Scaled gradient methods
Newton and quasi-Newton methods

Pontus Giselsson

Outline

- **Scaled gradient method**
- Backtracking
- Newton's method
- Quasi-Newton methods
- A numerical example

Scaled gradient method

- We consider problems

$$\underset{x}{\text{minimize}}\, f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable

- We consider scaled gradient methods

$$x_{k+1} = x_k - \gamma_k H_k^{-1} \nabla f(x_k)$$

where H_k is a symmetric positive definite scaling matrix

- Have seen that scaling can improve convergence

Selecting H_k

- The scaled gradient method is

$$
\begin{aligned}
x_{k+1} &= \underset{y}{\operatorname{argmin}}(f(x_k) + \nabla f(x_k)^T(y - x) + \tfrac{1}{2\gamma_k}\|y - x_k\|_{H_k}^2) \\
&= \underset{y}{\operatorname{argmin}}(f(x_k) + \tfrac{1}{2\gamma_k}\|y - (x_k - \gamma_k H_k^{-1}\nabla f(x_k))\|_{H_k}^2) \\
&= x_k - \gamma_k H_k^{-1}\nabla f(x_k)
\end{aligned}
$$

- H_k should capture (some) second-order (Hessian) information
- Examples:
  - $H_k = I$ is identity matrix (gives proximal gradient method)
  - $H_k = \mathbf{diag}(h)$ is fixed diagonal matrix with diagonal h
  - $H_k = H$ is fixed full or structured matrix
  - $H_k = \nabla^2 f(x_k)$ is true Hessian (Newton method)
  - H_k is from (limited memory) quasi-Newton
- More on this later, we first show convergence

Assumptions

- Similar assumptions as for proximal gradient method:
  - (i) $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable (not necessarily convex)
  - (ii) $\forall x_k$, x_{k+1}, it exists $\beta_k \in [\eta, \eta^{-1}]$, $\rho I \preceq H_k \preceq \rho^{-1} I$, $\eta, \rho \in (0, 1)$:

    $$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \tfrac{\beta_k}{2} \|x_k - x_{k+1}\|_{H_k}^2$$

    which means f is "locally β_k smooth w.r.t. $\|\cdot\|_{H_k}$
  - (iii) A minimizer exists (and $p^\star = \min_x (f(x) + g(x))$ is optimal value)
  - (iv) Algorithm parameters $\gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon]$, where $\epsilon > 0$
- Assumption on f satisfied with $\beta_k H_k = \beta I$ if f β-smooth

Convergence

Using

(a) Upper bound assumption on f, i.e., Assumption (ii)

(b) Algorithm update: $x_{k+1} - x_k = \gamma_k H_k^{-1} \nabla f(x_k)$

gives

$$
\begin{aligned}
f(x_{k+1}) &\overset{(a)}{\leq} f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \tfrac{\beta_k}{2} \|x_{k+1} - x_k\|_{H_k}^2 \\
&\overset{(b)}{\leq} f(x_k) - \gamma_k \nabla f(x_k)^T H_k^{-1} \nabla f(x_k) + \tfrac{\beta_k \gamma_k^2}{2} \|H_k^{-1} \nabla f(x_k)\|_{H_k}^2 \\
&= f(x_k) - \gamma_k (1 - \tfrac{\beta_k \gamma_k}{2}) \|\nabla f(x_k)\|_{H_k^{-1}}^2 \\
&\leq f(x_k) - \delta \|\nabla f(x_k)\|_{H_k^{-1}}^2
\end{aligned}
$$

where we used: $\gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon]$ implies $\gamma_k (1 - \frac{\beta_k \gamma_k}{2}) \geq \delta > 0$

Lyapunov inequality

- Subtract $p^\star$ from both sides to get Lyapunov inequality

$$\underbrace{f(x_{k+1}) - p^\star}_{V_{k+1}} \leq \underbrace{f(x_k) - p^\star}_{V_k} - \underbrace{\delta\|\nabla f(x_k)\|^2_{H_k^{-1}}}_{R_k}$$

- Consequences:
  - Function values converge (not necessarily to $p^\star$)
  - R_k is summable and, since $\delta > 0$, we have $\|\nabla f(x_k)\|_{H_k^{-1}} \to 0$
  - R_k summable also implies

$$\min_{i\in\{0,\ldots,k\}} \|\nabla f(x_i)\|^2_{H_k^{-1}} \leq \frac{f(x_0) - p^\star}{\delta(k+1)}$$

- Comment: The above analysis can also include $\mathrm{prox}^{H_k}_{\gamma_k g}$ term

Outline

- Scaled gradient method
- **Backtracking**
- Newton's method
- Quasi-Newton methods
- A numerical example

Selecting algorithm parameters

- How to select β_k, γ_k and H_k?
- Start with β_k and γ_k, given H_k

Choose β_k and γ_k

- Convergence based on assumption that β_k known that satisfies

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \tfrac{\beta_k}{2} \|x_k - x_{k+1}\|_{H_k}^2$$

  call this *descent condition* (DC)

- This descent condition generalizes the previous where $H_k = I$
- If $H_k = H$ and f β_H-smooth w.r.t. $\|\cdot\|_H$; $\beta_k = \beta_H$ works since

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \tfrac{\beta_H}{2} \|x - y\|_H^2$$

  for all x, y

Choose β_k and γ_k – Backtracking

- Same backtracking as before, but with generalized DC
- Backtracking, choose $\kappa > 1$, $\beta_{k,0} \in [\eta, \eta^{-1}]$, let $l_k = 0$, and loop:

1. choose $\gamma_k \in [\epsilon, \frac{2}{\beta_{k,l}} - \epsilon]$
2. compute $x_{k+1} = x_k - \gamma_k H_k^{-1} \nabla f(x_k)$
3. **if** descent condition (DC) satisfied

    set $k \leftarrow k + 1$ `// increment algorithm counter`
    set $\bar{l}_k \leftarrow l_k$ `// store final backtrack counter`
    break backtrack loop

   else

    set $\beta_{k,l_k+1} \leftarrow \kappa \beta_{k,l_k}$ `// increase backtrack parameter`
    set $l_k \leftarrow l_k + 1$ `// increment backtrack counter`

   end

- Note that larger β_{k,l_k} gives smaller step-length upper bound
- Initialization of $\beta_{k,0}$ depends on choice of H_k
- Works also with scaled proximal steps with $\mathrm{prox}_{\gamma_k g}^{H_k}$

Backtracking – Convergence

- For convergence, need to verify that (DC):

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta_k}{2}\|x_k - x_{k+1}\|_{H_k}^2$$

  will hold within finite number of backtracking steps
- Assume and recall that
  - $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth
  - $\beta_k \in [\eta, \eta^{-1}]$, $\rho I \preceq H_k \preceq \rho^{-1} I$, $\eta, \rho \in (0,1)$:

  which gives

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta}{2}\|x_k - x_{k+1}\|_2^2 \\ &\leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta}{2\rho}\|x_k - x_{k+1}\|_{H_k}^2 \end{aligned}$$

  i.e, (DC) satisfied whenever $\beta_k \geq \frac{\beta}{\rho}$ (maybe before)

Outline

- Scaled gradient method
- Backtracking
- **Newton's method**
- Quasi-Newton methods
- A numerical example

Newton's method

- Newton's method given by iteration ($H_k = \nabla^2 f(x_k)$)

$$x_{k+1} = x_k - \gamma_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable
- Propeties:
  - Sometimes quadratic local convergence if $\gamma_k = 1$
  - Unit step-size $\gamma_k = 1$ may diverge far from solution
  - Need backtracking to converge globally
- Note: $\nabla^2 f(x_k)$ must be positive definite, i.e., $\nabla^2 f(x) \succ 0$:
  - always true if problem strictly convex
  - if not, add ϵI with $\epsilon > 0$ such that $H_k = \nabla^2 f(x_k) + \epsilon I \succ 0$ (no local quadratic convergence, but still very fast)

Assumptions

- Assumptions
  - (i) $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable
  - (ii) f is σ-strongly convex and β-smooth
  - (iii) $\nabla^2 f$ is L-Lipschitz continuous
  - (iv) A minimizer exists (and $p^\star = \min_x(f(x) + g(x))$ is optimal value)
  - (v) Algorithm parameters γ_k, will be chosen from backtracking
- Assumption (iii) implies that

$$f(y) \le f(x) + \nabla f(x)^T(y - x) + \tfrac{1}{2}\|x - y\|^2_{\nabla^2 f(x)} + \tfrac{L}{6}\|x - y\|_2^3$$

for all x, y (note similarity to β-smoothness, ∇f is β-Lipschitz)

Newton method analysis

Will show:

- An example with divergence if $\gamma_k = 1$
- Quadratic convergence with $\gamma_k = 1$ close to solution
- Backtracking condition will eventually accept $\gamma_k = 1$

Newton method divergence – Example

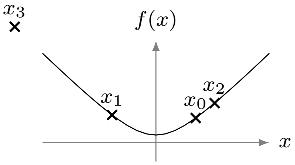
- Consider the smooth function $f(x) = \sqrt{1+x^2}$
- It is strictly convex, 1-smooth, and $\nabla^2 f$ is 1-Lipschitz
- Gradient method with $\gamma_k = 1$ works
- The gradient and second derivative satisfy

$$\nabla f(x) = \frac{x}{\sqrt{1+x^2}} \qquad \nabla^2 f(x) = \frac{1}{(1+x^2)^{3/2}}$$

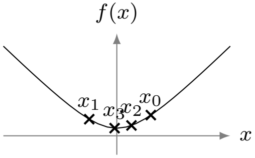
- The Newton update with $\gamma_k = 1$ becomes

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) = x_k - x_k(1+x_k^2) = -x_k^3 = -x_k(x_k)^2$$

which diverges if $|x_0| > 1$ and converges if $|x_0| < 1$



$x_0 = 1.05$



$x_0 = 0.9$

Quadratic convergence (1/2)

- We will show that $\|x_{k+1} - x^\star\|_2 \leq \frac{L}{2\sigma}\|x_k - x^\star\|_2^2$
- Using
  (a) that $\nabla f(x^\star) = 0$
  (b) that

$$(\nabla f(x^\star) - \nabla f(x_k)) = \int_0^1 (\nabla^2 f(x_k + t(x^\star - x_k))(x^\star - x_k))dt$$

  (c) and that $\int_0^1 a dt = a$ to conclude

$$x_k - x^\star = \nabla^2 f(x_k)^{-1} \int_0^1 \nabla^2 f(x_k)(x_k - x^\star)dt$$

  gives

$$\begin{aligned} x_{k+1} - x^\star &= x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) - x^\star \\ &= x_k - x^\star + \nabla^2 f(x_k)^{-1}(\nabla f(x^\star) - \nabla f(x_k)) \\ &= x_k - x^\star + \nabla^2 f(x_k)^{-1} \int_0^1 (\nabla^2 f(x_k + t(x^\star - x_k))(x^\star - x_k))dt \\ &= \nabla^2 f(x_k)^{-1} \int_0^1 (\nabla^2 f(x_k + t(x^\star - x_k) - \nabla^2 f(x_k))(x^\star - x_k))dt \end{aligned}$$

Quadratic convergence (2/2)

We continue by taking the norm of both sides of the equality

$$
\begin{aligned}
&\|x_{k+1} - x^\star\|_2 \\
&= \left\| \nabla^2 f(x_k)^{-1} \int_0^1 (\nabla^2 f(x_k + t(x^\star - x_k) - \nabla^2 f(x_k))(x^\star - x_k))dt \right\|_2 \\
&\leq \|\nabla^2 f(x_k)^{-1}\|_2 \left\| \int_0^1 (\nabla^2 f(x_k + t(x^\star - x_k) - \nabla^2 f(x_k))(x^\star - x_k))dt \right\|_2 \\
&\leq \tfrac{1}{\sigma} \int_0^1 \|\nabla^2 f(x_k + t(x^\star - x_k) - \nabla^2 f(x_k)\|_2 \|x^\star - x_k\|_2 dt \\
&\leq \tfrac{L}{\sigma} \int_0^1 t\|x^\star - x_k\|_2^2 dt \\
&= \tfrac{L}{2\sigma} \|x^\star - x_k\|_2^2
\end{aligned}
$$

where we have used

- Cauchy-Schwarz inequality twice
- that $\nabla^2 f$ is L-Lipschitz continuous
- that $\int_0^1 t dt = 1/2$

Local convergence

- We have shown that $\|x_{k+1} - x^\star\|_2 \leq \frac{L}{2\sigma}\|x_k - x^\star\|_2^2$
- Why is this only local convergence? Assume, e.g.,

$$\|x_k - x^\star\|_2 = 2 \qquad \text{and} \qquad \frac{L}{2\sigma} = 2$$

  then $\|x_{k+1} - x^\star\|_2 \leq 8$, and we cannot conclude convergence
- If $\|x_k - x^\star\|_2 \leq \frac{2\sigma}{L}(\frac{1}{2})^{2^k}$, we have R-quadratic convergence:

$$\|x_{k+1} - x^\star\|_2 \leq \frac{L}{2\sigma}\left(\frac{2\sigma}{L}\left(\frac{1}{2}\right)^{2^k}\right)^2 = \frac{2\sigma}{L}\left(\frac{1}{2}\right)^{2^{k+1}}$$

  with rate $\frac{1}{2}$, and we need $\|x_0 - x^\star\|_2 \leq \frac{2\sigma}{L}(\frac{1}{2})^{2^0}$ to start induction
- If we cannot start close enough, we need backtracking
- (Much more sophisticated analysis of Newton's method exists)

Backtracking

- We let
  - the initial backtracking parameter for every k satisfy $\beta_{k,0} \in (1,2)$
  - $\bar{l}_k$ be the final backtrack iteration with accepted $\beta_{k,\bar{l}_k}$
  - and set $\gamma_k = \beta_{k,0}/\beta_{k,\bar{l}_k} = \frac{1}{\kappa \bar{l}_k}$, where κ is backtrack increment

  with consequence that $\gamma_k = 1$ if accepted in first step, $\hat{l}_k = 0$
- The descent condition is in backtracking iteration l_k, if accepted:

$$
\begin{aligned}
f(x_{k+1}) &\le f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta_{k,l_k}}{2}\|x_{k+1} - x_k\|^2_{\nabla^2 f(x_k)} \\
&= f(x_k) - \gamma_k(1 - \tfrac{\gamma_k \beta_{k,l_k}}{2})\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \\
&= f(x_k) - \gamma_k(1 - \tfrac{\beta_{k,0}}{2})\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \\
&= f(x_k) - \gamma_k \alpha \|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}}
\end{aligned}
$$

  where we have defined $\alpha \in (0, 0.5) = 1 - \frac{\beta_{k,0}}{2}$
- We use this and instead backtrack directly on $\gamma_{k,l_k} = \frac{1}{\kappa^{l_k}}$

Unit step-size

- We will show that $\gamma_k = 1$ is eventually accepted, so we get

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

- By L-Lipschitz continuity of $\nabla^2 f$ we conclude for $\gamma_k = 1$:

$$\begin{aligned}
&f(x_{k+1}) \\
&\leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \tfrac{1}{2}\|x_k - x_{k+1}\|^2_{\nabla^2 f(x_k)} + \tfrac{L}{6}\|x_k - x_{k+1}\|_2^3 \\
&\leq f(x_k) - \gamma_k(1 - \tfrac{\gamma_k}{2})\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} + \tfrac{L}{6}\|x_k - x_{k+1}\|_2^3 \\
&\leq f(x_k) - \tfrac{1}{2}\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} + \tfrac{L}{6}\|\nabla^2 f(x_k)^{-1} \nabla f(x_k)\|_2^3 \\
&\leq f(x_k) - \tfrac{1}{2}\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} + \tfrac{L}{6\sigma^{3/2}}\|\nabla f(x_k)\|^3_{\nabla^2 f(x_k)^{-1}}
\end{aligned}$$

where we used $\nabla^2 f(x_k) \leq \frac{1}{\sigma} I$ due to σ-strong convexity of f

Unit step-size

- Now, assume that the gradient condition (GC)

$$\|\nabla f(x_k)\|_{\nabla^2 f(x_k)^{-1}} \leq \frac{6\sigma^{3/2}}{L}(\tfrac{1}{2} - \alpha)$$

holds, then we can continue the inequality as

$$\begin{aligned} f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} + \tfrac{L}{6\sigma^{3/2}}\|\nabla f(x_k)\|^3_{\nabla^2 f(x_k)^{-1}} \\ f(x_k) - \tfrac{1}{2}\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} + (\tfrac{1}{2} - \alpha)\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \\ \leq f(x_k) - \alpha\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \end{aligned}$$

this guarantees that backtracking condition holds if (GC) holds

- Backtracking analysis implies

$$\|\nabla f(x_k)\|_{\nabla^2 f(x_k)^{-1}} \to 0$$

as $k \to \infty$, so (GC) will eventually be satisfied

Outline

- Scaled gradient method
- Backtracking
- Newton's method
- **Quasi-Newton methods**
- A numerical example

Quasi-Newton methods

- Mimic Newton's method but with less computational effort
- Approximate Hessian by $H_k \approx \nabla^2 f(x_k)$ to get

$$x_{k+1} = x_k - \gamma_k H_k^{-1} \nabla f(x_k)$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable
- Select γ_k using backtracking (as in Newton's method)
- Many schemes for finding H_k, will cover BFGS[1]

[1] BFGS: Broyden-Fletcher-Goldfarb-Shanno

Secant condition

- Consider quadratic approximation of the function f

$$\hat{f}_{x_k}(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \tfrac{1}{2}\|x_k - x\|_{H_k}^2$$

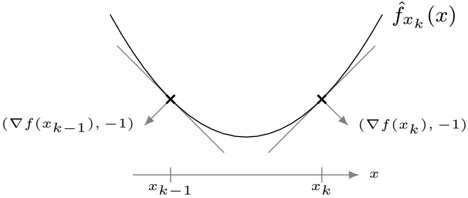
- Gradients coincide at x_k: $\nabla \hat{f}_{x_k}(x_k) = \nabla f(x_k)$
- Secant condition: Let H_k be such that

$$\nabla \hat{f}_{x_k}(x_{k-1}) = \nabla f(x_{k-1}),$$

which is satisfied when *secant condition* holds:

$$H_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1})$$

Proof: differentiate $\hat{f}_{x_k}$ (w.r.t x) and evaluate at x_{k-1}



Quasi-Newton update

- Define $s_k = x_k - x_{k-1}$ and $y_k = \nabla f(x_k) - \nabla f(x_{k-1})$, then

$$H_k s_k = y_k$$

  is secant condition
- Quasi-Newton: select H_k such that secant condition satisfied
  - H_k contains
    - n^2 variables in general case
    - $n(n+1)/2$ variables if H_k is also enforced to be symmetric
  - secant condition contains only n constraints $\Rightarrow$ underdetermined
  - Select H_k "close" to H_{k-1} subject to,
    - secant condition holds
    - possible symmetry enforcing constraint $H_k = H_k^T$

$$\begin{array}{lll} \underset{H_k}{\text{minimize}} & D(H_k, H_{k-1}) & \\ \text{subject to} & H_k s_k = y_k & \texttt{// secant condition} \\ & H_k = H_k^T & \texttt{// symmetry constraint} \end{array}$$

    where D measures distance between H_k and H_{k-1}
  - Often initialized as $H_0 = I$

Different choices of D

- A method called Broyden method is obtained by
  - $D(H_k, H_{k-1}) = \|H_k - H_{k-1}\|_F^2$
  - without symmetry constraint

  where
  - H_k not necessarily symmetric and positive definite
- A method called BFGS is obtained by
  - $D(H_k, H_{k-1}) = \mathrm{tr}(H_{k-1}^{-1} H_k) - \log\det(H_{k-1}^{-1} H_k) - n$
  - with symmetry constraint

  where
  - Cost called *relative entropy*
  - H_k is symmetric and positive definite (under some assumptions)
- BFGS is preferred over Broyden for smooth minimization

The BFGS Hessian inverse update formula

- Solving BFGS problem gives Hessian inverse $H_k^{-1} = B_k$ update:

$$B_k = (I - \frac{s_k y_k^T}{y_k^T s_k}) B_{k-1} (I - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k}$$

- Using inverse B_k is preferrable, since the algorithm becomes

$$x_{k+1} = x_k - \gamma_k B_k \nabla f(x_k)$$

  and the matrix inversion is avoided

- Cheaper than Newton's method, but requires storing $B_k \in \mathbb{R}^{n \times n}$

Evaluating direction

Let ($B_+ = B_k$, $B = B_{k-1}$, $s = s_k$, $y = y_k$), then B_+g satisfies

$$B_+g = (I - s\,\underbrace{\frac{y^T}{y^T s})\,\underbrace{B\,\underbrace{(g - y\,\underbrace{\frac{s^T g}{y^T s}}_{\alpha})}_{q}}_{p}}_{\beta} + s\,\underbrace{\frac{s^T g}{y^T s}}_{\alpha}$$

$$= p - s\beta + s\alpha = p + s(\alpha - \beta)$$

where

$$\alpha = \frac{s^T g}{y^T s} \in \mathbb{R} \qquad q = g - y\alpha \in \mathbb{R}^n \qquad p = Bq \in \mathbb{R}^n \qquad \beta = \frac{y^T p}{y^T s} \in \mathbb{R}$$

Implicit form BFGS

- Instead of storing B_k, we store all s_l and y_l for $l = \{1, \ldots, k\}$
- Recursively use previous update k times to get:
  1. Let $q = \nabla f(x_k)$
  2. For $l = k, \ldots, 1$ do
     (a) Compute $\alpha_l = \frac{s_l^T q}{y_l^T s_l}$
     (b) Update $q = q - \alpha_l y_l$
  3. Let $p = B_0 q$
  4. For $l = 1, \ldots, k$ do
     (a) Let $\beta_l = \frac{y_l^T p}{y_l^T s_l}$
     (b) Update $p = p + (\alpha_l - \beta_l) s_l$

  where final $p = B_k \nabla f(x_k)$
- Memory requirement: $2nk$, grows with iteration k
- Inefficient implementation for BFGS, but used for LBFGS

LBFGS – Limited memory BFGS

- LBFGS is implicit BFGS but look only m step back in history
- Algorithm cuts loops in two-loop procedure to be of length m
  1. Let $q = \nabla f(x_k)$
  2. For $l = k, \ldots, k - m + 1$ do
     (a) Compute $\alpha_l = \frac{s_l^T q}{y_l^T s_l}$
     (b) Update $q = q - \alpha_l y_l$
  3. Let $p = B_k^0 q$
  4. For $l = k - m + 1, \ldots, k$ do
     (a) Let $\beta_l = \frac{y_l^T p}{y_l^T s_l}$
     (b) Update $p = p + (\alpha_l - \beta_l) s_l$

  where final p is direction: $x_{k+1} = x_k - \gamma_k p$
- Common initialization: $B_k^0 = \lambda_k I$ for some $\lambda_k > 0$
- Often very small $m \in \{3, \ldots, 10\}$ performs very well
- Memory requirement: $2nm$ (compared to n^2 for BFGS)

Outline

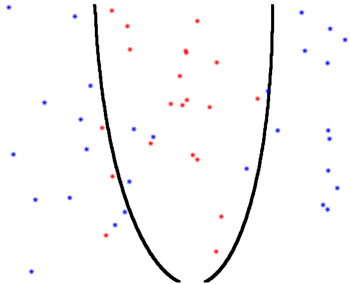
- Scaled gradient method
- Backtracking
- Newton's method
- Quasi-Newton methods
- **A numerical example**

Example – Logistic regression

- Logistic regression with $\theta = (w, b)$:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \log(1 + e^{w^T\phi(x_i)+b}) - y_i(w^T\phi(x_i) + b) + \tfrac{\lambda}{2}\|w\|_2^2$$

  on the following data set (from logistic regression lecture)
- Polynomial features of degree 6, Tikhonov regularization $\lambda = 0.01$
- Number of decision variables: 28



Algorithms

Compare the following algorithms, all with backtracking:

1. Gradient method
2. Gradient method with fixed diagonal scaling
3. Gradient method with fixed full scaling
4. Newton's method
5. BFGS
6. Limited-memory BFGS with buffer size $m = 3$

Fixed scaling methods

- Logistic regression gradient and Hessian satisfy

$$\nabla f(\theta) = X^T(\sigma(X\theta) - Y) + \lambda w \quad \nabla^2 f(\theta) = X^T \sigma'(X\theta) X + \lambda I_w$$

  where σ is the (vector-version of) sigmoid, and $I_w(w, b) = w$
- The gradient of the sigmoid is 0.25-Lipschitz continuous
- Gradient method with fixed full scaling (3.) uses

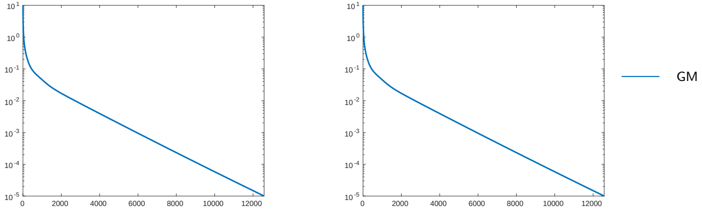
$$H_k = H = 0.25 X^T X + \lambda I_w$$

- Gradient method with fixed diagonal scaling (2.) uses

$$H_k = H = \mathbf{diag}(0.25 X^T X + \lambda I_w)$$

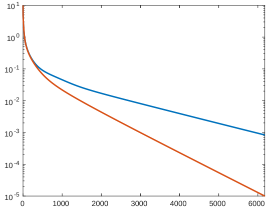
Example – Numerics

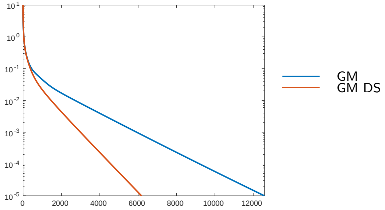
- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- Standard gradient method with backtracking (GM)



Example – Numerics

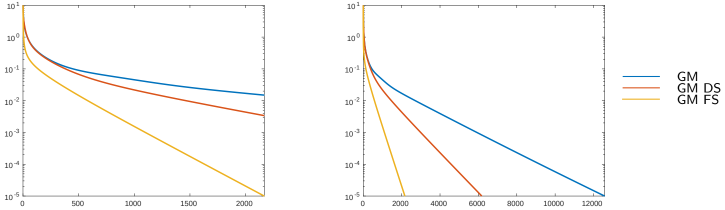
- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- Gradient method with diagonal scaling (GM DS)





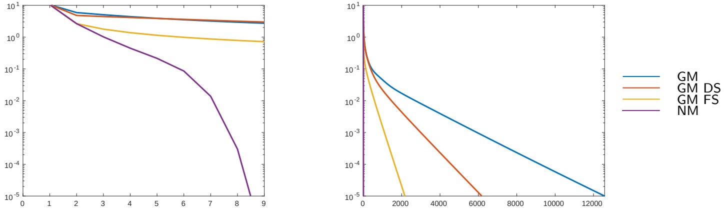
Example – Numerics

- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- Gradient method with full matrix scaling (GM FS)



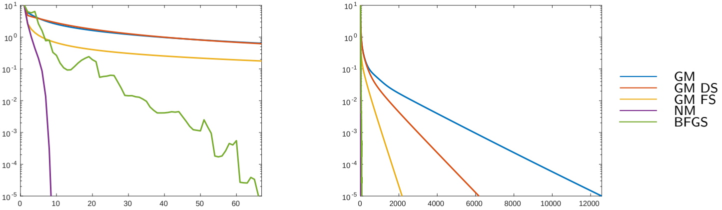
Example – Numerics

- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- Newtons method with backtracking (NM)



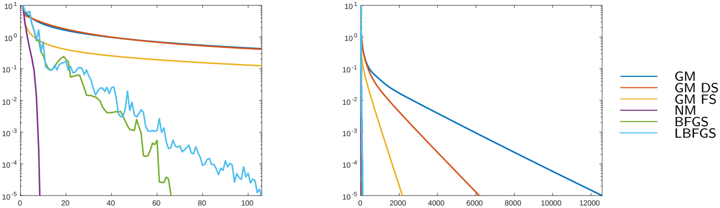
Example – Numerics

- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- BFGS with backtracking (BFGS)



Example – Numerics

- Logistic regression polynomial features of degree 6, $\lambda = 0.01$
- LBFGS with backtracking and buffer length $m = 3$ (LBFGS)



Comments

- We have only compared number of iterations
- Iteration cost in Newton and BFGS much higher than for GM
- Iteration cost for LBFGS similar to for GM
- LBFGS performs very well for smooth problems