Optimization methods
Lecture 8: Newton method. Quasi-Newton methods

Alexandr Katrutsa

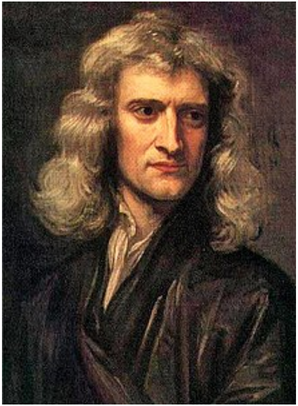
Modern State of Artificial Intelligence Masters Program
Moscow Institute of Physics and Technology

Brief reminder of the previous lecture

- Randomness in optimization problems
- Stochastic gradient descent
- Variance reduction technique

Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$



Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Second-order method

Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Second-order method
- Quadratic approximation

$$\hat{f}(\mathbf{h}) = f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h}$$

Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Second-order method
- Quadratic approximation

$$\hat{f}(\mathbf{h}) = f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h}$$

- If $f''(\mathbf{x}) \succ 0$, then

  $$\hat{f}(\mathbf{h}) \to \min_{\mathbf{h}}$$

  is convex

Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Second-order method
- Quadratic approximation

$$\hat{f}(\mathbf{h}) = f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h}$$

- If $f''(\mathbf{x}) \succ 0$, then

$$\hat{f}(\mathbf{h}) \to \min_{\mathbf{h}}$$

  is convex
- From the FOOC follows

$$f'(\mathbf{x}) + f''(\mathbf{x})\mathbf{h} = 0 \quad \Rightarrow \quad \mathbf{h}^* = -f''(\mathbf{x})^{-1} f'(\mathbf{x})$$

Newton method

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Second-order method
- Quadratic approximation

$$\hat{f}(\mathbf{h}) = f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h}$$

- If $f''(\mathbf{x}) \succ 0$, then
$$\hat{f}(\mathbf{h}) \to \min_{\mathbf{h}}$$
is convex
- From the FOOC follows

$$f'(\mathbf{x}) + f''(\mathbf{x})\mathbf{h} = 0 \quad \Rightarrow \quad \mathbf{h}^* = -f''(\mathbf{x})^{-1} f'(\mathbf{x})$$

- Newton method $\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)}$

Newton method for the system of non-linear equations

- System of non-linear equations

$$G(\mathbf{x}) = 0, \quad G : \mathbb{R}^n \to \mathbb{R}^n$$

Newton method for the system of non-linear equations

- System of non-linear equations

$$G(\mathbf{x}) = 0, \quad G : \mathbb{R}^n \to \mathbb{R}^n$$

- Linear approximation

$$G(\mathbf{x}_k + \Delta\mathbf{x}) \approx G(\mathbf{x}_k) + G'(\mathbf{x}_k)\Delta\mathbf{x} = 0,$$

where $G'(\mathbf{x})$ is Jacobi matrix

Newton method for the system of non-linear equations

- System of non-linear equations

$$G(\mathbf{x}) = 0, \quad G : \mathbb{R}^n \to \mathbb{R}^n$$

- Linear approximation

$$G(\mathbf{x}_k + \Delta\mathbf{x}) \approx G(\mathbf{x}_k) + G'(\mathbf{x}_k)\Delta\mathbf{x} = 0,$$

  where $G'(\mathbf{x})$ is Jacobi matrix
- If $G'(\mathbf{x})$ is invertible, then

$$\Delta\mathbf{x} = -G'(\mathbf{x}_k)^{-1}G(\mathbf{x}_k)$$

Newton method for the system of non-linear equations

- System of non-linear equations

$$G(\mathbf{x}) = 0, \quad G : \mathbb{R}^n \to \mathbb{R}^n$$

- Linear approximation

$$G(\mathbf{x}_k + \Delta\mathbf{x}) \approx G(\mathbf{x}_k) + G'(\mathbf{x}_k)\Delta\mathbf{x} = 0,$$

where $G'(\mathbf{x})$ is Jacobi matrix

- If $G'(\mathbf{x})$ is invertible, then

$$\Delta\mathbf{x} = -G'(\mathbf{x}_k)^{-1}G(\mathbf{x}_k)$$

- Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - G'(\mathbf{x}_k)^{-1}G(\mathbf{x}_k)$$

What is the difference between these methods?

- Assume the function $f(\mathbf{x})$ in problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

is convex

What is the difference between these methods?

- Assume the function $f(\mathbf{x})$ in problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

  is convex
- Then FOOC is a system of non-linear equations

$$f'(\mathbf{x}^*) = G(\mathbf{x}) = 0$$

What is the difference between these methods?

- Assume the function $f(\mathbf{x})$ in problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

  is convex
- Then FOOC is a system of non-linear equations

$$f'(\mathbf{x}^*) = G(\mathbf{x}) = 0$$

- Linear system to get direction $\mathbf{h}$

$$f'(\mathbf{x}) + f''(\mathbf{x})\mathbf{h} = 0$$

  is equivalent to the system in Newton method for problem (1)

Convergence

Assumption $f''(\mathbf{x}) \succ 0$:

- if $f''(\mathbf{x}) \nsucc 0$, method does not work
- there exist modifications to process this case

Convergence

Assumption $f''(\mathbf{x}) \succ 0$:

- if $f''(\mathbf{x}) \not\succ 0$, method does not work
- there exist modifications to process this case

Local convergence: Newton method convergence depends on the choice of $\mathbf{x}_0$ and it can

- converge
- diverge
- oscillate

Convergence

Assumption $f''(\mathbf{x}) \succ 0$:

- if $f''(\mathbf{x}) \nsucc 0$, method does not work
- there exist modifications to process this case

Local convergence: Newton method convergence depends on the choice of $\mathbf{x}_0$ and it can

- converge
- diverge
- oscillate

Damped Newton method

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)}$$

- Adaptive step size is similar to the gradient descent
- Adaptive step size expands convergence region

Local superlinear convergence

- Let $\mathbf{x}^*$ be local minimum, then

$$f'(\mathbf{x}^*) = 0, \quad f''(\mathbf{x}^*) \succ 0$$

Local superlinear convergence

- Let $\mathbf{x}^*$ be local minimum, then

$$f'(\mathbf{x}^*) = 0, \quad f''(\mathbf{x}^*) \succ 0$$

- Taylor expansion

$$0 = f'(\mathbf{x}^*) = f'(\mathbf{x}_k) + f''(\mathbf{x}_k)(\mathbf{x}^* - \mathbf{x}_k) + o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

Local superlinear convergence

- Let $\mathbf{x}^*$ be local minimum, then

$$f'(\mathbf{x}^*) = 0, \quad f''(\mathbf{x}^*) \succ 0$$

- Taylor expansion

$$0 = f'(\mathbf{x}^*) = f'(\mathbf{x}_k) + f''(\mathbf{x}_k)(\mathbf{x}^* - \mathbf{x}_k) + o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- After multiplying by $f''(\mathbf{x}_k)^{-1}$

$$\mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

Local superlinear convergence

- Let $\mathbf{x}^*$ be local minimum, then

$$f'(\mathbf{x}^*) = 0, \quad f''(\mathbf{x}^*) \succ 0$$

- Taylor expansion

$$0 = f'(\mathbf{x}^*) = f'(\mathbf{x}_k) + f''(\mathbf{x}_k)(\mathbf{x}^* - \mathbf{x}_k) + o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- After multiplying by $f''(\mathbf{x}_k)^{-1}$

$$\mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- Newton method iteration $\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$, therefore

$$\mathbf{x}_{k+1} - \mathbf{x}^* = o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

Local superlinear convergence

- Let $\mathbf{x}^*$ be local minimum, then

$$f'(\mathbf{x}^*) = 0, \quad f''(\mathbf{x}^*) \succ 0$$

- Taylor expansion

$$0 = f'(\mathbf{x}^*) = f'(\mathbf{x}_k) + f''(\mathbf{x}_k)(\mathbf{x}^* - \mathbf{x}_k) + o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- After multiplying by $f''(\mathbf{x}_k)^{-1}$

$$\mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- Newton method iteration $\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$, therefore

$$\mathbf{x}_{k+1} - \mathbf{x}^* = o(\|\mathbf{x}^* - \mathbf{x}_k\|)$$

- Local superlinear convergence $(\mathbf{x}_k \neq \mathbf{x}^*)$

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = \lim_{k\to\infty} \frac{o(\|\mathbf{x}_k - \mathbf{x}^*\|)}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$$

Local quadratic convergence

Theorem

Let

- $f(\mathbf{x})$ *be local strongly convex with constant* μ*:*
  $\exists \; \mathbf{x}^* : \; f''(\mathbf{x}^*) \succeq \mu \mathbf{I}$

Local quadratic convergence

Theorem

Let

- $f(\mathbf{x})$ *be local strongly convex with constant* μ*:*
  $\exists\ \mathbf{x}^* :\ f''(\mathbf{x}^*) \succeq \mu \mathbf{I}$
- *hessian is Lipschitz:* $\|f''(\mathbf{x}) - f''(\mathbf{y})\| \leq M\|\mathbf{x} - \mathbf{y}\|$

Local quadratic convergence

Theorem

Let

- $f(\mathbf{x})$ *be local strongly convex with constant* μ*:*
  $\exists\ \mathbf{x}^* :\ f''(\mathbf{x}^*) \succeq \mu\mathbf{I}$
- *hessian is Lipschitz:* $\|f''(\mathbf{x}) - f''(\mathbf{y})\| \leq M\|\mathbf{x} - \mathbf{y}\|$
- *initial guess* $\mathbf{x}_0$ *is sufficiently close to the solution* $\mathbf{x}^*$*:*
  $\|\mathbf{x}_0 - \mathbf{x}^*\| \leq \frac{2\mu}{3M}$

Local quadratic convergence

Theorem

Let

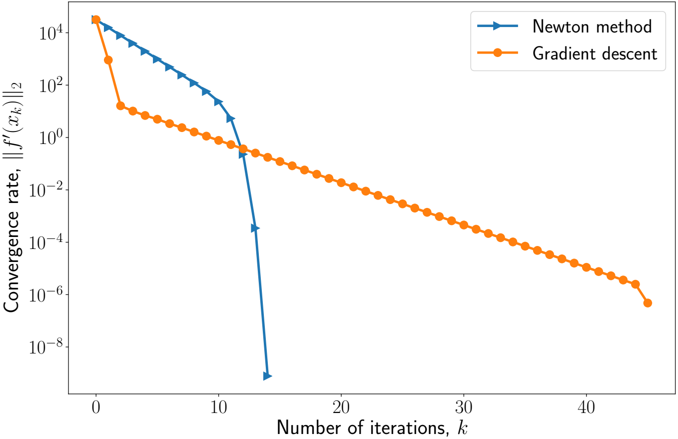
- $f(\mathbf{x})$ *be local strongly convex with constant* μ*:*
  $\exists\ \mathbf{x}^* :\ f''(\mathbf{x}^*) \succeq \mu \mathbf{I}$
- *hessian is Lipschitz:* $\|f''(\mathbf{x}) - f''(\mathbf{y})\| \leq M\|\mathbf{x} - \mathbf{y}\|$
- *initial guess* $\mathbf{x}_0$ *is sufficiently close to the solution* $\mathbf{x}^*$*:*
  $\|\mathbf{x}_0 - \mathbf{x}^*\| \leq \frac{2\mu}{3M}$

then Newton method converges ***quadratically***

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq \frac{M\|\mathbf{x}_k - \mathbf{x}^*\|^2}{2(\mu - M\|\mathbf{x}_k - \mathbf{x}^*\|)}$$

Example

$$
-\sum_{i=1}^{m} \log(1 - \mathbf{a}_i^\top \mathbf{x}) - \sum_{i=1}^{n} \log(1 - x_i^2) \to \min_{\mathbf{x} \in \mathbb{R}^n}
$$



Proof in 9 steps

Proof in 9 steps

1. $\mathbf{r}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = \mathbf{r}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$

Proof in 9 steps

1. $\mathbf{r}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = \mathbf{r}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$
2. The well-known fact from calculus

$$\phi(b) - \phi(a) = \int_0^1 \phi'(a + t(b-a))(b-a)dt$$

Proof in 9 steps

1. $\mathbf{r}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = \mathbf{r}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$
2. The well-known fact from calculus
$$\phi(b) - \phi(a) = \int_0^1 \phi'(a + t(b-a))(b-a)dt$$
3. If we use them for gradients
$$f'(\mathbf{x}_k) = f'(\mathbf{x}_k) - f'(\mathbf{x}^*) = \int_0^1 f''(\mathbf{x}^* + t\mathbf{r}_k)\mathbf{r}_k dt$$

Proof in 9 steps

1. $\mathbf{r}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = \mathbf{r}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$
2. The well-known fact from calculus

$$\phi(b) - \phi(a) = \int_0^1 \phi'(a + t(b-a))(b-a)dt$$

3. If we use them for gradients

$$f'(\mathbf{x}_k) = f'(\mathbf{x}_k) - f'(\mathbf{x}^*) = \int_0^1 f''(\mathbf{x}^* + t\mathbf{r}_k)\mathbf{r}_k dt$$

4. Substitute in the first step and derive

$$\mathbf{r}_{k+1} = \underbrace{\left(\mathbf{I} - f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}^* + t\mathbf{r}_k)]dt\right)}_{\mathbf{G}_k} \mathbf{r}_k$$

Proof in 9 steps

1. $\mathbf{r}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k) = \mathbf{r}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$
2. The well-known fact from calculus
$$\phi(b) - \phi(a) = \int_0^1 \phi'(a + t(b-a))(b-a)dt$$
3. If we use them for gradients
$$f'(\mathbf{x}_k) = f'(\mathbf{x}_k) - f'(\mathbf{x}^*) = \int_0^1 f''(\mathbf{x}^* + t\mathbf{r}_k)\mathbf{r}_k dt$$
4. Substitute in the first step and derive
$$\mathbf{r}_{k+1} = \underbrace{\left(\mathbf{I} - f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}^* + t\mathbf{r}_k)]dt\right)}_{\mathbf{G}_k} \mathbf{r}_k$$
5. $\|\mathbf{r}_{k+1}\| \leq \|\mathbf{G}_k\| \|\mathbf{r}_k\|$

6. Hessian is Lipschitz, then

$$\mathbf{G}_k = f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)]dt$$

$$\|\mathbf{G}_k\| \leq \|f''(\mathbf{x}_k)^{-1}\| \int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\|dt$$

6. Hessian is Lipschitz, then

$$\mathbf{G}_k = f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)] dt$$

$$\|\mathbf{G}_k\| \leq \|f''(\mathbf{x}_k)^{-1}\| \int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\| dt$$

7. Estimate the integral

$$\int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\| dt \leq \int_0^1 M\|\mathbf{r}_k - t\mathbf{r}_k\| dt = \frac{M\|\mathbf{r}_k\|}{2}$$

6. Hessian is Lipschitz, then

$$\mathbf{G}_k = f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)]dt$$

$$\|\mathbf{G}_k\| \leq \|f''(\mathbf{x}_k)^{-1}\| \int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\|dt$$

7. Estimate the integral

$$\int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\|dt \leq \int_0^1 M\|\mathbf{r}_k - t\mathbf{r}_k\|dt = \frac{M\|\mathbf{r}_k\|}{2}$$

8. From the Lipschitz hessian and strong convexity of f at $\mathbf{x}^*$ follows that

$$f''(\mathbf{x}_k) \succeq f''(\mathbf{x}^*) - M\|\mathbf{r}_k\|\mathbf{I} \succeq (\mu - M\|\mathbf{r}_k\|)\mathbf{I}$$

6. Hessian is Lipschitz, then

$$\mathbf{G}_k = f''(\mathbf{x}_k)^{-1} \int_0^1 [f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)]dt$$

$$\|\mathbf{G}_k\| \leq \|f''(\mathbf{x}_k)^{-1}\| \int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\|dt$$

7. Estimate the integral

$$\int_0^1 \|f''(\mathbf{x}_k) - f''(\mathbf{x}^* + t\mathbf{r}_k)\|dt \leq \int_0^1 M\|\mathbf{r}_k - t\mathbf{r}_k\|dt = \frac{M\|\mathbf{r}_k\|}{2}$$

8. From the Lipschitz hessian and strong convexity of f at $\mathbf{x}^*$ follows that

$$f''(\mathbf{x}_k) \succeq f''(\mathbf{x}^*) - M\|\mathbf{r}_k\|\mathbf{I} \succeq (\mu - M\|\mathbf{r}_k\|)\mathbf{I}$$

9. Estimate norm of the inverse hessian

$$\|f''(\mathbf{x}_k)^{-1}\| \leq \frac{1}{\mu - M\|\mathbf{r}_k\|}$$

Pro & Contra

Pro & Contra

Pro

- Quadratic convergence
- Extremely high accuracy of the solution

Pro & Contra

Pro

- Quadratic convergence
- Extremely high accuracy of the solution

Contra

- Storage of hessian: $O(n^2)$ memory cost
- Linear system is solved in every iteration: $O(n^3)$ operations in general case
- Hessian can be singular

What is similar in gradient descent and Newton method?

If objective is L-smooth, then

What is similar in gradient descent and Newton method?

If objective is L-smooth, then

- Gradient descent

$$
\begin{aligned}
& f(\mathbf{x}+\mathbf{h}) \leq f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h} \rangle + \frac{1}{2\alpha}\mathbf{h}^\top \mathbf{I}\mathbf{h} \equiv f_g(\mathbf{h}) \\
& \min_{\mathbf{h}} f_g(\mathbf{h}) \Rightarrow \mathbf{h}^* = -\alpha f'(\mathbf{x}) \\
& \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f'(\mathbf{x}_k)
\end{aligned}
$$

What is similar in gradient descent and Newton method?

If objective is L-smooth, then

- Gradient descent

$$f(\mathbf{x}+\mathbf{h}) \leq f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2\alpha}\mathbf{h}^\top \mathbf{I}\mathbf{h} \equiv f_g(\mathbf{h})$$

$$\min_{\mathbf{h}} f_g(\mathbf{h}) \Rightarrow \mathbf{h}^* = -\alpha f'(\mathbf{x})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f'(\mathbf{x}_k)$$

- Newton method

$$f(\mathbf{x}+\mathbf{h}) \approx f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h} \equiv f_N(\mathbf{h})$$

$$\min_{\mathbf{h}} f_N(\mathbf{h}) \Rightarrow \mathbf{h}^* = -(f''(\mathbf{x}))^{-1} f'(\mathbf{x})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$$

What is similar in gradient descent and Newton method?

If objective is L-smooth, then

- Gradient descent

$$f(\mathbf{x}+\mathbf{h}) \leq f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2\alpha}\mathbf{h}^\top \mathbf{I}\mathbf{h} \equiv f_g(\mathbf{h})$$

$$\min_{\mathbf{h}} f_g(\mathbf{h}) \Rightarrow \mathbf{h}^* = -\alpha f'(\mathbf{x})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f'(\mathbf{x}_k)$$

- Newton method

$$f(\mathbf{x}+\mathbf{h}) \approx f(\mathbf{x}) + \langle f'(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2}\mathbf{h}^\top f''(\mathbf{x})\mathbf{h} \equiv f_N(\mathbf{h})$$

$$\min_{\mathbf{h}} f_N(\mathbf{h}) \Rightarrow \mathbf{h}^* = -(f''(\mathbf{x}))^{-1} f'(\mathbf{x})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)$$

- Something better than $f_g(\mathbf{x})$, but faster than $f_N(\mathbf{x})$?

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Requirement to the hessian estimate $\mathbf{B}_k$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Requirement to the hessian estimate $\mathbf{B}_k$

- Fast update $\mathbf{B}_k \to \mathbf{B}_{k+1}$, **that uses only gradient**

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Requirement to the hessian estimate $\mathbf{B}_k$

- Fast update $\mathbf{B}_k \to \mathbf{B}_{k+1}$, **that uses only gradient**
- Fast search of $\mathbf{h}_k$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h}\rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Requirement to the hessian estimate $\mathbf{B}_k$

- Fast update $\mathbf{B}_k \to \mathbf{B}_{k+1}$, **that uses only gradient**
- Fast search of $\mathbf{h}_k$
- Efficient storing $\mathbf{B}_k$

Quasi-Newton method

- Quadratic estimate $f(\mathbf{x}_{k+1})$

$$f_q(\mathbf{h}) = f(\mathbf{x}_k) + \langle f'(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\mathbf{h}^\top \mathbf{B}_k \mathbf{h}, \quad \mathbf{B}_k \succ 0$$

- Minimum of $f_q(\mathbf{h})$ is at point

$$\mathbf{h}_k = -\mathbf{B}_k^{-1} f'(\mathbf{x}_k)$$

- Quasi-Newton method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} f'(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{H}_k f'(\mathbf{x}_k)$$

Requirement to the hessian estimate $\mathbf{B}_k$

- Fast update $\mathbf{B}_k \to \mathbf{B}_{k+1}$, **that uses only gradient**
- Fast search of $\mathbf{h}_k$
- Efficient storing $\mathbf{B}_k$
- Superlinear convergence

How update $\mathbf{B}_k$?

Two gradients rule

- $f'_q(-\alpha_k \mathbf{h}_k) = f'(\mathbf{x}_k) \Rightarrow f'(\mathbf{x}_{k+1}) - \alpha_k \mathbf{B}_{k+1} \mathbf{h}_k = f'(\mathbf{x}_k)$
- $f'_q(0) = f'(\mathbf{x}_{k+1})$ is true by construction

How update $\mathbf{B}_k$?

Two gradients rule

- $f'_q(-\alpha_k \mathbf{h}_k) = f'(\mathbf{x}_k) \Rightarrow f'(\mathbf{x}_{k+1}) - \alpha_k \mathbf{B}_{k+1} \mathbf{h}_k = f'(\mathbf{x}_k)$
- $f'_q(0) = f'(\mathbf{x}_{k+1})$ is true by construction

Secant equation

- $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$
- $\mathbf{y}_k = f'(\mathbf{x}_{k+1}) - f'(\mathbf{x}_k)$

$$\mathbf{B}_{k+1} \mathbf{s}_k = \mathbf{y}_k,$$

How update $\mathbf{B}_k$?

Two gradients rule

- $f'_q(-\alpha_k \mathbf{h}_k) = f'(\mathbf{x}_k) \Rightarrow f'(\mathbf{x}_{k+1}) - \alpha_k \mathbf{B}_{k+1}\mathbf{h}_k = f'(\mathbf{x}_k)$
- $f'_q(0) = f'(\mathbf{x}_{k+1})$ is true by construction

Secant equation

- $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$
- $\mathbf{y}_k = f'(\mathbf{x}_{k+1}) - f'(\mathbf{x}_k)$

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k,$$

- New estimate of hessian has to be close to the current

Parameters

- Initialization of $\mathbf{B}_0$

Parameters

- Initialization of $\mathbf{B}_0$
- No operations which cost $O(n^3)$

Parameters

- Initialization of $\mathbf{B}_0$
- No operations which cost $O(n^3)$

Parameters

- Initialization of $\mathbf{B}_0$
- No operations which cost $O(n^3)$

Examples of quasi-Newton methods

- Barzilai-Borwein
- BFGS

Barzilai-Borwein method

- Approximate hessian by diagonal matrix:

$$\alpha_k f'(\mathbf{x}_k) = \alpha_k \mathbf{I} f'(\mathbf{x}_k) = \left(\frac{1}{\alpha_k}\mathbf{I}\right)^{-1} f'(\mathbf{x}_k) \approx (f''(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)$$

Barzilai-Borwein method

- Approximate hessian by diagonal matrix:

$$\alpha_k f'(\mathbf{x}_k) = \alpha_k \mathbf{I} f'(\mathbf{x}_k) = \left(\frac{1}{\alpha_k}\mathbf{I}\right)^{-1} f'(\mathbf{x}_k) \approx (f''(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)$$

- Secant equation

$$\alpha_k^{-1}\mathbf{s}_{k-1} \approx \mathbf{y}_{k-1}$$

Barzilai-Borwein method

- Approximate hessian by diagonal matrix:

$$\alpha_k f'(\mathbf{x}_k) = \alpha_k \mathbf{I} f'(\mathbf{x}_k) = \left(\frac{1}{\alpha_k}\mathbf{I}\right)^{-1} f'(\mathbf{x}_k) \approx (f''(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)$$

- Secant equation

$$\alpha_k^{-1}\mathbf{s}_{k-1} \approx \mathbf{y}_{k-1}$$

- Auxiliary problem and analytic solution

$$\min_{\alpha_k} \|\mathbf{s}_{k-1} - \alpha_k \mathbf{y}_{k-1}\|_2 \Rightarrow \alpha_k = \frac{\mathbf{s}_{k-1}^\top \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^\top \mathbf{y}_{k-1}}$$

Barzilai-Borwein method

- Approximate hessian by diagonal matrix:

$$\alpha_k f'(\mathbf{x}_k) = \alpha_k \mathbf{I} f'(\mathbf{x}_k) = \left(\frac{1}{\alpha_k}\mathbf{I}\right)^{-1} f'(\mathbf{x}_k) \approx (f''(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)$$

- Secant equation

$$\alpha_k^{-1}\mathbf{s}_{k-1} \approx \mathbf{y}_{k-1}$$

- Auxiliary problem and analytic solution

$$\min_{\alpha_k} \|\mathbf{s}_{k-1} - \alpha_k \mathbf{y}_{k-1}\|_2 \Rightarrow \alpha_k = \frac{\mathbf{s}_{k-1}^\top \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^\top \mathbf{y}_{k-1}}$$

- It has stochastic modifications, paper from NIPS 2016

BFGS method

Broyden, Fletcher, Goldfarb, Shanno

BFGS method

- The problem

$$\begin{aligned} &\min_{\mathbf{H}} \|\mathbf{H}_k - \mathbf{H}\| \\ \text{s.t. } &\mathbf{H} = \mathbf{H}^\top \\ &\mathbf{H}\mathbf{y}_k = \mathbf{s}_k \end{aligned}$$

BFGS method

- The problem

$$
\begin{aligned}
&\min_{\mathbf{H}} \|\mathbf{H}_k - \mathbf{H}\| \\
\text{s.t. } &\mathbf{H} = \mathbf{H}^\top \\
&\mathbf{H}\mathbf{y}_k = \mathbf{s}_k
\end{aligned}
$$

- The solution

$$
\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^\top)\mathbf{H}_k(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^\top) + \rho_k \mathbf{s}_k \mathbf{s}_k^\top,
$$

where $\rho_k = \dfrac{1}{\mathbf{y}_k^\top \mathbf{s}_k}$

BFGS method

- The problem

$$\begin{aligned} &\min_{\mathbf{H}} \|\mathbf{H}_k - \mathbf{H}\| \\ \text{s.t. } &\mathbf{H} = \mathbf{H}^\top \\ &\mathbf{H}\mathbf{y}_k = \mathbf{s}_k \end{aligned}$$

- The solution

$$\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^\top)\mathbf{H}_k(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^\top) + \rho_k \mathbf{s}_k \mathbf{s}_k^\top,$$

where $\rho_k = \dfrac{1}{\mathbf{y}_k^\top \mathbf{s}_k}$

Almost theorem

If f is strongly convex and the hessian is Lipschitz. The BFGS converges superlinearly under some mild technical assumptions.

Main features of BFGS

- Good performance in practice

Main features of BFGS

- Good performance in practice
- Self-correction property

Limited memory quasi-Newton methods

- Complexity of storage and update of hessian is $O(n^2)$

Limited memory quasi-Newton methods

- Complexity of storage and update of hessian is $O(n^2)$
- But we need not the matrix but the result of matrix by vector $f'(\mathbf{x})$ product

Limited memory quasi-Newton methods

- Complexity of storage and update of hessian is $O(n^2)$
- But we need not the matrix but the result of matrix by vector $f'(\mathbf{x})$ product
- Vectors $\mathbf{y}$ and $\mathbf{s}$ from fist iterations can suffer estimate of $\mathbf{B}$ or $\mathbf{H}$ in later iterations

Limited memory quasi-Newton methods

- Complexity of storage and update of hessian is $O(n^2)$
- But we need not the matrix but the result of matrix by vector $f'(\mathbf{x})$ product
- Vectors $\mathbf{y}$ and $\mathbf{s}$ from fist iterations can suffer estimate of $\mathbf{B}$ or $\mathbf{H}$ in later iterations

Idea

Use only the last $m \ll n$ pairs $(\mathbf{s}, \mathbf{y})$ and update $\mathbf{H}_{m,0}$ in every iteration

Limited memory quasi-Newton methods

- Complexity of storage and update of hessian is $O(n^2)$
- But we need not the matrix but the result of matrix by vector $f'(\mathbf{x})$ product
- Vectors $\mathbf{y}$ and $\mathbf{s}$ from fist iterations can suffer estimate of $\mathbf{B}$ or $\mathbf{H}$ in later iterations

Idea

Use only the last $m \ll n$ pairs $(\mathbf{s}, \mathbf{y})$ and update $\mathbf{H}_{m,0}$ in every iteration

- Complexity is $O(mn)$

L-BFGS method

- One of the best method in practice

L-BFGS method

- One of the best method in practice
- Size of history m has to be pre-defined

L-BFGS method

- One of the best method in practice
- Size of history m has to be pre-defined
- BFGS updates H recursively

$$\mathbf{H}_{k+1} = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^\top, \quad \mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^\top$$

L-BFGS method

- One of the best method in practice
- Size of history m has to be pre-defined
- BFGS updates H recursively

$$\mathbf{H}_{k+1} = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^\top, \quad \mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^\top$$

- Unroll m steps of recursion

$$\begin{aligned}
&\mathbf{H}_{k+1} = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^\top \\
&= \mathbf{V}_k^\top \mathbf{V}_{k-1}^\top \mathbf{H}_{k-1} \mathbf{V}_{k-1} \mathbf{V}_k + \rho_{k-1} \mathbf{V}_k^\top \mathbf{V}_{k-1}^\top \mathbf{s}_{k-1} \mathbf{s}_{k-1}^\top \mathbf{V}_{k-1} \mathbf{V}_k \\
&+ \rho_k \mathbf{s}_k \mathbf{s}_k^\top = \mathbf{V}_k^\top \ldots \mathbf{V}_{k-m+1}^\top \mathbf{H}_{m,0} \mathbf{V}_{k-m+1} \ldots \mathbf{V}_k \\
&+ \rho_{k-m+1} \mathbf{V}_k^\top \ldots \mathbf{V}_{k-m+2}^\top \mathbf{s}_{k-m+1} \mathbf{s}_{k-m+1}^\top \mathbf{V}_{k-m+2} \ldots V_k \\
&+ \ldots + \rho_k \mathbf{s}_k \mathbf{s}_k^\top
\end{aligned}$$

L-BFGS method

- One of the best method in practice
- Size of history m has to be pre-defined
- BFGS updates H recursively

$$\mathbf{H}_{k+1} = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^\top, \quad \mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^\top$$

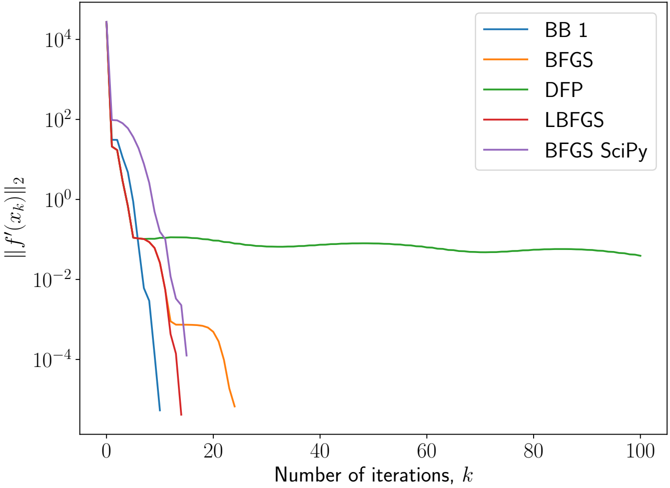
- Unroll m steps of recursion

$$\begin{aligned}
&\mathbf{H}_{k+1} = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^\top \\
&= \mathbf{V}_k^\top \mathbf{V}_{k-1}^\top \mathbf{H}_{k-1} \mathbf{V}_{k-1} \mathbf{V}_k + \rho_{k-1} \mathbf{V}_k^\top \mathbf{V}_{k-1}^\top \mathbf{s}_{k-1} \mathbf{s}_{k-1}^\top \mathbf{V}_{k-1} \mathbf{V}_k \\
&+ \rho_k \mathbf{s}_k \mathbf{s}_k^\top = \mathbf{V}_k^\top \ldots \mathbf{V}_{k-m+1}^\top \mathbf{H}_{m,0} \mathbf{V}_{k-m+1} \ldots \mathbf{V}_k \\
&+ \rho_{k-m+1} \mathbf{V}_k^\top \ldots \mathbf{V}_{k-m+2}^\top \mathbf{s}_{k-m+1} \mathbf{s}_{k-m+1}^\top \mathbf{V}_{k-m+2} \ldots V_k \\
&+ \ldots + \rho_k \mathbf{s}_k \mathbf{s}_k^\top
\end{aligned}$$

- Efficient computing of $\mathbf{H}_k f'(\mathbf{x})$ without explicit forming of $\mathbf{H}_k$

Example

$$
-\sum_{i=1}^{m} \log(1 - \mathbf{a}_i^\top \mathbf{x}) - \sum_{i=1}^{n} \log(1 - x_i^2) \to \min_{\mathbf{x} \in \mathbb{R}^n}
$$



Pro & Contra

Pro & Contra

Pro

- One iteration complexity is $O(n^2) + ...$ in contrast to the Newton method $O(n^3) + ...$
- L-BFGS requires linear amount of memory w.r.t. to the dimension of the problem
- Self-correction property of BFGS
- Superlinear convergence

Pro & Contra

Pro

- One iteration complexity is $O(n^2) + \ldots$ in contrast to the Newton method $O(n^3) + \ldots$
- L-BFGS requires linear amount of memory w.r.t. to the dimension of the problem
- Self-correction property of BFGS
- Superlinear convergence

Contra

- Stochastic generalization does not work
- Initialization of $\mathbf{B}_0$ or $\mathbf{H}_0$
- Convergence theory is still under development