Scaling, Newton’s method
quasi-Newton

Pontus Giselsson
Learning goals

• Understand how scaling can be used in proximal gradient method
• Know that scaling can improve performance greatly
• Know Newton’s method and computational effort per iteration
• Know quasi-Newton method and basic idea with secant condition
• Have seen Newton proximal gradient method
Optimization algorithm overview

Algorithms can roughly be divided into the following classes:

- Second-order methods
- Quasi second-order methods
- First-order methods
- Stochastic and coordinate-wise first-order methods
Composite optimization

We consider problems

$$\text{minimize } f(x) + g(x)$$

where

- \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is
  - continuously differentiable (\( \nabla f \) continuous)
  - \( \beta \)-smooth (will use new definition with scaled norm, next slide)
- \( g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\} \) is closed convex
Smoothness w.r.t. $\| \cdot \|_H$

What is $\| \cdot \|_H$:

- Requirement: $H \in \mathbb{R}^{n \times n}$ is symmetric positive definite ($H \succ 0$)
- The norm $\|x\|_H^2 := x^T H x$, for $H = I$, we get $\|x\|_I^2 = \|x\|_2^2$

Smoothness:

- Function $f : \mathbb{R}^n \to \mathbb{R}$ is $\beta$-smooth if for all $x, y \in \mathbb{R}^n$:

\[
  f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2} \|x - y\|_2^2
\]

- We say $f \beta_H$-smoothness w.r.t. scaled norm $\| \cdot \|_H$ if

\[
  f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta_H}{2} \|x - y\|_H^2
\]

  for all $x, y \in \mathbb{R}^n$

- If $f$ is smooth (w.r.t. $\| \cdot \|_2$) it is also smooth w.r.t. $\| \cdot \|_H$
Example: A quadratic

• Let \( f(x) = \frac{1}{2} x^T H x = \frac{1}{2} \|x\|_H^2 \) with \( H \succ 0 \)

• \( f \) is \( 1 \)-smooth w.r.t \( \| \cdot \|_H \) (with equality):

\[
\begin{align*}
    f(y) & = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} \|x - y\|_H^2 \\
    & = \frac{1}{2} x^T H x + (Hx)^T (y - x) + \frac{1}{2} \|x - y\|_H^2 \\
    & = \frac{1}{2} x^T H x + (Hx)^T (y - x) + \frac{1}{2} (\|x\|_H^2 - 2(Hx)^T y + \|y\|_H^2) \\
    & = \frac{1}{2} \|y\|_H^2 = f(y)
\end{align*}
\]

• \( f \) is \( \lambda_{\text{max}}(H) \)-smooth (w.r.t. \( \| \cdot \|_2 \)), continue equality:

\[
\begin{align*}
    f(y) & = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} \|x - y\|_H^2 \\
    \leq & \frac{1}{2} x^T H x + (Hx)^T (y - x) + \frac{\lambda_{\text{max}}(H)}{2} \|x - y\|_2^2
\end{align*}
\]

much more conservative estimate of function!
Proximal Gradient Method
Proximal gradient method

- Proximal gradient method:

\[
x_{k+1} = \arg\min_y \left( f(x_k) + \nabla f(x_k)^T (y - x) + \frac{1}{2\gamma_k} \|y - x_k\|^2 + g(y) \right)
\]

\(\hat{f}_{x_k}(y)\)

approximates function \(f(y)\) around \(x_k\) by \(\hat{f}_{x_k}(y)\)

- The better the approximation, the faster the convergence

- By scaling: we mean to use an approximation of the form

\[
f_{x_k}(y) = f(x_k) + \nabla f(x_k)^T (y - x_k) + \frac{1}{2\gamma_k} \|x_k - y\|^2_{H_k}
\]

where \(H_k \in \mathbb{R}^{n \times n}\) is a positive definite matrix
Gradient descent – Example

• Gradient descent on $\beta$-smooth quadratic problem

$$\minimize_{x} \quad \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Step-size $\gamma = \frac{1}{\beta}$ and norm $\| \cdot \|_2$ in model
Gradient descent – Example

• Gradient descent on $\beta$-smooth quadratic problem

$$\text{minimize } \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Step-size $\gamma = \frac{1}{\beta}$ and norm $\| \cdot \|_2$ in model
Gradient descent – Example

• Gradient descent on $\beta$-smooth quadratic problem

$$\minimize_x \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Step-size $\gamma = \frac{1}{\beta}$ and norm $\| \cdot \|_2$ in model
Gradient descent – Example

• Gradient descent on $\beta$-smooth quadratic problem

\[
\text{minimize } \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

• Step-size $\gamma = \frac{1}{\beta}$ and norm $\| \cdot \|_2$ in model
Gradient descent – Example

• Gradient descent on $\beta$-smooth quadratic problem

$$\min_{x} \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Step-size $\gamma = \frac{1}{\beta}$ and norm $\| \cdot \|_2$ in model
Scaled norm in model

• Gradient descent on $\beta$-smooth quadratic problem

$$\minimize_{x} \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Model $H = \text{diag}(\nabla^2 f)$, $\gamma$ is inverse smoothness w.r.t. $\| \cdot \|_H$
Scaled norm in model

• Gradient descent on β-smooth quadratic problem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
\end{align*}
\]

• Model \( H = \text{diag}(\nabla^2 f) \), \( \gamma \) is inverse smoothness w.r.t. \( \| \cdot \|_H \)
Scaled norm in model

- Gradient descent on $\beta$-smooth quadratic problem

$$\text{minimize } \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Model $H = \text{diag}(\nabla^2 f)$, $\gamma$ is inverse smoothness w.r.t. $\| \cdot \|_H$
Scaled norm in model

- Gradient descent on $\beta$-smooth quadratic problem

$$\min_{x} \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Model $H = \text{diag}(\nabla^2 f)$, $\gamma$ is inverse smoothness w.r.t. $\| \cdot \|_H$
Scaled norm in model

• Gradient descent on $\beta$-smooth quadratic problem

$$\min_x \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

• Model $H = \text{diag}(\nabla^2 f)$, $\gamma$ is inverse smoothness w.r.t. $\| \cdot \|_H$
Example: A quadratic

• Let $f(x) = \frac{1}{2} x^T H x$ with $H \succ 0$, which is 1-smooth w.r.t. $\| \cdot \|_H$
• Approximation with scaled norm $\| \cdot \|_H$ and $\gamma_k = 1$ satisfies $\forall x_k$:

$$\hat{f}_{x_k}(y) = f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{1}{2}\|x_k - y\|_H^2 = f(y)$$

since $f$ is 1-smooth w.r.t. $\| \cdot \|_H$ with equality
• An iteration then reduces to solving problem itself:

$$x_{k+1} = \arg\min_y (\hat{f}_{x_k}(y) + g(y)) = \arg\min_y (f(y) + g(y))$$

• Model very accurate, but very expensive iterations
Scaled proximal gradient method

- Proximal gradient method with scaled norm $\| \cdot \|_{H_k}$:

$$x_{k+1} = \arg\min_y \left( f(x_k) + \nabla f(x_k)^T (y - x) + \frac{1}{2\gamma_k} \| y - x_k \|_2^{H_k} + g(y) \right)$$

$$= \arg\min_y \left( g(y) + \frac{1}{2\gamma_k} \| y - (x_k - \gamma_k H_k^{-1} \nabla f(x_k)) \|_2^{H_k} \right)$$

$$=: \text{prox}_{\gamma_k g}^{H_k}(x_k - \gamma_k H_k^{-1} \nabla f(x_k))$$

where $H_k = I$ gives nominal algorithm (note $\gamma_k$ could be in $H_k$)

- Computational difference per iteration:
  1. Need to invert $H_k^{-1}$ (or solve $H_k d_k = \nabla f(x_k)$)
  2. Need to compute prox with new metric

$$\text{prox}_{\gamma_k g}^{H_k}(z) := \arg\min_x (g(x) + \frac{1}{2\gamma_k} \| x - z \|_{H_k}^2)$$
Computational cost

• Assume that $H_k$ is dense or general sparse
  • $H_k^{-1}$ dense: cubic complexity (vs maybe quadratic for gradient)
  • $H_k^{-1}$ sparse: lower than cubic complexity
  • $\text{prox}_{\gamma_k H_k} g$: difficult optimization problem

• Assume that $H_k$ is diagonal
  • $H_k^{-1}$: invert diagonal elements – linear complexity
  • $\text{prox}_{\gamma_k H_k} g$: often as cheap as nominal prox (e.g., for separable $g$)

• Assume that $H_k$ is block-diagonal with small blocks
  • $H_k^{-1}$: invert individual blocks – also cheap
  • $\text{prox}_{\gamma_k H_k} g$: often quite cheap (e.g., for block-separable $g$)

• If $H_k = I$, method is nominal method
Before discussing choices of $H_k$, we show convergence
Proximal gradient – Optimality condition

• Proximal gradient iteration:

\[ x_{k+1} = \text{prox}_{H_k}^g(x_k - H_k^{-1}\nabla f(x_k)) \]

\[ = \arg\min_y (g(y) + \frac{1}{2}\|y - (x_k - H_k^{-1}\nabla f(x_k))\|^2_{H_k}) \]

where \( x_{k+1} \) is unique due to strong convexity of \( h \)

• Fermat’s rule (and since CQ holds) gives optimality condition:

\[ 0 \in \partial g(x_{k+1}) + \partial h(x_{k+1}) \]

\[ = \partial g(x_{k+1}) + H_k(x_{k+1} - (x_k - H_k^{-1}\nabla f(x_k))) \]

\[ = \partial g(x_{k+1}) + \nabla f(x_k) + H_k(x_{k+1} - x_k) \]

since \( h \) differentiable

• A consequence: \( \partial g(x_{k+1}) \) is nonempty
Fixed-points and convergence

Scaled proximal gradient has

• same fixed-points as nominal proximal gradient
• same convergence guarantees
Proximal gradient – Fixed-point set

• Let $T_{PG}^{H,\gamma} := \text{prox}_{\gamma g}^{H}(I - \gamma H^{-1} \nabla f)$; algorithm $x_{k+1} = T_{PG}^{H,\gamma} x_k$

• Proximal gradient fixed-point set definition

$$\text{fix}T_{PG}^{H,\gamma} = \{ x : x = T_{PG}^{H,\gamma} x \} = \{ x : x = \text{prox}_{\gamma g}^{H}(x - \gamma H^{-1} \nabla f(x)) \}$$

i.e., set of points for which $x_{k+1} = x_k$
Let $H \succ 0$, $\gamma > 0$. Then $\bar{x} \in \text{fix}T^{H,\gamma}_{\text{PG}}$ if and only if $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$.

• Proof: by proximal gradient step optimality condition

\[
\bar{x} \in \text{fix}T^{H,\gamma}_{\text{PG}} \iff \bar{x} = \text{prox}_{\gamma g}^{H}(\bar{x} - \gamma H^{-1}\nabla f(\bar{x})) \\
\iff 0 \in \partial g(\bar{x}) + \gamma^{-1}H(\bar{x} - (\bar{x} - \gamma H^{-1}\nabla f(\bar{x}))) \\
\iff 0 \in \partial g(\bar{x}) + \nabla f(\bar{x})
\]

• Consequence: fixed-point set same for all $H \succ 0$, $\gamma > 0$
• Still call inclusion $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ fixed-point characterization
Meaning of fixed-point characterization

From proximal gradient lecture:

- Fixed-point characterization solves problem when $f$ convex
- For nonconvex $f$, are critical points
Assumptions for convergence – Nonconvex case

• Difference to proximal gradient:
  • new upper bound on \( f \) with scaled norm \( \| \cdot \|_{H_k} \)

• Assumptions:
  
  (i) \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable (not necessarily convex)
  (ii) \( \forall x_k, x_{k+1}, \) it exists \( \beta_k \in [\eta, \eta^{-1}], \rho I \preceq H_k \preceq \rho^{-1} I, \eta, \rho \in (0, 1) \):
  \[
  f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_k - x_{k+1} \|_{H_k}^2
  \]
  which means \( f \) is “locally \( \beta_k \) smooth w.r.t. \( \| \cdot \|_{H_k} \)
  
  (iii) \( g : \mathbb{R}^n \to \mathbb{R} \cup \{ \infty \} \) is closed convex
  (iv) A minimizer exists (and \( p^* = \min_x (f(x) + g(x)) \) is optimal value)
  (v) 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 \( f \) \( \beta \)-smooth
A basic inequality

Using

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

(b) Prox optimality condition: There exists $s_{k+1} \in \partial g(x_{k+1})$

\[
0 = s_{k+1} + \gamma_k^{-1} H_k (x_{k+1} - (x_k - \gamma_k H_k^{-1} \nabla f(x_k)))
\]

(c) Subgradient definition: $g(x_k) \geq g(x_{k+1}) + s_{k+1}^T (x_k - x_{k+1})$

\[
\begin{align*}
 f(x_{k+1}) + g(x_{k+1}) & \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_{k+1} - x_k \|_{H_k}^2 + g(x_{k+1}) \\
 & \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_{k+1} - x_k \|_{H_k}^2 + g(x_k) \\
 & \quad - s_{k+1}^T (x_k - x_{k+1}) \\
 & \overset{(b)}{=} f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_{k+1} - x_k \|_{H_k}^2 + g(x_k) \\
 & \quad + \gamma_k^{-1} H_k (x_{k+1} - (x_k - \gamma_k H_k^{-1} \nabla f(x_k)))^T (x_k - x_{k+1}) \\
 & = f(x_k) + g(x_k) - (\gamma_k^{-1} - \frac{\beta_k}{2}) \| x_{k+1} - x_k \|_{H_k}^2
\end{align*}
\]
Function value decrease

• What conclusions can we draw from

\[ f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + g(x_k) - (\gamma_k^{-1} - \frac{\beta_k}{2})\|x_{k+1} - x_k\|_2^2 \]

• The requirements: \( \gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon] \):
  • since \( \beta_k \in [\eta, \eta^{-1}] \) there is \( \epsilon > 0 \) such that \( [\epsilon, \frac{2}{\beta_k} - \epsilon] \) nonempty
  • therefore \( \delta > 0 \) exists such that

\[ \gamma_k^{-1} \in \left[ \frac{\beta_k}{2} + \delta, \delta^{-1} \right] \quad \Rightarrow \quad \gamma_k^{-1} - \frac{\beta_k}{2} \geq \delta > 0 \]

which implies, since \( H_k \succeq \rho I \), that function value decreases:

\[ f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + g(x_k) - \rho \delta \|x_{k+1} - x_k\|_2^2 \]

• Not very useful!
Fixed-point residual converges

• Rearrange inequality from previous slide:

\[ \rho \delta \| x_{k+1} - x_k \|^2_2 \leq f(x_k) + g(x_k) - (f(x_{k+1}) + g(x_{k+1})) \]

• Telescope summation gives for all \( n \in \mathbb{N} \):

\[
\rho \delta \sum_{k=1}^{n} \| x_{k+1} - x_k \|^2_2 \leq \sum_{k=1}^{n} (f(x_k) + g(x_k) - (f(x_{k+1}) + g(x_{k+1})))
\]

\[
= f(x_1) + g(x_1) - (f(x_{n+1}) + g(x_{n+1}))
\]

\[
\leq f(x_1) + g(x_1) - p^* < \infty
\]

where \( p^* = \min_x (f(x) + g(x)) \) and \( < \infty \) since \( x_1 \in \text{dom} g \)

• Since \( \rho \delta > 0 \), this implies:

\[
\| \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k)) - x_k \|_2 = \| x_{k+1} - x_k \|_2 \to 0
\]
Residual convergence – Implication

What does $\|\text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2 \to 0$ mean and imply?

- That fixed-point equation will be satisfied in the limit
- By prox-grad optimality condition:
  \[
  \partial g(x_{k+1}) + \nabla f(x_k) \ni \gamma_k^{-1} H_k(x_k - x_{k+1}) \to 0
  \]
  as $k \to \infty$ (since $\gamma_k \geq \epsilon$, i.e., $0 < \gamma_k^{-1} \leq \epsilon^{-1}$) or equivalently
  \[
  \partial g(x_{k+1}) + \nabla f(x_{k+1}) \ni \gamma_k^{-1} H_k(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k) \to 0
  \]
  where $u_k \to 0$ is concluded by continuity of $\nabla f$, implications:
  - Fixed-point characterization satisfied in the limit
  - Nonconvex $f$: Critical point definition satisfied in the limit
  - Convex $f$: Global optimality condition satisfied in the limit
  - However, does not imply that $(x_k)$ converges to a fixed-point
Selecting algorithm parameters

- How to select $\beta_k$, $\gamma_k$, and $H_k$?
- Start with $\beta_k$ and $\gamma_k$, given $H_k$.
Choose $\beta_k$ and $\gamma_k$.

- Convergence based on assumption that $\beta_k$ known that satisfies
  \[
  f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \|x_k - x_{k+1}\|^2_{H_k}
  \]
  call this descent condition (DC)

- If $H_k = H$ and $f$ is $\beta$-smooth w.r.t. $\|\cdot\|_H$; $\beta_k = \beta$ works since
  \[
  f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2} \|x - y\|^2_H
  \]
  for all $x, y$
Choose $\beta_k$ and $\gamma_k$ – Backtracking

• Backtracking, choose $\delta > 1$, $\beta_k \in [\eta, \eta^{-1}]$ and increase $\beta_k$ as:
  1. choose $\gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon]$
  2. compute $x_{k+1} = \text{prox}_{\gamma_k}^{H_k} (x_k - \gamma_k H_k^{-1} \nabla f(x_k))$
  3. if descent condition (DC) satisfied
     break
     else
     set $\beta_k \leftarrow \delta \beta_k$ and go to 1
     end
• Initialization of $\beta_k$ depends on choice of $H_k$
Backtracking – Convergence

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

\[ f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{\beta_k}{2} \|x_k - x_{k+1}\|_2^2 \]

will hold within finite number of backtracking steps

• Assume and recall that

  • \( f : \mathbb{R}^n \to \mathbb{R} \) is \( \beta \)-smooth
  • \( \beta_k \in [\eta, \eta^{-1}] \), \( \rho I \preceq H_k \preceq \rho^{-1} I \), \( \eta, \rho \in (0, 1) \):

which gives

\[ f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{\beta}{2} \|x_k - x_{k+1}\|_2^2 \]

\[ \leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{\beta}{2\rho} \|x_k - x_{k+1}\|_2^2 \]

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

- $H_k$ should capture (some) second-order (Hessian) information
- Examples:
  - $H_k = I$ is identity matrix (gives proximal gradient method)
  - $H_k = \text{diag}(h)$ 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 (proximal Newton method)
  - $H_k$ is chosen from (limited memory) quasi-Newton
- Recall, prox update at iteration $x_k$ is
  \[
  \text{prox}^{H_k}_{\gamma_k}(x_k - \gamma_k H_k^{-1} \nabla f(x_k))
  \]
- Will first focus on case without prox, i.e., $g = 0$;
  \[
  x_{k+1} = x_k - \gamma_k H_k^{-1} \nabla f(x_k)
  \]
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
- Requires backtracking scheme (no fixed $\gamma_k$ works in general)
  - Initialize $\beta_k = 1$ in every iteration $k$ in backtracking
  - Select stepsize $\gamma_k = \frac{1}{\beta_k}$
  - Within finite number of iterations, $\gamma_k = 1$ will be accepted
- Typically requires very few iterations to converge
- However, can be costly to invert matrix (solve linear system)
- Note: $\nabla^2 f(x_k)$ must be positive definite, i.e., $\nabla^2 f(x) \succ 0$:
  - always true if problem strictly convex; else:
  - add $\epsilon I$ with $\epsilon > 0$ such that $H_k = \nabla^2 f(x_k) + \epsilon I \succ 0$
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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice continuously differentiable

• Many schemes for finding $H_k$, will cover BFGS$^1$

---

$^1$ BFGS: Broyden-Fletcher-Goldfarb-Shanno
Secant condition

- Consider quadratic approximation of the function $f$
  \[ \hat{f}_{x_k}(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} \| x_k - x \|_H^2 \]

- Approximation satisfies $\nabla \hat{f}_{x_k}(x_k) = \nabla f(x_k)$

- 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
  - secant condition contains only $n$ constraints $\Rightarrow$ underdetermined
  - Select $H_k$ “close” to $H_{k-1}$ subject to secant condition holds:

    $$\begin{align*}
    \text{minimize} & \quad D(H_k, H_{k-1}) \\
    \text{subject to} & \quad H_k s_k = y_k
    \end{align*}$$

    where $D$ measures distance between $H_k$ and $H_{k-1}$

- Often initialized as $H_0 = I$
Different choices of $D$

- $D(H_k, H_{k-1}) = \|H_k - H_{k-1}\|_F^2$
  - gives Broyden method
  - $H_k$ not necessarily symmetric and positive definite
- $D(H_k, H_{k-1}) = \text{tr}(H_k^{-1}H_{k-1}) - \log \det(H_k^{-1}H_{k-1}) - n$
  - Cost called *relative entropy* and method is BFGS
  - $H_k$ is symmetric and positive definite
  - This method is preferred over Broyden for smooth minimization
The BFGS update formula

- BFGS update formula \((H_+ = H_k, H = H_{k-1}, s = s_k, y = y_k)\):

\[
H_+ = H - \frac{H s s^T H}{s^T H s} + \frac{y y^T}{y^T s}
\]  

(1)

- Estimating the Hessian requires a matrix inverse in algorithm:

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

- Store and update inverse Hessian \(B_k = H_k^{-1}\) instead
The BFGS Hessian inverse update formula

• Write Hessian update formula as

\[ H_+ = H + \begin{bmatrix} Hs & y \end{bmatrix} \begin{bmatrix} \frac{1}{s^T H s} & 0 \\ 0 & \frac{1}{y^T s} \end{bmatrix} \begin{bmatrix} s^T H^T \\ y^T \end{bmatrix} \]

• Identify structure and use Woodbury matrix inversion formula:

\[(A + UDV)^{-1} = A^{-1} - A^{-1}U(D^{-1} + VA^{-1}U)^{-1}VA^{-1}\]

To after algebraic manipulation find inverse update

\[ B_+ = (I - \frac{sy^T}{y^T s})B(I - \frac{ys^T}{y^T s}) + \frac{s s^T}{y^T s} \]

where \( B = H^{-1} \) and \( B_+ = H_+^{-1} \)

• Algorithm performs multiplications with \( B_k \) (initialize \( B_0 = I \))

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

• Cheaper updates than using Newton’s method
Limited-memory BFGS (LBFGS)

• BFGS stores full matrix $B_k$ and has dense matrix update:

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

memory requirement: $n^2$ elements (or $\frac{n(n+1)}{2}$ since symmetric)

• LBFGS uses less memory, derived from implicit form BFGS

• Inverse update formula

$$B_+ = (I - \frac{sy^T}{y^Ts}) B (I - \frac{ys^T}{y^Ts}) + \frac{ss^T}{y^Ts}$$

which can be evaluated when multiplied by vector $g$ as

$$B_+ g = (I - s \frac{y^T}{y^Ts}) B (g - y\frac{s^Tg}{y^Ts}) + s \frac{s^Tg}{y^Ts}$$

which gives

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

where $\alpha = \frac{s^Tg}{y^Ts}, q = g - y\alpha, p = Bq, \beta = \frac{y^Tp}{y^Ts}$
Implicit form BFGS

- Instead of storing $B_k$, we store all $s_l$ and $y_l$ for $l = \{1, \ldots, k\}$
- Recursively use (2) $k$ times; gives two-loop calculation of $B + g$
  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 $p = p + (\alpha_l - \beta_l) s_l$

- Memory requirement: $2nk$, grows with iteration $k$
- Inefficient implementation for BFGS, but used for LBFGS
LBFGS – Limited memory BFGS

• Recursively use (2) \( m \) times, where \( m \) is buffer size
• 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 \( p = B_{k-m} q \)
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 \( p = p + (\alpha_l - \beta_l) s_l \)

• Memory requirement: \( 2nm \), where buffer size \( m \) can be small
Example – Logistic regression

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

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

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 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 \( \sigma \) 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

\[ H_k = H = 0.25X^TX + \lambda I_w \]

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

\[ H_k = H = \text{diag}(0.25X^TX + \lambda I_w) \]
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 three (LBFGS)
Adding nonsmooth term

- Newton’s method and BFGS for smooth minimization
- Example with logistic regression is smooth
- What happens when we have nonsmooth term? Iteration is

\[ x_{k+1} = \text{prox}_{\gamma_k g}^H(x_k - \gamma_k H_k^{-1} \nabla f(x_k)) \]

\[ = \arg\min_y (g(y) + \frac{1}{2\gamma_k} \|y - (x_k - \gamma_k H_k^{-1} \nabla f(x_k))\|^2_{H_k}) \]

- Depending on structure of \( H_k \), \text{prox} can be expensive to compute
Diagonal scaling

- A diagonal $H_k$ does often not increase computational cost of prox
- If model used in method

$$\hat{f}_{x_k}(y) = f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{\beta_k}{2} \|y - x_k\|^2_{H_k}$$

better than nominal model,

$$\hat{f}_{x_k}(y) = f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{\beta_k}{2} \|y - x_k\|^2_2$$

method can be improved
- Difficult to find good diagonal scaling, examples:
  - If $f$ quadratic, let $H_k$ be diagonal elements of Hessian
  - Adaptive choices for stochastic gradient descent (Adagrad, Adam)
Dense matrix scaling

- Assume we select $H_k$ to be dense (all elements nonzero)
- Cost of computing prox will dominate cost of $H_k^{-1}\nabla f(x_k)$
- No reason not to use Hessian
- Prox using BFGS will be as expensive, but worse approximation
Newton proximal gradient

• The Newton proximal gradient method is

\[ x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)) \]

\[ = \arg\min_y (f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{1}{2\gamma_k} \|y - x_k\|_2^2 + \nabla^2 f(x_k) + g(y)) \]

• If \( f \) quadratic, iteration reduces to solving problem
• For nonquadratic \( f \), in each iteration: solve prox with dense \( H_k \)
• Scaled coordinate descent can solve these subproblems efficiently
• Proximal Newton plus scaled coordinate descent can be efficient
Newton proximal gradient – Backtracking

Backtracking is needed for convergence

- Previous backtracking works; find $\beta_k$ such that

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

requires recomputing prox for every test-value of $\beta_k$, expensive!

- Algorithm and backtracking variation (avoids prox recomputation)

$$ v_k = \arg\min_y (f(x_k) + \nabla f(x_k)^T (y - x_k) + \frac{1}{2} \| y - x_k \|^2 \nabla^2 f(x_k) + g(y)) $$

$$ x_{k+1} = x_k + \gamma_k (v_k - x_k) $$

with backtracking from $\gamma_k = 1$ and decrease while

$$ F(x_{k+1}) > F(x_k) + \alpha (\gamma_k \nabla f(x_k)^T (x_{k+1} - x_k) + (g(x_{k+1}) - g(x_k))) $$

where using $\alpha \in (0, \frac{1}{2}]$, $F = f + g$, requires different analysis
Second-order alternative: Interior point method

- Can treat nonsmooth term $g$ in $\min_x (f(x) + g(x))$ by
  1. replacing it by smooth approximation $\hat{g}$
  2. solve smooth problem $f + \hat{g}$
     (use previous iterate solution as initial point)
  3. refine approximation $\hat{g}$ of $g$ and goto 2.

- This is idea behind interior point methods
- Why refining approximation?
  - Newton method has very fast local convergence
  - If solution with previous $\hat{g}$ close enough, new solution found fast