabla^2 f$ has to $\exists$, be continuous, and you have to use it
- Although, you can use $\nabla^2 f$ without ever computing it
- First-and-a-half-order methods provide interesting trade-offs
- A lot of details need be considered, numerical aspects nontrivial
- Your mileage may vary