# Modern Optimization Techniques

## 2. Unconstrained Optimization / 2.4. Quasi-Newton Methods

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Computer Science
University of Hildesheim, Germany

# Syllabus

| | | |
|---|---|---|
| Mon. 30.10. | (0) | 0. Overview |
| | | **1. Theory** |
| Mon. 6.11. | (1) | 1. Convex Sets and Functions |
| | | **2. Unconstrained Optimization** |
| Mon. 13.11. | (2) | 2.1 Gradient Descent |
| Mon. 20.11. | (3) | 2.2 Stochastic Gradient Descent |
| Mon. 27.11. | (4) | 2.3 Newton's Method |
| Mon. 4.12. | (5) | 2.4 Quasi-Newton Methods |
| Mon. 11.12. | (6) | 2.5 Subgradient Methods |
| Mon. 18.12. | (7) | 2.6 Coordinate Descent |
| | — | — *Christmas Break* — |
| | | **3. Equality Constrained Optimization** |
| Mon. 8.1. | (8) | 3.1 Duality |
| Mon. 15.1. | (9) | 3.2 Methods |
| | | **4. Inequality Constrained Optimization** |
| Mon. 22.1. | (10) | 4.1 Primal Methods |
| Mon. 29.1. | (11) | 4.2 Barrier and Penalty Methods |
| Mon. 5.2. | (12) | 4.3 Cutting Plane Methods |

# Outline

# Outline

1. Excursion: Inverting Matrices

2. The Idea of Quasi-Newton Methods

3. BFGS and L-BFGS

# Matrix Inversion

Given a matrix $A \in \mathbb{R}^{n \times n}$, its inverse $A^{-1}$ is a matrix such that:

$$AA^{-1} = \mathbf{I}$$

where

- $\mathbf{I}$ is the identity matrix

- If no such matrix $A^{-1}$ exists $A$ is called
  - **singular** or
  - **non-invertible**

# Matrix Inversion — Easy cases

1. **small matrices:**
   - for $A \in \mathbb{R}^{2\times 2}$ the inverse can be computed analytically:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

   - slightly more complex closed formula for $A \in \mathbb{R}^{3\times 3}$

2. **orthogonal matrices:**
   - $A \in \mathbb{R}^{n\times n}$ is **orthogonal** if $A^T A = \mathbf{I}$
   - thus $A^{-1} = A^T$

# Matrix Inversion — Easy cases

3. **diagonal matrices:**
   - $A \in \mathbb{R}^{n \times n}$ is **diagonal** if $A_{ij} = 0$ for all $i \neq j$
   - thus $A = \text{diag}(a_1, a_2, \ldots, a_n)$ with

$$\text{diag}(a_1, \ldots, a_n) := \begin{pmatrix} a_1 & 0 & \ldots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & a_n \end{pmatrix}$$

   - $A^{-1} = \text{diag}(\frac{1}{a_1}, \frac{1}{a_2}, \ldots, \frac{1}{a_n})$

# General Matrix Inversion

Generally, inverting a matrix $A \in \mathbb{R}^{n \times n}$ is equivalent to solving a linear system of equations with $n$ different right sides:

$$AA^{-1} = I \iff Ax^i = e^i, \quad e^i := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow i\text{-th position} \ , \quad i = 1, \dots, n$$

via $A^{-1} = (x^1, x^2, \dots, x^n)$

If an inverse is used only once to compute $x := A^{-1}b$ for a vector $b \in \mathbb{R}^n$, it usually is faster to solve the linear system of equations $Ax = b$ instead.

# General Matrix Inversion / Complexity

Inverting matrices and solving systems of linear equations can be accomplished two ways:

1. algebraic algorithms ("direct algorithms")
   - like Gaussian elimination, LU decomposition, QR decomposition
   - complexity generally $O(n^3)$
   - there exist specialized matrix inversion algorithms with lower costs
     - Strassen algorithm $O(n^{2.807})$
     - Coppersmith–Winograd algorithm $O(n^{2.376})$
     - but they are impractical and not used in implementations

2. optimization algorithms ("iterative algorithms")
   - Gauss-Seidel, Gradient-descent type of algorithms

# Inverse of a Rank-One Update

Lemma (Inverse of a Rank-One Update – Sherman-Morrison formula)
*For $A \in \mathbb{R}^{n \times n}$ invertible and $a, b \in \mathbb{R}^n$:*

$$(A + ab^T)^{-1} = A^{-1} - \frac{A^{-1}ab^TA^{-1}}{1 + b^TA^{-1}a}$$

Meaning:

- the inverse of a rank-one update can be computed fast
  - in $O(n^2)$ instead of in $O(n^3)$
  - if the inverse of the original matrix is available

# Inverse of a Rank-One Update / Proof

Show that the right side has the property of the inverse:

$$
\begin{aligned}
(A + ab^T)(A^{-1} - \frac{A^{-1}ab^TA^{-1}}{1 + b^TA^{-1}a})
&= I + ab^TA^{-1} - \frac{ab^TA^{-1} + ab^TA^{-1}ab^TA^{-1}}{1 + b^TA^{-1}a})\\
&= I + ab^TA^{-1} - \frac{a(1 + b^TA^{-1}a)b^TA^{-1}}{1 + b^TA^{-1}a})\\
&= I + ab^TA^{-1} - ab^TA^{-1} = I
\end{aligned}
$$

# Outline

# Underlying Idea

- Approximate the Hessian with a matrix $H$ that is fast to invert.

$$H \approx \nabla^2 f(x)$$

  - Use a low-rank update

$$H^{(0)} := I$$

$$H^{\text{next}} = H + \sum_{k=1}^{K} a_k b_k^T$$

    - fast to invert using $K$-times inverses of rank-one updates

$$(H^{-1})^{(0)} = I$$

$$(H^{-1})^{\text{next}} = H^{-1} + \dots$$

- Compute the next direction using the inverse of the Hessian approximation:

$$\Delta x = -H^{-1} \nabla f(x)$$

# Properties of the Hessian $\nabla^2 f(x)$

- It fulfills the **secant condition**

$$H(y - x) = \nabla f(y) - \nabla f(x)$$

approximately:

$$\nabla^2 f(y)(y - x) \stackrel{y \to x}{\to} \nabla f(y) - \nabla f(x)$$

  - due to first order Taylor expansion of $\nabla f$:

$$\nabla f(x) \approx \nabla f(y) + \nabla^2 f(y)(x - y)$$

  - if $H$ fulfills the secant condition,
    then the second order approximation of $f$ by $\nabla f$ and $H$ around $x$
    has gradient $\nabla f(x)$ at $x$

- it is symmetric

- it is positive semidefinite

- it is positive definite
  - for a strongly convex objective function

# Hessian Approximations

Idea: search for a matrix $H$ that

- has some of the properties of the Hessian and
- is fast to compute
  - e.g., by a low-rank update from the previous approximation:

$$H^{(0)} := I$$

$$H^{\text{next}} = H + \sum_{k=1}^{K} a_k b_k^T, \quad a_k, b_k \in \mathbb{R}^n$$

# Symmetric Rank-One Update

Lemma (Symmetric Rank-One Update)

*There exists exactly one low-rank update such that*

i) *$H$ fulfils the secant condition*

$$H^{next}s = g, \quad s := x^{next} - x, \quad g := \nabla f(x^{next}) - \nabla f(x)$$

ii) *$H$ is symmetric and*

iii) *which is a rank-one update:*

$$a_1 = b_1 := \frac{g - Hs}{((g - Hs)^T s)^{\frac{1}{2}}}$$

$$H^{next} = H + \frac{(g - Hs)(g - Hs)^T}{(g - Hs)^T s}$$

# Symmetric Rank-One Update / Proof

If $H$ and $H^{\text{next}}$ are symmetric, then $a_1 b_1^T$ must be also symmetric.

$$a_1 b_1^T \stackrel{!}{=} (a_1 b_1^T)^T = b_1 a_1^T \quad | \cdot a_1$$

$$a_1 b_1^T a_1 \stackrel{!}{=} b_1 a_1^T a_1 \quad \rightsquigarrow b_1 = \beta a_1, \quad \beta \in \mathbb{R}, \beta \neq 0$$

$$\begin{aligned}
H^{\text{next}} &\underset{iii}{=} H + \beta a_1 a_1^T \\
H^{\text{next}} s &\underset{i}{=} g \\
\beta a_1 a_1^T s &= g - Hs \quad \rightsquigarrow \quad a_1 = \gamma(g - Hs), \quad \gamma \in \mathbb{R} \\
\beta\gamma(g - Hs)\gamma(g - Hs)^T s &= g - Hs \\
\beta\gamma^2 (g - Hs)^T s &= 1
\end{aligned}$$

$$\beta = 1, \quad \gamma = ((g - Hs)^T s)^{-\frac{1}{2}}, \quad a_1 = \frac{g - Hs}{((g - Hs)^T s)^{\frac{1}{2}}}$$

# Symmetric Rank-One Update / Inverse

Lemma (Symmetric Rank-One Update / Inverse)

*The inverse $H^{-1}$ of the approximate Hessian in the symmetric rank-one update is*

$$(H^{-1})^{next} = H^{-1} + \frac{(s - H^{-1}g)(s - H^{-1}g)^T}{(s - H^{-1}g)^T g}$$

# Symmetric Rank-One Update / Inverse / Proof

Apply Morrison-Sherman to the rank-one update of the Hessian approximation:

$$
\begin{aligned}
(H^{-1})^{\text{next}} &= H^{-1} - \frac{H^{-1}(g-Hs)(g-Hs)^T H^{-1}}{(g-Hs)^T s(1+\frac{(g-Hs)^T H^{-1}(g-Hs)}{(g-Hs)^T s})} \\
&= H^{-1} - \frac{(H^{-1}g-s)(H^{-1}g-s)^T}{(g-Hs)^T s + (g-Hs)^T H^{-1}(g-Hs)} \\
&= H^{-1} - \frac{(H^{-1}g-s)(H^{-1}g-s)^T}{(H^{-1}g-s)^T(Hs+g-Hs)} \\
&= H^{-1} - \frac{(H^{-1}g-s)(H^{-1}g-s)^T}{(H^{-1}g-s)^T g} \\
&= H^{-1} + \frac{(s-H^{-1}g)(s-H^{-1}g)^T}{(s-H^{-1}g)^T g}
\end{aligned}
$$

# Newton's Method (Review)

**min-newton**$(f, \nabla f, \nabla^2 f, x^{(0)}, \mu, \epsilon, K)$ :
  for $k := 1, \ldots, K$:
    $\Delta x^{(k-1)} := -\nabla^2 f(x^{(k-1)})^{-1} \nabla f(x^{(k-1)})$
    if $-\nabla f(x^{(k-1)})^T \Delta x^{(k-1)} < \epsilon$:
      return $x^{(k-1)}$
    $\mu^{(k-1)} := \mu(f, x^{(k-1)}, \Delta x^{(k-1)})$
    $x^{(k)} := x^{(k-1)} + \mu^{(k-1)} \Delta x^{(k-1)}$
  return "not converged"

where

- $f$ objective function
- $\nabla f$, $\nabla^2 f$ gradient and Hessian of objective function $f$
- $x^{(0)}$ starting value
- $\mu$ step length controller
- $\epsilon$ convergence threshold for Newton's decrement
- $K$ maximal number of iterations

# Quasi-Newton Method / SR1

**min-qnewton-sr1**$(f, \nabla f, x^{(0)}, \mu, \epsilon, K)$ :
  $A^{(0)} := I$
  for $k := 1, \ldots, K$:
    $\Delta x^{(k-1)} := -A^{(k-1)} \nabla f(x^{(k-1)})$
    if $-\nabla f(x^{(k-1)})^T \Delta x^{(k-1)} < \epsilon$:
      return $x^{(k-1)}$
    $\mu^{(k-1)} := \mu(f, x^{(k-1)}, \Delta x^{(k-1)})$
    $x^{(k)} := x^{(k-1)} + \mu^{(k-1)} \Delta x^{(k-1)}$
    $s^{(k)} := x^{(k)} - x^{(k-1)}$
    $g^{(k)} := \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$
    $A^{(k)} := A^{(k-1)} + \frac{(s^{(k)} - A^{(k-1)} g^{(k)})(s^{(k)} - A^{(k-1)} g^{(k)})^T}{(s^{(k)} - A^{(k-1)} g^{(k)})^T g^{(k)}}$
  return "not converged"

where

- $A = H^{-1}$ the inverse of the approximative Hessian

# Outline

# Positive Definite Hessian Approximations

- There is no rank-one update with positive definite Hessian approximation $H$.
- There are many rank-two update schemes with positive definite Hessian approximation $H$.
- Most widely used: BFGS
  - developed independently by Broyden, Fletcher, Goldfarb and Shanno in 1970

$$H^{\text{next}} = H - \frac{Hs(Hs)^T}{s^T Hs} + \frac{gg^T}{g^T s}$$

# BFGS

## Lemma (BFGS)

*The BFGS update*

$$H^{next} = H - \frac{Hs(Hs)^T}{s^T Hs} + \frac{gg^T}{g^T s}$$

i) *fulfils the secant condition,*

ii) *yields symmetric $H$ and*

iii) *yields positive definite $H$, if $g^T s > 0$.*

*The inverse $H^{-1}$ of the approximate Hessian is*

$$(H^{-1})^{next} = H^{-1} + \frac{(s - H^{-1}g)s^T + s(s - H^{-1}g)^T}{s^T g} - \frac{(s - H^{-1}g)^T g}{(s^T g)^2} ss^T$$

$$= (I - \frac{sg^T}{s^T g})H^{-1}(I - \frac{gs^T}{s^T g}) + \frac{ss^T}{s^T g}$$

# BFGS / Proof (1/3)

i) BFGS fulfils the secant condition:

$$H^{\text{next}} s = Hs - \frac{Hs(Hs)^T s}{s^T Hs} + \frac{gg^T s}{g^T s}$$
$$= Hs - Hs + g = g$$

ii) BFGS yields symmetric $H$: obvious.
iii) BFGS yields positive definite $H$:
If $H$ is positive definite, it can be represented $H = LL^T$ with a non-singular $L$ (Cholesky decomposition).

$$H^{\text{next}} = LWL^T$$
$$W := I - \frac{\tilde{s}\tilde{s}^T}{\tilde{s}^T\tilde{s}} + \frac{\tilde{g}\tilde{g}^T}{\tilde{g}^T\tilde{s}}, \quad \tilde{s} := L^T s, \quad \tilde{g} := L^{-1} g$$

$H^{\text{next}}$ will be pos.def., if $W$ is.

# BFGS / Proof (2/3)

for any $v \in \mathbb{R}^n$:

$$0 \overset{?}{<} v^T W v = v^T v - \frac{(v^T\tilde{s})^2}{\tilde{s}^T\tilde{s}} + \frac{(v^T\tilde{g})^2}{\tilde{g}^T\tilde{s}}$$

$$= ||v||^2 - \frac{||v||^2||\tilde{s}||^2\cos^2\theta_1}{||\tilde{s}||^2} + \frac{(v^T\tilde{g})^2}{\tilde{g}^T\tilde{s}}$$

$$= ||v||^2(1-\cos^2\theta_1) + \frac{(v^T\tilde{g})^2}{\tilde{g}^T\tilde{s}}$$

$$= ||v||^2\sin^2\theta_1 + \frac{(v^T\tilde{g})^2}{\tilde{g}^T\tilde{s}}$$

$$\tilde{g}^T\tilde{s} = g^T s \underset{\text{assumption}}{>} 0$$

- if $v = \lambda\tilde{s}, \lambda \in \mathbb{R}, \lambda \neq 0$:
  - $\sin^2\theta_1 = 0$, but
  - $(v^T\tilde{g})^2 = \lambda^2(\tilde{s}^T\tilde{g})^2 > 0$
- if $v \neq \lambda\tilde{s}, \lambda \in \mathbb{R}, \lambda \neq 0$:
  - $\sin^2\theta_1 > 0$

# BFGS / Proof (3/3)

To derive the inverse of the approximate Hessian,
apply Morrison-Sherman twice.

# Quasi-Newton Method / BFGS

**min-qnewton-bfgs**$(f, \nabla f, x^{(0)}, \mu, \epsilon, K)$ :

$\quad A^{(0)} := I$

$\quad$ for $k := 1, \ldots, K$:

$\qquad \Delta x^{(k-1)} := -A^{(k-1)} \nabla f(x^{(k-1)})$

$\qquad$ if $-\nabla f(x^{(k-1)})^T \Delta x^{(k-1)} < \epsilon$:

$\qquad\quad$ return $x^{(k-1)}$

$\qquad \mu^{(k-1)} := \mu(f, x^{(k-1)}, \Delta x^{(k-1)})$

$\qquad x^{(k)} := x^{(k-1)} + \mu^{(k-1)} \Delta x^{(k-1)}$

$\qquad s^{(k)} := x^{(k)} - x^{(k-1)}$

$\qquad g^{(k)} := \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$

$$A^{(k)} := A^{(k-1)} + \frac{(s^{(k)} - A^{(k-1)} g^{(k)})(s^{(k)})^T + s^{(k)}(s^{(k)} - A^{(k-1)} g^{(k)})^T}{(s^{(k)})^T g^{(k)}}$$

$$- \frac{(s^{(k)} - A^{(k-1)} g^{(k)})^T g^{(k)}}{((s^{(k)})^T g^{(k)})^2} s^{(k)} (s^{(k)})^T$$

$\quad$ return "not converged"

where

- $A = H^{-1}$ the inverse of the approximative Hessian

# Avoid Materialization of $A$

- In the previous form, BFGS still requires $n^2$ storage to materialize the inverse $A$ of the approximate Hessian.

- For any vector $v \in \mathbb{R}^n$, images $A^{(K)}v$ can be computed from the recursive formula from vectors $g^{(k)}, s^{(k)}$ $(k = 1, \ldots, K)$

# Compute Image $Av$ without Materialization of $A$

$\textbf{bfgs-image-iha}(v, (s^{(k)})_{k=1,\ldots,K}, (g^{(k)})_{k=1,\ldots,K}, (\rho^{(k)})_{k=1,\ldots,K}, A^{(0)})$ :

  $q := v$

  for $k := K, \ldots, 1$:

    $\alpha_k := \rho^{(k)} (s^{(k)})^T q$

    $q := q - \alpha_k g^{(k)}$

  $r := A^{(0)} q$

  for $k := 1, \ldots, K$:

    $\beta := \rho^{(k)} (g^{(k)})^T r$

    $r := r + s^{(k)} (\alpha_i - \beta)$

  return $r$

where

- $v \in \mathbb{R}^n$ vector who's image to compute, usually $\nabla f(x^{(k)})$
- $(s^{(k)})_{k=1,\ldots,K}, (g^{(k)})_{k=1,\ldots,K}$ as defined earlier
- $\rho^{(k)} := 1/(g^{(k)})^T s^{(k)}$
- $A^{(0)}$ initial inverse Hessian, e.g. $I$.

# Quasi-Newton Method / BFGS w/o Materialization of $A$

**min-qnewton-bfgs-nomat**$(f, \nabla f, x^{(0)}, \mu, \epsilon, K)$ :

  for $k := 1, \ldots, K$:

    $\Delta x^{(k-1)} := -$bfgs-image-iha$(\nabla f(x^{(k-1)}, s^{(1:k-1)},$

      $g^{(1:k-1)}, \rho^{(1:k-1)}, I)$

    if $-\nabla f(x^{(k-1)})^T \Delta x^{(k-1)} < \epsilon$:

      return $x^{(k-1)}$

    $\mu^{(k-1)} := \mu(f, x^{(k-1)}, \Delta x^{(k-1)})$

    $x^{(k)} := x^{(k-1)} + \mu^{(k-1)} \Delta x^{(k-1)}$

    $s^{(k)} := x^{(k)} - x^{(k-1)}$

    $g^{(k)} := \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$

    $\rho^{(k)} := 1/(g^k)^T s^{(k)}$

  return "not converged"

# Avoid Materialization of $A$

- Storing all vectors $g^{(1:K)}, s^{(1:K)}$ requires $2Kn$ storage, i.e. is only memory efficient for $K \ll n$.
- Instead of computing the inverse $A$ of the approximate Hessian by all these vectors, we could
  - forget the older ones, i.e.,
  - just store and compute the $M \ll n$ most recent ones.
- This approach is called **Limited Memory BFGS** (L-BFGS)

# Quasi-Newton Method / L-BFGS

**min-qnewton-lbfgs**$(f, \nabla f, x^{(0)}, \mu, \epsilon, K, M)$ :
  for $k := 1, \ldots, K$:
    $k_0 := \max\{1, k-1-M+1\}$
    $\Delta x^{(k-1)} := -\text{bfgs-image-iha}(\nabla f(x^{(k-1)}, s^{(k_0:k-1)},$
      $g^{(k_0:k-1)}, \rho^{(k_0:k-1)}, I)$
    if $-\nabla f(x^{(k-1)})^T \Delta x^{(k-1)} < \epsilon$:
      return $x^{(k-1)}$
    $\mu^{(k-1)} := \mu(f, x^{(k-1)}, \Delta x^{(k-1)})$
    $x^{(k)} := x^{(k-1)} + \mu^{(k-1)} \Delta x^{(k-1)}$
    $s^{(k)} := x^{(k)} - x^{(k-1)}$
    $g^{(k)} := \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$
    $\rho^{(k)} := 1/(g^k)^T s^{(k)}$
  return "not converged"

Implementations need to ensure that the old vectors $s^{(1:k_0-1)}, g^{(1:k_0-1)}$ do not consume any memory (i.e., are overwritten by the more recent ones).

# Summary

- Rank One Updates $A + ab^T$ of a matrix $A$ can be inverted fast (in $O(n^2)$; if an inverse of $A$ is available; Sherman-Morrison formula).
- Quasi-Newton methods are Newton methods with approximated Hessian.
  - approximations should share properties of the Hessian
    - secant condition, symmetry, positive definiteness
  - maintain the inverse of the Hessian (not the Hessian itself)
- symmetric rank one update:
  - only one such rank one update (not even pos.def.)
- BFGS rank two update:
  - one out of many such rank two updates
  - pos.def.
- Images of a vector under the inverse Hessian can be computed even without materializing the inverse Hessian:
  - compute the image recursively from the images under the rank one update steps
  - Limited Memory BFGS (L-BFGS)

# Further Readings

- Quasi-Newton methods are not covered by Boyd and Vandenberghe [2004]
- BFGS:
  - [Nocedal and Wright, 2006, ch. 6]
  - [Griva et al., 2009, ch. 12.3]
    the update formulas for the inverse are in ch. 13.5.
  - [Sun and Yuan, 2006, ch. 5.1]
- L-BFGS:
  - [Nocedal and Wright, 2006, ch. 7]
  - [Griva et al., 2009, ch. 13.5]

# References I

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge Univ Press, 2004.

Igor Griva, Stephen G. Nash, and Ariela Sofer. *Linear and nonlinear optimization*. Society for Industrial and Applied Mathematics, 2009.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

Wenyu Sun and Ya-Xiang Yuan. *Optimization Theory and Methods, Nonlinear Programming*. Springer, 2006.