Modern Optimization Techniques

2. Unconstrained Optimization / 2.3. Newton's Method

Lars Schmidt-Thieme

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

original slides by Lucas Rego Drumond (ISMLL)

Syllabus

Tue. 18.10.	(0)	0. Overview
		1. Theory
Tue. 25.10.	(1)	1. Convex Sets and Functions
		2. Unconstrained Optimization
Tue. 1.11	(2)	2.1 Gradient Descent
Tue. 8.11.	(3)	2.2 Stochastic Gradient Descent
Tue. 15.11.	(4)	(ctd.)
Tue. 22.11.	(5)	2.3 Newton's Method
Tue. 29.11.	(6)	2.3b Newton's Method (Part 2)
Tue. 6.12.	(7)	2.4 Subgradient Methods
Tue. 13.12.	(8)	2.5 Coordinate Descent
		3. Equality Constrained Optimization
Tue. 20.12.	(9)	3.1 Duality
	—	— Christmas Break —
Tue. 10.1.	(10)	3.2 Methods
		4. Inequality Constrained Optimization
Tue. 17.1.	(11)	4.1 Interior Point Methods
Tue. 24.1.	(11)	4.2 Cutting Plane Method
		5. Distributed Optimization
Tue. 31.1.	(12)	5.1 Alternating Direction Method of Multipliers

Outline

Outline

An idea using second order approximations

Be $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^N$ open and f convex:

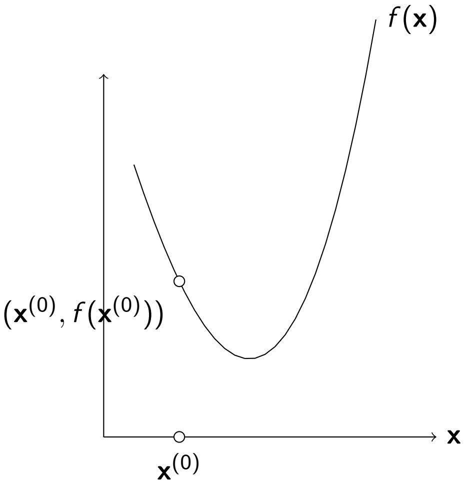
$$\arg\min_{x \in X} \quad f(\mathbf{x})$$

- Let $\mathbf{x}^{(k)}$ the last iterate
- Compute a quadratic approximation $\hat{f}$ of f around $\mathbf{x}^{(k)}$
- Find the minimum of the quadratic approximation $\hat{f}$ and take it as next iterate:

$$\mathbf{x}^{(k+1)} := \arg\min_{x \in X} \hat{f}(\mathbf{x})$$

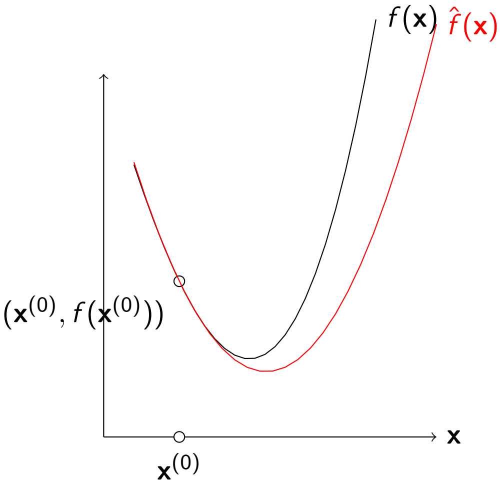
An idea using second order approximations

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - 3)^2 + \frac{1}{10}\mathbf{x}^3$$



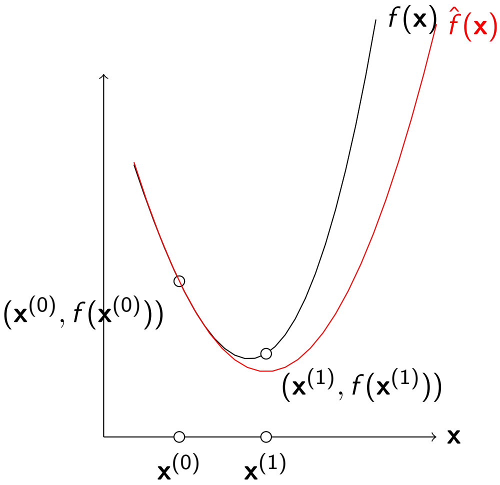
An idea using second order approximations

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-3)^2 + \frac{1}{10}\mathbf{x}^3$$



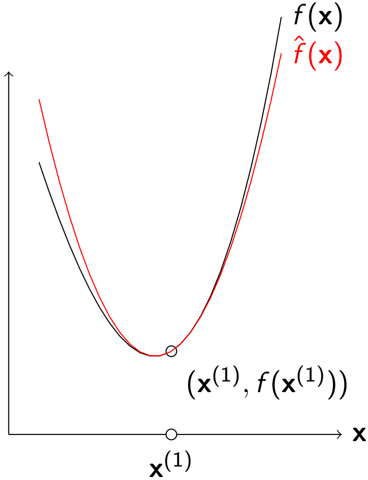
An idea using second order approximations

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-3)^2 + \frac{1}{10}\mathbf{x}^3$$



An idea using second order approximations

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-3)^2 + \frac{1}{10}\mathbf{x}^3$$



Taylor Approximation

Be $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^N$ an infinitely differentiable function,
$\mathbf{a} \in X$ any point.

f can be represented by its **Taylor expansion**:

$$f(x) = \sum_{k=0}^{\infty} \frac{\nabla^k f(\mathbf{a})}{k!}(\mathbf{x}-\mathbf{a})^k$$
$$= f(\mathbf{a}) + \frac{\nabla f(\mathbf{a})}{1!}(\mathbf{x}-\mathbf{a}) + \frac{\nabla^2 f(\mathbf{a})}{2!}(\mathbf{x}-\mathbf{a})^2 + \frac{\nabla^3 f(\mathbf{a})}{3!}(\mathbf{x}-\mathbf{a})^3 + \cdots$$

For x close enough to a and K large enough,
f can be approximated by its **truncated Taylor expansion**:

$$f(\mathbf{x}) \approx \sum_{k=0}^{K} \frac{\nabla^k f(\mathbf{a})}{k!}(\mathbf{x}-\mathbf{a})^k$$

Note: For $N > 1$, $\nabla^k f(x)$ is a tensor of order k and $\nabla^k f(x)(x-a)^k$ a tensor product.

Second Order Approximation

Let us take the second order approximation of a twice differentiable function $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^N$ at a point $\mathbf{x}$:

$$\hat{f}(\mathbf{y}) := f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^T\nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})$$

We want to find the point $x^{\text{next}} := \arg\min_y \hat{f}(y)$:

$$\nabla_{\mathbf{y}}\hat{f}(\mathbf{y}) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) \overset{!}{=} 0$$

$$\rightsquigarrow \quad \mathbf{y} = \mathbf{x} - \nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x})$$

Newton's Step

- Newton's method is a descent method
- It uses the descent direction

$$\Delta \mathbf{x} := -\nabla^2 f(\mathbf{x})^{-1} \nabla f(\mathbf{x})$$

  called **Newton step**.
  - the negative gradient
  - twisted by the local curvature (Hessian)
- Newton's step is affine invariant,
  while the gradient step is not.

Newton's Step / Proof

(i) Show that the Gradient step is not affine invariant.
for $g(y) := f(Ay)$ with a pos.def. matrix A

$$\nabla_y g(y) = A^T \nabla_x f(Ay) \stackrel{?}{=} A^{-1} \nabla_x f(x), \quad \text{for } x := Ay$$

No, as in general $A^T \neq A^{-1}$.

(ii) Show that Newton's step is affine invariant.

$$\begin{aligned}
\nabla_y^2 g(y) &= A^T \nabla_x^2 f(Ay) A \\
\Delta y &= (\nabla_y^2 g(y))^{-1} \nabla_y g(y) \\
&= A^{-1} \nabla_x^2 f(Ay)^{-1} (A^T)^{-1} A^T \nabla_x f(Ay) \\
&= A^{-1} \nabla_x^2 f(Ay)^{-1} \nabla_x f(Ay) \\
&= A^{-1} \nabla_x^2 f(x)^{-1} \nabla_x f(x), \quad \text{for } x := Ay
\end{aligned}$$

Newton's Stepsize

- For quadratic objective functions f:
  - Newton's method will find the optimum in a single step
    - with stepsize 1

    (**pure Newton**)

- For general objective functions:
  - a possibly smaller stepsize has to be used
    (**damped Newton**)
  - any stepsize controller is applicable

Newton Decrement

$$\lambda(x) := (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{\frac{1}{2}}$$

is called **newton decrement**.

Basic properties:

(i)

$$\lambda(x) = (\Delta x^T \nabla^2 f(x) \Delta x)^{\frac{1}{2}}$$

(ii)

$$\nabla \lambda(x)^2 = -f(x)^T \Delta x$$

(iii)

$$f(x) - \inf_y \hat{f}(y) = f(x) - \hat{f}(x + \Delta x) = \frac{1}{2}\lambda(x)^2$$

(iv) The Newton decrement is affine invariant.

Newton Decrement / Proofs

ad (i) insert the definition of $\Delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$

ad (ii) same as for (ii)

ad (iii) expand the definition of $\hat{f}$ at $y := x + \Delta x$ and use (i).

$$\hat{f}(y) := f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$

ad (iv) for $g(y) := f(Ay)$ with a pos.def. matrix A

$$\nabla_y g(y) = A^T \nabla_x f(Ay), \quad \nabla_y^2 g(y) = A^T \nabla_x^2 f(Ay) A$$

$$\begin{aligned} \lambda_g(y) &= \nabla_x f(Ay)^T A A^{-1} \nabla_x^2 f(Ay)^{-1} (A^T)^{-1} A^T \nabla_x f(Ay)^T \\ &= \nabla_x f(Ay)^T \nabla_x^2 f(Ay)^{-1} \nabla_x f(Ay)^T \\ &= \lambda_f(x) \text{ at } x := Ay \end{aligned}$$

Newton's Method

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

Considerations

- Works extremely well for a lot of problems
- requires f to be twice differentiable
- Computing, storing and inverting the Hessian limits scalability for high dimensional problems
  - as the Hessian has N^2 elements.

Newton's method - Example

For $\mathbf{x} \in \mathbb{R}$

$$\min_{\mathbf{x}} \quad (2\mathbf{x} - 4)^4$$

Algorithm:

- $\nabla f(\mathbf{x}) = 8\ (2\mathbf{x} - 4)^3$
- $\nabla^2 f(\mathbf{x}) = 48\ (2\mathbf{x} - 4)^2$
- Step:

$$\begin{aligned} \Delta \mathbf{x} &= \nabla^2 f(\mathbf{x})^{-1} \nabla f(\mathbf{x}) \\ &= -\frac{1}{6}(2\mathbf{x} - 4) \end{aligned}$$

- Update:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \mu^{(k)} \Delta x^{(k)} \\ &= x^{(k)} - \frac{1}{6}(2x^{(k)} - 4) \end{aligned}$$

Newton's method - Example

$$
\begin{aligned}
x^{(0)} &:= 10\\
x^{(1)} &= 10.0 - \frac{1}{6}(2 \cdot 10.0 - 4) = 7.33333\\
x^{(2)} &= 7.33333 - \frac{1}{6}(2 \cdot 7.33333 - 4) = 5.55556\\
x^{(3)} &= 5.55556 - \frac{1}{6}(2 \cdot 5.55556 - 4) = 4.37037\\
x^{(4)} &= 4.37037 - \frac{1}{6}(2 \cdot 4.37037 - 4) = 3.58025\\
x^{(5)} &= 3.58025 - \frac{1}{6}(2 \cdot 3.58025 - 4) = 3.0535\\
x^{(6)} &= 3.0535 - \frac{1}{6}(2 \cdot 3.0535 - 4) = 2.70233\\
x^{(7)} &= 2.70233 - \frac{1}{6}(2 \cdot 2.70233 - 4) = 2.46822\\
x^{(8)} &= 2.46822 - \frac{1}{6}(2 \cdot 2.46822 - 4) = 2.31215
\end{aligned}
$$

Outline

1. Newton's Method

2. Convergence

3. Example: Logistic Regression

Newton Decrement / Strongly Convex Functions

If f is strongly convex ($\nabla^2 f(x) \succeq mI, m \in \mathbb{R}^+$), then

(i)

$$m||\Delta x||_2^2 \leq \lambda(x)^2 \leq M||\Delta x||_2^2$$

(ii)

$$\frac{1}{M}||\nabla f(x)||_2^2 \leq \lambda(x)^2 \leq \frac{1}{m}||\nabla f(x)||_2^2$$

where $\nabla^2 f(x) \preceq MI, M \in \mathbb{R}^+$.

Newton Decrement / Strongly Convex Functions / Proofs

ad (i)

$$\lambda(x)^2 = \Delta x^T \nabla^2 f(x) \Delta x \geq m||\Delta x||_2^2$$
$$\lambda(x)^2 = \Delta x^T \nabla^2 f(x) \Delta x \leq M||\Delta x||_2^2$$

ad (ii) The inverse of $\nabla^2 f(x)$ has inverse eigenvalues, thus

$$\nabla^2 f(x)^{-1} \preceq \frac{1}{m} I$$
$$\nabla^2 f(x)^{-1} \succeq \frac{1}{M} I$$

Then proceed as (i).

Convergence / Assumptions

Until the end of this section, assume

I. f is strongly convex (m, M),

II. $\nabla^2 f(x)$ is Lipschitz-continuous:
$||\nabla^2 f(y) - \nabla^2 f(x)||_2 \leq L||y - x||_2, \quad L \in \mathbb{R}^+$ and

III. backtracking steplength control is used $(\alpha \leq \frac{1}{2}, \beta)$

Convergence / Damped Phase

Theorem (Convergence of Newton's Algorithm / Damped Phase)

Far away from the optimum,

(i) *backtracking may select stepsizes* $t \leq 1$ *(be damped) and*

(ii) f *is reduced by at least a constant each step.*

$$\text{for } \nabla ||f(x)||_2 \geq \eta : \quad f(x^{next}) - f(x) \leq -\gamma$$

$$\text{with } \gamma := \alpha\beta\frac{m}{M^2}\eta^2$$

Convergence / Damped Phase / Proof

$$f(x+t\Delta x) \underset{\text{s.c. ii}}{\leq} f(x) + t\nabla f(x)^T \Delta x + \frac{M}{2}||\Delta x||_2^2 t^2$$

$$\underset{\text{dec. ii}}{\leq} f(x) - t\lambda(x)^2 + \frac{M}{2m} t^2 \lambda(x)^2 \quad (1)$$

$\hat{t} := m/M$ satisfies exit condition of backtracking:

$$f(x+\hat{t}\Delta x) \underset{(1)}{\leq} f(x) - \frac{m}{M}\lambda(x)^2 + \frac{m}{2M}\lambda(x)^2$$

$$= f(x) - \frac{m}{2M}\lambda(x)^2$$

$$\underset{\alpha \leq \frac{1}{2}}{\leq} f(x) - \alpha \hat{t} \lambda(x)^2$$

and thus stepsize

$$t \geq \beta \frac{m}{M} \quad (2)$$

Convergence / Damped Phase / Proof (2/2)

$$
\begin{aligned}
f(x^{\text{next}}) - f(x) &\leq -\alpha t \lambda(x)^2 \\
&\underset{(2)}{\leq} -\alpha\beta\frac{m}{M}\lambda(x)^2 \\
&\underset{\text{dec s.c. ii}}{\leq} -\alpha\beta\frac{m}{M^2}||\nabla f(x)||_2^2 \\
&\underset{||\nabla f(x)||_2 \geq \eta}{\leq} -\alpha\beta\frac{m}{M^2}\eta^2 = -\gamma
\end{aligned}
$$

Convergence / Pure Phase

Theorem (Convergence of Newton's Algorithm / Pure Phase)

Close to the optimum,

(i) *backtracking always selects stepsize* $t = 1$ *and*

(ii) $\nabla f(x)$ *is shrunken quadratically.*

$$\text{for } ||\nabla f(x)||_2 < \eta: \ ||\nabla f(x^{next})||_2 \leq \frac{L}{2m^2}(||\nabla f(x)||_2)^2$$

$$\text{with } \eta \leq 3(1-2\alpha)\frac{m^2}{L}$$

(iii) *it stays close to the optimum.*

$$\text{for } ||\nabla f(x)||_2 < \eta: \ ||\nabla f(x^{next})||_2 < \eta$$

$$\text{with } \eta := \min\{1, 3(1-2\alpha)\}\frac{m^2}{L}$$

Convergence / Pure Phase / Proof (1/6)

(i) show backtracking accepts stepsize $t = 1$, if $\eta \leq 3(1-2\alpha)\frac{m^2}{L}$

$$||\nabla^2 f(x+t\Delta) - \nabla^2 f(x)||_2 \underset{\text{II}}{\leq} tL||\Delta x||_2$$

$$\begin{aligned} \rightsquigarrow \ |\Delta x^T(\nabla^2 f(x+t\Delta) - \nabla^2 f(x))\Delta x| \\ \leq ||\nabla^2 f(x+t\Delta) - \nabla^2 f(x)||_2 ||\Delta x||_2^2 \\ = tL||\Delta x||_2^3 \end{aligned} \quad (1)$$

Convergence / Pure Phase / Proof (2/6)

Compute a lower bound for

$$
\begin{aligned}
\tilde{f}(t) &:= f(x + t\Delta x)\\
\tilde{f}'(t) &= \Delta x^T \nabla f(x + t\Delta x)\\
\tilde{f}''(t) &= \Delta x^T \nabla^2 f(x + t\Delta x)\Delta x\\
|\tilde{f}''(t) - \tilde{f}''(0)| &\underset{(1)}{\leq} tL||\Delta x||_2^3\\
\tilde{f}''(t) &\leq \tilde{f}''(0) + tL||\Delta x||_2^3\\
&\underset{\text{dec i, dec s.c. i}}{\leq} \lambda(x)^2 + t\frac{L}{m^{\frac{3}{2}}}\lambda(x)^3 \qquad | \int_0^1 (\ldots)dt\\
\tilde{f}'(t) &\leq \tilde{f}'(0) + t\lambda(x)^2 + t^2\frac{L}{2m^{\frac{3}{2}}}\lambda(x)^3\\
&\underset{\text{dec ii}}{\leq} -\lambda(x)^2 + t\lambda(x)^2 + t^2\frac{L}{2m^{\frac{3}{2}}}\lambda(x)^3
\end{aligned}
$$

Convergence / Pure Phase / Proof (3/6)

$$\tilde{f}'(t) \leq -\lambda(x)^2 + t\lambda(x)^2 + t^2 \frac{L}{2m^{\frac{3}{2}}}\lambda(x)^3 \qquad | \int_0^1 (\ldots)dt$$

$$\tilde{f}(t) \leq \tilde{f}(0) - t\lambda(x)^2 + \frac{1}{2}t^2\lambda(x)^2 + t^3 \frac{L}{6m^{\frac{3}{2}}}\lambda(x)^3 \qquad |t = 1$$

$$f(x + \Delta x) = \tilde{f}(1) \leq \tilde{f}(0) - \lambda(x)^2 + \frac{1}{2}\lambda(x)^2 + \frac{L}{6m^{\frac{3}{2}}}\lambda(x)^3$$

$$= f(x) - \lambda(x)^2(\frac{1}{2} - \frac{L}{6m^{\frac{3}{2}}}\lambda(x)) \qquad (2)$$

Convergence / Pure Phase / Proof (4/6)

$$
\begin{aligned}
\lambda(x) &\underset{\text{dec s.c. ii}}{\leq} \frac{1}{m^{\frac{1}{2}}}||\nabla f(x)||_2 \\
&\underset{||\nabla f(x)||_2 < \eta}{<} \frac{1}{m^{\frac{1}{2}}}\eta = \frac{1}{m^{\frac{1}{2}}} 3(1-2\alpha)\frac{m^2}{L} = 3(1-2\alpha)\frac{m^{\frac{3}{2}}}{L} \qquad (3) \\
f(x+\Delta x) &\underset{(2)}{\leq} f(x) - \lambda(x)^2(\frac{1}{2} - \frac{L}{6m^{\frac{3}{2}}}\lambda(x)) \\
&\underset{(3)}{\leq} f(x) - \lambda(x)^2(\frac{1}{2} - \frac{L}{6m^{\frac{3}{2}}} 3(1-2\alpha)\frac{m^{\frac{3}{2}}}{L}) \\
&= f(x) - \alpha\lambda(x)^2
\end{aligned}
$$

and thus stepsize $t = 1$ fulfils the exit condition.

Convergence / Pure Phase / Proof (5/6)

(ii) show decrease in $\nabla f(x^{\text{next}})$:

$$\begin{aligned}
||\nabla f(x^{\text{next}})||_2 &\underset{t=1}{=} ||\nabla f(x + \Delta x)||_2 \\
&\underset{\text{def } \Delta x}{=} ||\nabla f(x + \Delta x) - \nabla f(x) - \nabla^2 f(x)\Delta x||_2 \\
&\underset{(*)}{=} || \int_0^1 (\nabla^2 f(x + t\Delta x) - \nabla^2 f(x))\Delta x \, dt||_2 \\
&\leq \int_0^1 ||(\nabla^2 f(x + t\Delta x) - \nabla^2 f(x))||_2 dt \; ||\Delta x||_2 \\
&\underset{\text{II}}{\leq} \int_0^1 Lt||\Delta x||_2 dt ||\Delta x||_2 = \frac{1}{2} L||\Delta x||_2^2 \\
&\underset{\text{def } \Delta x}{=} \frac{1}{2} L||\nabla^2 f(x)^{-1} \nabla f(x)||_2^2 \\
&\underset{\text{dec s.c. ii}}{\leq} \frac{L}{2m^2} ||\nabla f(x)||_2^2
\end{aligned}$$

where (*) $\nabla f(x + \Delta x) = \nabla^2 f(x)\Delta x + \int_0^1 \nabla^2 f(x + t\Delta x)\Delta x \, dt$

Convergence / Pure Phase / Proof (6/6)

(iii) show that Newton stays close to the optimum:

$$||\nabla f(x^{\text{next}})||_2 \underset{ii}{\leq} \frac{L}{2m^2}||\nabla f(x)||_2^2 \leq \frac{L}{2m^2}\eta^2 \underset{\text{def } \eta}{\leq} \frac{1}{2}\eta < \eta$$

Convergence

Theorem (Convergence of Newton's Algorithm)

If

(i) *f is strongly convex (m, M),*

(ii) *$\nabla^2 f(x)$ is Lipschitz-continuous:*
$||\nabla^2 f(y) - \nabla^2 f(x)||_2 \leq L||y-x||_2, \quad L \in \mathbb{R}^+$ *and*

(iii) *backtracking steplength control is used ($\alpha \leq \frac{1}{2}, \beta$)*

then

$$f(x^{(k)}) - p^* \leq \frac{2m^3}{L^2}\left(\frac{1}{2}\right)^{2^{k-l+1}}, \quad k \geq l$$

$$l := \lceil \frac{f(x^{(0)}) - p^*}{\gamma} \rceil, \quad \gamma := \alpha\beta\frac{m}{M^2}\eta^2, \quad \eta := \min\{1, 3(1-2\alpha)\}\frac{m^2}{L}$$

Convergence / Proof

- If initially we are far away from the minimum, latest after l steps we must be close (damped phase ii) and then

$$\frac{L}{2m^2}\nabla f(x^{(l)}) \leq \frac{L}{2m^2}\eta \leq \frac{L}{2m^2}\frac{m^2}{L} \leq \frac{1}{2} \quad (1)$$

- In the pure phase $k > l$ we have (pure phase ii)

$$\frac{L}{2m^2}\nabla f(x^{(k)}) \leq (\frac{L}{2m^2}\nabla f(x^{(k-1)}))^2 \underset{\text{rec}}{\leq} (\frac{L}{2m^2}\nabla f(x^{(l)}))^{2^{k-l}} \underset{(1)}{\leq} (\frac{1}{2})^{2^{k-l}}$$

$$\nabla f(x^{(k)}) \leq \frac{2m^2}{L}(\frac{1}{2})^{2^{k-l}} \quad (2)$$

$$f(x^{(k)}) - p^* \underset{\text{s.c. i}}{\leq} \frac{1}{2m}||\nabla f(x^{(k)})||_2^2 \underset{(2)}{\leq} \frac{1}{2m}\left(\frac{2m^2}{L}(\frac{1}{2})^{2^{k-l}}\right)^2$$

$$= \frac{2m^3}{L^2}(\frac{1}{2})^{2^{k-l+1}}$$

Outline

1. Newton's Method
2. Convergence
3. Example: Logistic Regression

Practical Example: Household Location

Suppose we have the following data about different households:

- Number of workers in the household (a_1)
- Household composition (a_2)
- Weekly household spending (a_3)
- Gross normal weekly household income (a_4)
- **Region** (y): North $y = 1$ or south $y = 0$

We want to creat a model of the location of the household

Practical Example: Household Spending

If we have data about m households, we can represent it as:

$$A_{m,n} = \begin{pmatrix} 1 & a_{1,2} & \dots & a_{1,n} \\ 1 & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

We can model the household location is a linear combination of the household features with parameters $\mathbf{x}$:

$$\hat{y}_i = \sigma(\mathbf{x}^T \mathbf{a_i}) = \sigma(\mathbf{x}_0 1 + \mathbf{x}_1 a_{i,1} + \mathbf{x}_2 a_{i,2} + \mathbf{x}_3 a_{i,3} + \mathbf{x}_4 a_{i,4})$$

where: $\sigma(x) = \frac{1}{1+e^{-x}}$

Example II - Logistic Regression

The logistic regression learning problem is

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$A_{m,n} = \begin{pmatrix} 1 & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ 1 & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{m,1} & a_{m,2} & a_{m,3} & a_{m,4} \end{pmatrix} \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Logistic Regression

First we need to compute the gradient of our objective function:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} y_i \frac{1}{\sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

Logistic Regression

First we need to compute the gradient of our objective function:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} y_i \frac{1}{\sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

Logistic Regression

First we need to compute the gradient of our objective function:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} y_i \frac{1}{\sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

$$-(1 - y_i) \frac{1}{1 - \sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

Logistic Regression

First we need to compute the gradient of our objective function:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} y_i \frac{1}{\sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$
$$- (1 - y_i) \frac{1}{1 - \sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$
$$= \sum_{i=1}^{m} y_i a_{ik} \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) - (1 - y_i) a_{ik} \sigma(\mathbf{x}^T \mathbf{a_i})$$

Logistic Regression

First we need to compute the gradient of our objective function:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i \log \sigma(\mathbf{x}^T \mathbf{a_i}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}^T \mathbf{a_i}))$$

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} y_i \frac{1}{\sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

$$-(1 - y_i) \frac{1}{1 - \sigma(\mathbf{x}^T \mathbf{a_i})} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) a_{ik}$$

$$= \sum_{i=1}^{m} y_i a_{ik} \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i})\right) - (1 - y_i) a_{ik} \sigma(\mathbf{x}^T \mathbf{a_i})$$

$$= \sum_{i=1}^{m} a_{ik} \left(y_i - \sigma(\mathbf{x}^T \mathbf{a_i})\right)$$

Logistic Regression

$$\frac{\partial f}{\partial \mathbf{x}_k} = \sum_{i=1}^{m} a_{ik} \left(y_i - \sigma(\mathbf{x}^T \mathbf{a_i}) \right)$$

Now we need to compute the Hessian matrix:

$$\frac{\partial^2 f}{\partial \mathbf{x}_k \partial \mathbf{x}_j} = \sum_{i=1}^{m} -a_{ik} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(1 - \sigma(\mathbf{x}^T \mathbf{a_i}) \right) a_{ij}$$
$$= \sum_{i=1}^{m} a_{ik} a_{ij} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(\sigma(\mathbf{x}^T \mathbf{a_i}) - 1 \right)$$

The Hessian H is an $n \times n$ matrix such that:

$$H_{k,j} = \sum_{i=1}^{m} a_{ik} a_{ij} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(\sigma(\mathbf{x}^T \mathbf{a_i}) - 1 \right)$$

Logistic Regression

So we have our gradient $\nabla f \in \mathbb{R}^n$ such that

$$\nabla_{\mathbf{x}_k} f = \sum_{i=1}^{m} a_{ik} \left(y_i - \sigma(\mathbf{x}^T \mathbf{a_i}) \right)$$

And the Hessian $H \in \mathbb{R}^{n \times n}$:

$$H_{k,j} = \sum_{i=1}^{m} a_{ik} a_{ij} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(\sigma(\mathbf{x}^T \mathbf{a_i}) - 1 \right)$$

the newton update rule is:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu H^{-1} \nabla f$$

Newton's Method for Logistic Regression - Considerations

The newton update rule is:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu H^{-1} \nabla f$$

Biggest problem:

How to efficiently compute H^{-1} for:

$$H_{k,j} = \sum_{i=1}^{m} a_{ik} a_{ij} \sigma(\mathbf{x}^T \mathbf{a_i}) \left(\sigma(\mathbf{x}^T \mathbf{a_i}) - 1 \right)$$

Considerations:

- H is symmetric: $H_{k,j} = H_{j,k}$

Further Readings

- Newton's method including convergence proof
  - [Boyd and Vandenberghe, 2004, ch. 9.5]

References I

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