CS450: Numerical Analysis Optimization

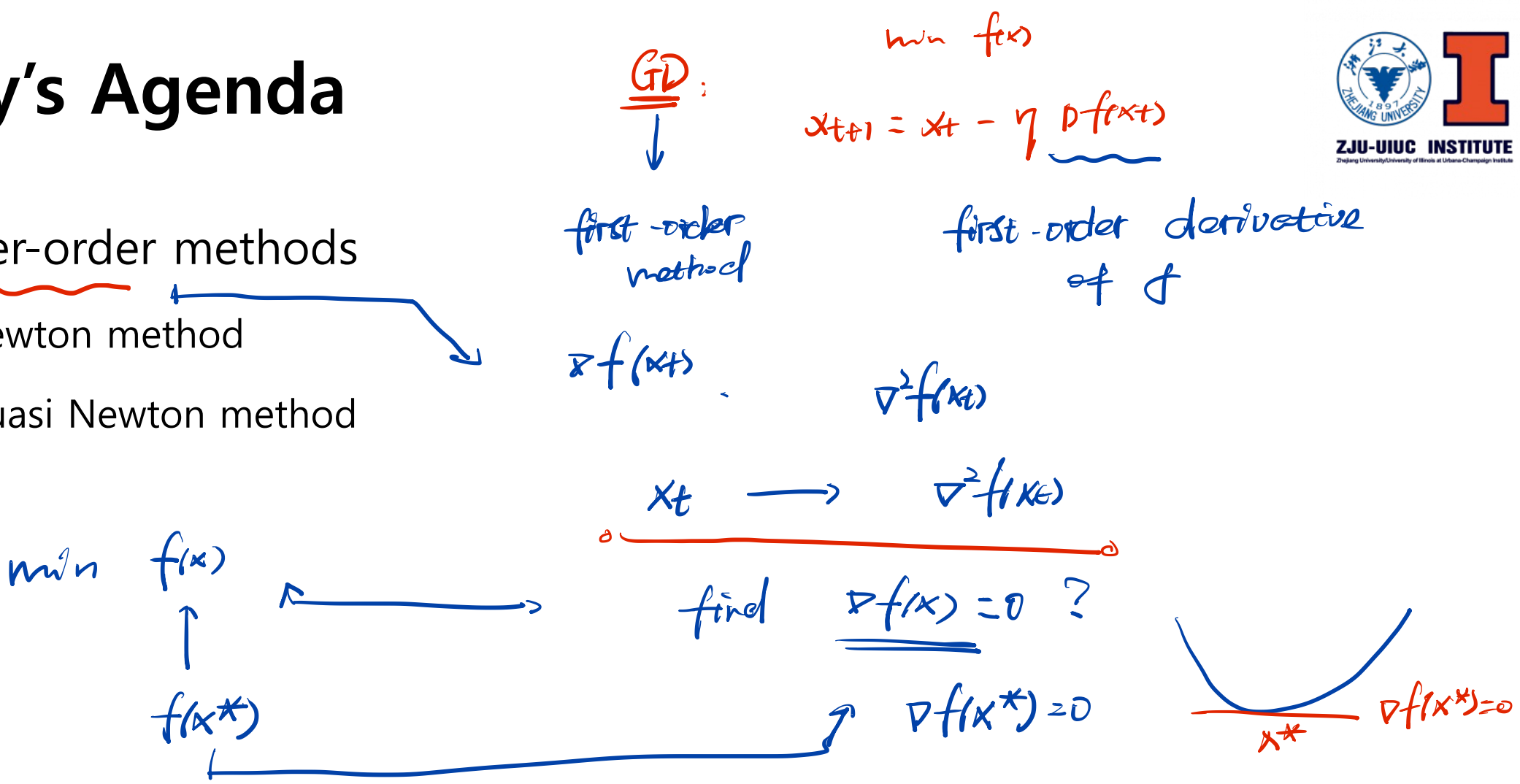
Howard Hao Yang

Assistant Professor, ZJU-UIUC Institute

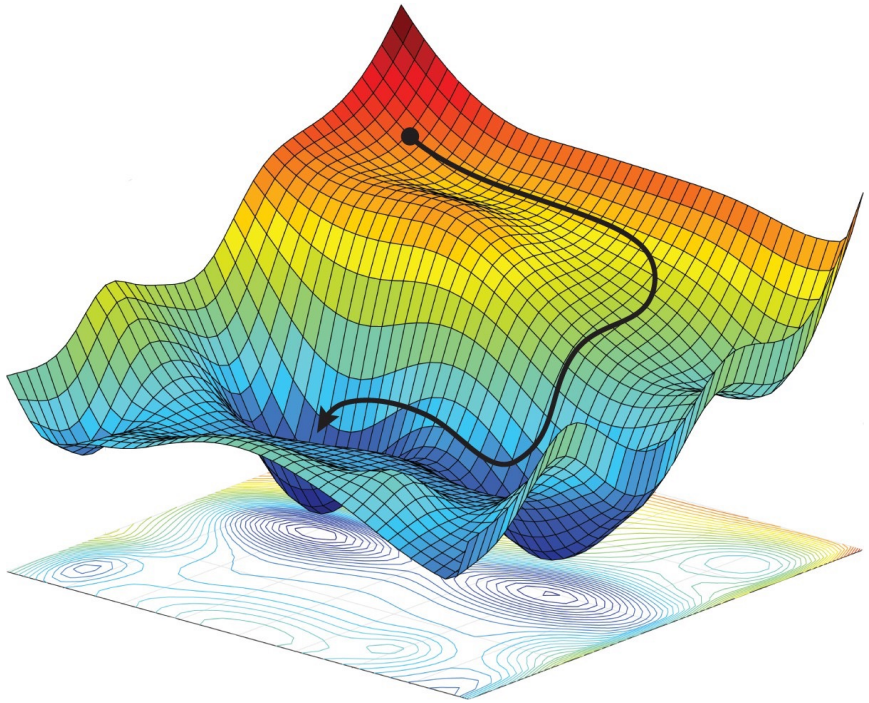
15/12/2023

Today's Agenda

- Higher-order methods
  - Newton method
  - Quasi Newton method



Optimization



- The goal is to accomplish the following optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in R^n$$

Optimality Conditions

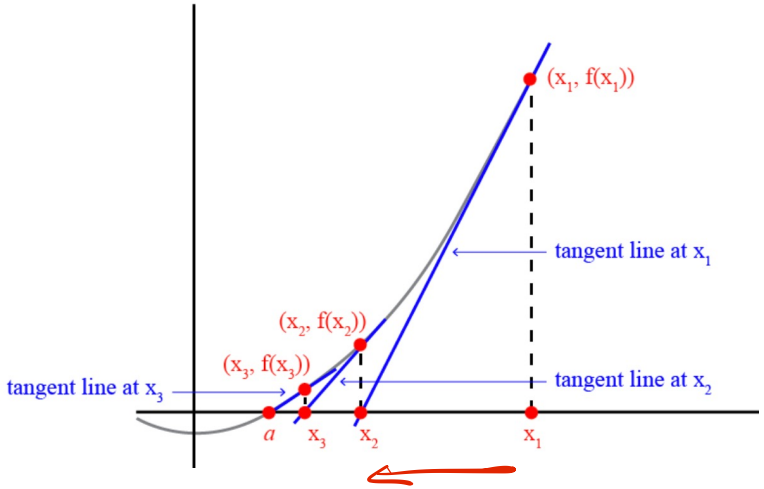
- Suppose we found a candidate $\boldsymbol{x}^*$, how do we know it actually is (the) one?
- n-dimension case:
  - Necessary: $\nabla f(\boldsymbol{x}^*) = 0$ (i.e., $\boldsymbol{x}^*$ is an extremal point)
  - Sufficient: $\nabla f(\boldsymbol{x}^*) = 0$ and $H_f(\boldsymbol{x}^*)$ is positive semidefinite (i.e., $\boldsymbol{x}^*$ is a local minimum), where $H_f(\boldsymbol{x}^*)$ denotes the Hessian matrix

Optimality Conditions

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- Suppose we found a candidate $\boldsymbol{x}^*$, how do we know it actually is (the) one?
- n-dimension case:
  - Necessary: $\nabla f(\boldsymbol{x}^*) = 0$ (i.e., $\boldsymbol{x}^*$ is an extremal point)
  - Sufficient: $\nabla f(\boldsymbol{x}^*) = 0$ and $H_f(\boldsymbol{x}^*)$ is positive semidefinite (i.e., $\boldsymbol{x}^*$ is a local minimum), where $H_f(\boldsymbol{x}^*)$ denotes the Hessian matrix
- Can we convert the optimization problem into solving nonlinear equation systems as $\nabla f(\boldsymbol{x}) = \mathbf{0}$, and then leverage techniques for root finding in nonlinear systems to solve for it?

Recall: Newton's Method

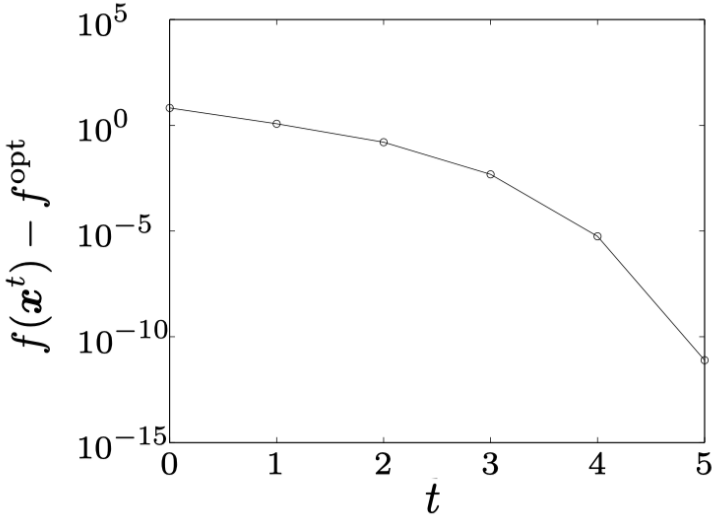


- To find the root of $f(x) = 0$, algorithm details:

  1) start with an initial guess x_0

  2) keep iterating $x_{k+1} = g(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}$

Newton's Method in Optimization



Good: fast conv.

Bad:

① $\nabla^2 f(x_t)$ $\simeq O(n^2)$

② $(\nabla^2 f(x_t))^{-1}$ $\approx O(n^3)$

- Given an unconstrained optimization problem,

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} \in R^n$$

- Newton's method runs as (solving $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$)
  - Start with an initial point $\boldsymbol{x}_0$ and update as follows

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \left(\nabla^2 f(\boldsymbol{x}_t)\right)^{-1} \nabla f(\boldsymbol{x}_t)$$

Newton's Method in Optimization (Cont'd)

ZJU-UIUC INSTITUTE

- Quadratic convergence: attains ϵ accuracy within $O(\log\log\frac{1}{\epsilon})$ iterations
  - If the objective is quadratic, e.g., the following, Newton's method solves it in one iteration (why?)

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^T\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^*)$$

- Issues:
  - Need to compute a Hessian matrix $\nabla^2 f(\boldsymbol{x}_t)$ at every iteration
  - Need to invert the Hessian matrix $\nabla^2 f(\boldsymbol{x}_t)$ at every iteration
- Each of these two steps can be generally expensive

Quasi Newton Methods (1)

- Compare the following:

Gradient descent:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \cdot \boldsymbol{I} \cdot \nabla f(\boldsymbol{x}_t)$$

Quasi Newton method:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \cdot \boldsymbol{H}_t \cdot \nabla f(\boldsymbol{x}_t)$$

Newton method:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \cdot \left(\nabla^2 f(\boldsymbol{x}_t)\right)^{-1} \cdot \nabla f(\boldsymbol{x}_t)$$

Quasi Newton Methods (2)

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

$$\begin{cases} v_{t+1} = v_t + (\nabla f(x_t))^2 \\ x_{t+1} = x_t - \eta \dfrac{\nabla f(x_t)}{\sqrt{v_t}} \end{cases}$$

- Questions:
  1) How are adaptive optimization methods (RMSprop, AdaGrad, Adam) linked to the previous form?

$$x_{t+1} = x_t - \eta_t \cdot \begin{bmatrix} \frac{1}{\sqrt{v_{1,t}}} & & 0 \\ & \frac{1}{\sqrt{v_{2,t}}} & \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{v_{n,t}}} \end{bmatrix} \nabla f(x_t)$$

  2) Can we do something similar?

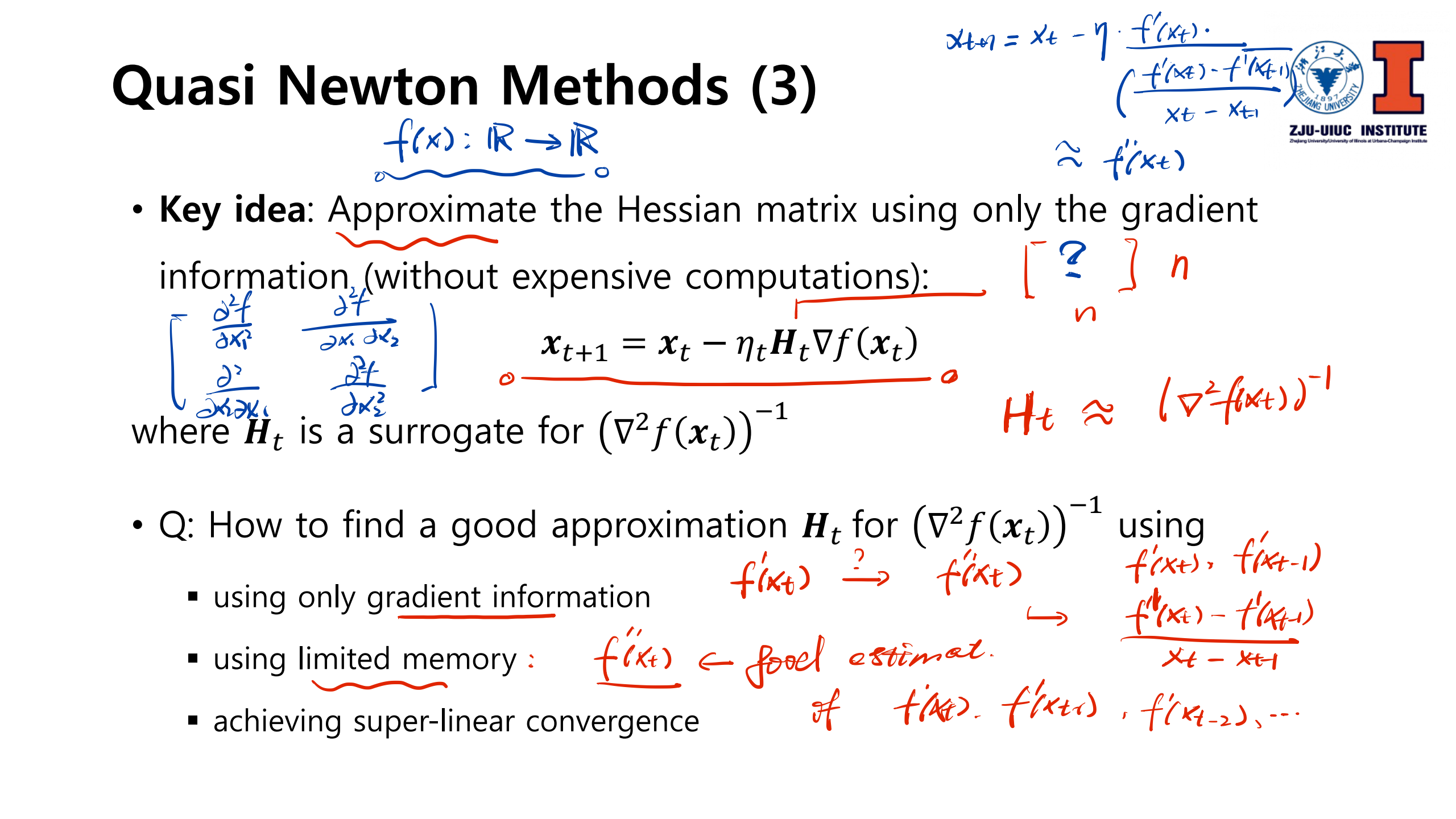
Quasi Newton Methods (3)

- **Key idea**: Approximate the Hessian matrix using only the gradient information (without expensive computations):

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{H}_t \nabla f(\boldsymbol{x}_t)$$

where $\boldsymbol{H}_t$ is a surrogate for $\left(\nabla^2 f(\boldsymbol{x}_t)\right)^{-1}$

- Q: How to find a good approximation $\boldsymbol{H}_t$ for $\left(\nabla^2 f(\boldsymbol{x}_t)\right)^{-1}$ using
  - using only gradient information
  - using limited memory
  - achieving super-linear convergence

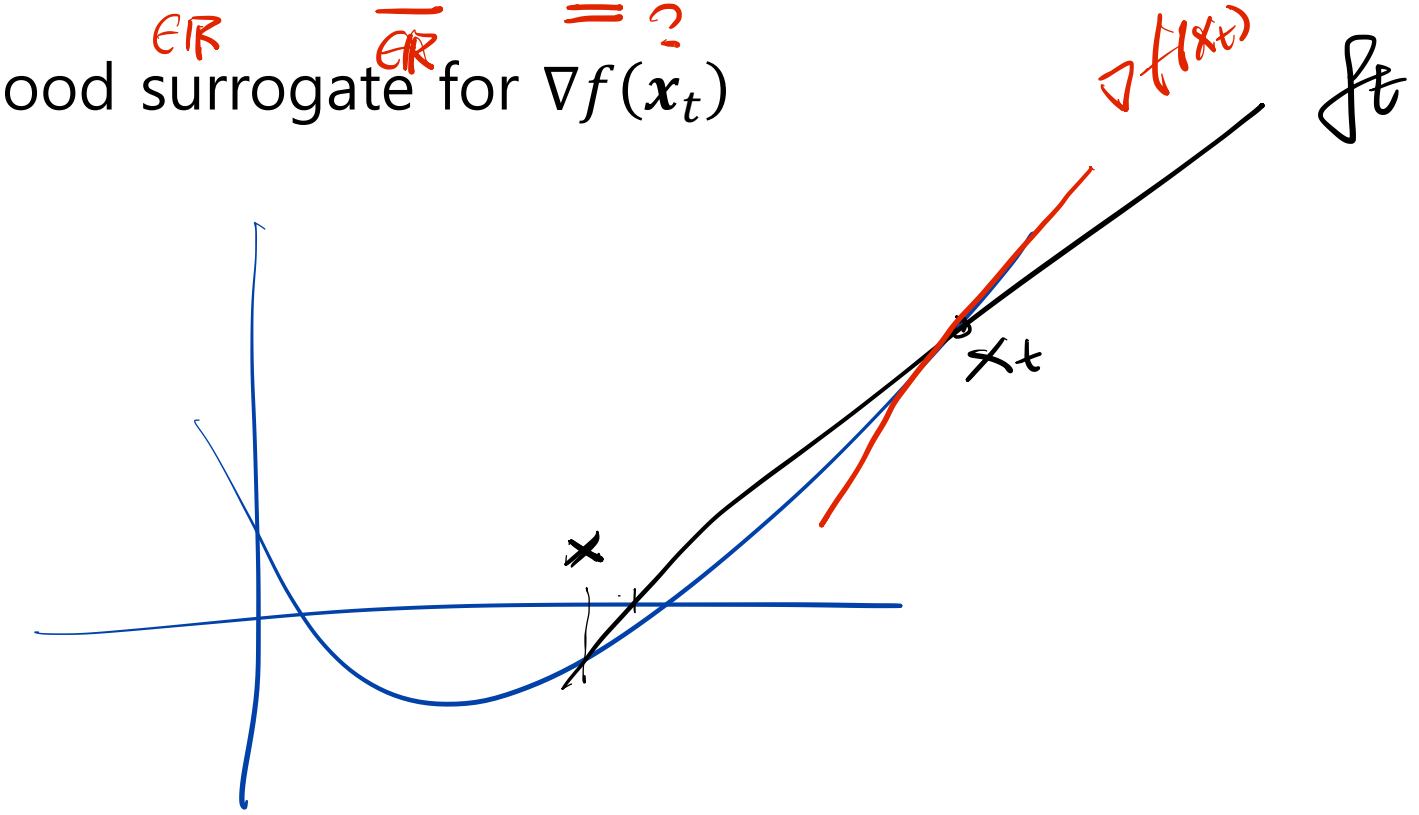


Approximating Hessian

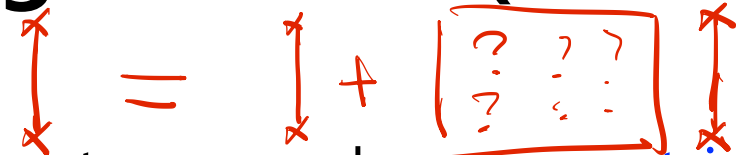
- Approximating gradient: if $\boldsymbol{x}$ is close to $\boldsymbol{x}_t$, some vector $\boldsymbol{g}_t$ satisfies

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_t) + \boldsymbol{g}_t^T(\boldsymbol{x} - \boldsymbol{x}_t)$$

then $\boldsymbol{g}_t$ can be a good surrogate for $\nabla f(\boldsymbol{x}_t)$



Approximating Hessian (Cont'd)



- Likewise, if $\boldsymbol{x}$ is close to $\boldsymbol{x}_t$, and some matrix $\boldsymbol{G}_{t+1}$ satisfies

$$\nabla f(\boldsymbol{x}) = \nabla f(\boldsymbol{x}_{t+1}) + \boldsymbol{G}_{t+1}^T(\boldsymbol{x} - \boldsymbol{x}_{t+1})$$

then $\boldsymbol{G}_{t+1}$ can be a good surrogate for $\nabla^2 f(\boldsymbol{x}_{t+1})$

- By using information from previous iterations, the following holds

$$\nabla f(\boldsymbol{x}_t) = \nabla f(\boldsymbol{x}_{t+1}) + \boldsymbol{H}_{t+1}^{-1}(\boldsymbol{x}_t - \boldsymbol{x}_{t+1})$$

  - $\boldsymbol{H}_{t+1}^{-1}$ can be regarded as $\boldsymbol{G}_{t+1}^{-1}$ evaluated at $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$

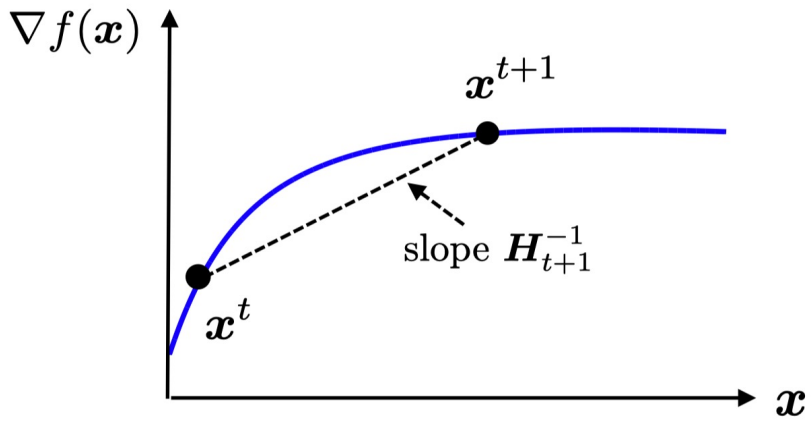
$$H_{t+1}^{-1} \approx \frac{\nabla f(x_t) - \nabla f(x_{t+1})}{x_t - x_{t+1}} \approx \nabla^2 f(x_{t+1})$$

$\{ x_t, \nabla f(x_t), x_{t+1}, \nabla f(x_{t+1}) \}$

$\in \mathbb{R}^n \quad \in \mathbb{R}^n \quad \in \mathbb{R}^n \quad \in \mathbb{R}^n$

(*) $H_{t+1}^{-1} \approx \nabla^2 f(x_{t+1})$

Approximating Hessian (Cont'd)



- Rearranging terms, arrive at the following (which is termed the secant equation)

$$\boldsymbol{H}_{t+1}^{-1}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) = \nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)$$

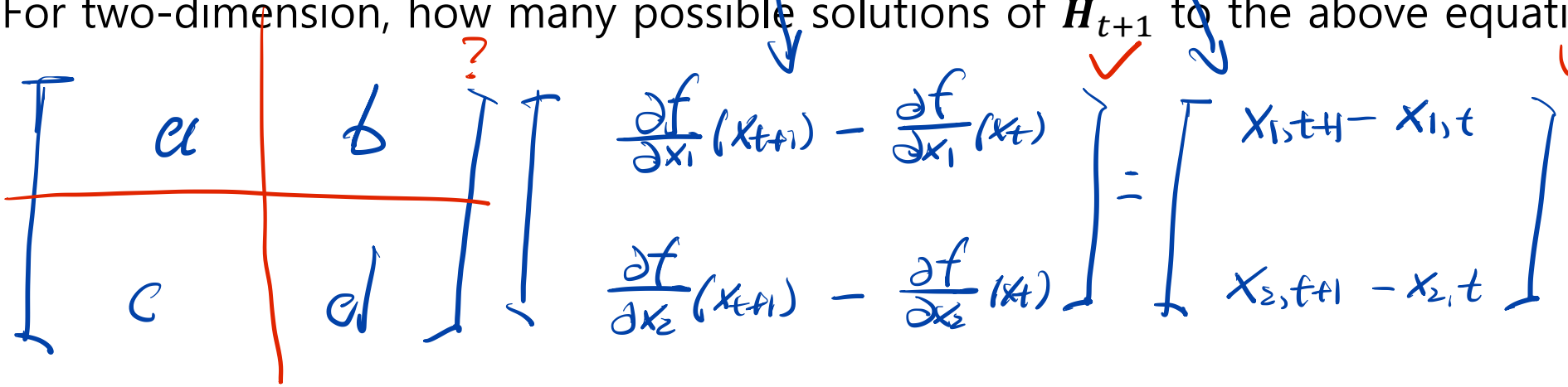
- Requires $\boldsymbol{H}_{t+1}^{-1}$ to map the displacement $\boldsymbol{x}_{t+1} - \boldsymbol{x}_t$ into the changes of $\nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)$

Secant Equation (1)

- The secant equation

$$\boldsymbol{H}_{t+1}\left(\nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)\right) = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t$$

  - For one-dimension, how many possible solutions of $\boldsymbol{H}_{t+1}$ to the above equation?
  - For two-dimension, how many possible solutions of $\boldsymbol{H}_{t+1}$ to the above equation?



- Admits infinitely many solutions, since degrees of freedom $O(n^2)$ in choosing $\boldsymbol{H}_{t+1}$ far exceeds number of constraints n

Secant Equation (2)

- The secant equation

$$\boldsymbol{H}_{t+1}\left(\nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)\right) = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \quad (*)$$

- Admits infinitely many solutions, since degrees of freedom $O(n^2)$ in choosing $\boldsymbol{H}_{t+1}$ far exceeds number of constraints n
- Which $\boldsymbol{H}_{t+1}$ shall we choose? (solution to (*))
  1. We want $\boldsymbol{H}_{t+1}$ to be symmetric (why?) : $H_{t+1} = H_{t+1}^T$
  2. We want $\boldsymbol{H}_{t+1}$ to be "close" to $\boldsymbol{H}_t$: $\min_H \|H - H_t\|$
  3. If $\boldsymbol{H}_t$ is positive definite, we want $\boldsymbol{H}_{t+1}$ to also be positive definite

$w^T \nabla^2 f(x_t)\, w \geq 0$

Formalizing the Selection

- Chose an update by minimizing the closeness to $\boldsymbol{H}_t$

$$\min_{\boldsymbol{H}} \|\boldsymbol{H} - \boldsymbol{H}_t\|$$

$$\text{s.t.} \quad \boldsymbol{H} = \boldsymbol{H}^T \quad \text{..... (symmetric condition)}$$

$$\boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{s}_t \quad \text{..... (secant equation)}$$

where $\boldsymbol{y}_t = \nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)$ and $\boldsymbol{s}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t$

- Exploit past information regarding $\boldsymbol{H}_t$
- Each choice of norm gives a different $\boldsymbol{H}_{t+1}$
- How to solve this?

Formalizing the Selection (Cont'd)

- The secant equation stipulates the following

$$\boldsymbol{H}_{t+1}^{-1}\boldsymbol{s}_t = \boldsymbol{y}_t$$

- Suppose we already have $\boldsymbol{H}_t^{-1}$ and update $\boldsymbol{H}_{t+1}^{-1}$ as follows (why?)

$$\boldsymbol{H}_{t+1}^{-1} = \boldsymbol{H}_t^{-1} + a\boldsymbol{u}\boldsymbol{u}^T + b\boldsymbol{v}\boldsymbol{v}^T$$

- Substitute into the equation above

$$(H_t^{-1} + auu^T + bvv^T)\,s_t = y_t$$

Formalizing the Selection (Cont'd)

- Substitute into the equation above gives

$$\boldsymbol{y}_t - \boldsymbol{H}_t^{-1}\boldsymbol{s}_t = a\boldsymbol{u}(\boldsymbol{u}^T\boldsymbol{s}_t) + b\boldsymbol{v}(\boldsymbol{v}^T\boldsymbol{s}_t)$$

setting $\boldsymbol{u} = \boldsymbol{y}_t$ and $\boldsymbol{v} = \boldsymbol{H}_t^{-1}\boldsymbol{s}_t$ allows us to solve for a and b

Formalizing the Selection (Cont'd)

ZJU-UIUC INSTITUTE

- The secant equation stipulates the following

$$\boldsymbol{H}_{t+1}^{-1}\boldsymbol{s}_t = \boldsymbol{y}_t$$

- Suppose we already have $\boldsymbol{H}_t^{-1}$ and update $\boldsymbol{H}_{t+1}^{-1}$ as follows (why?)

$$\boldsymbol{H}_{t+1}^{-1} = \boldsymbol{H}_t^{-1} + a\boldsymbol{u}\boldsymbol{u}^T + b\boldsymbol{v}\boldsymbol{v}^T$$

- Substitute into the equation above gives

$$\boldsymbol{y}_t - \boldsymbol{H}_t^{-1}\boldsymbol{s}_t = a\boldsymbol{u}(\boldsymbol{u}^T\boldsymbol{s}_t) + b\boldsymbol{v}(\boldsymbol{v}^T\boldsymbol{s}_t)$$

setting $\boldsymbol{u} = \boldsymbol{y}_t$ and $\boldsymbol{v} = \boldsymbol{H}_t^{-1}\boldsymbol{s}_t$ allows us to solve for a and b, given as

$$a = \frac{1}{\boldsymbol{y}_t{}^T\boldsymbol{s}_t}, \qquad b = -\frac{1}{\boldsymbol{s}_t{}^T\boldsymbol{H}_t^{-1}\boldsymbol{s}_t}$$

Formalizing the Selection (Cont'd)

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- The update of $\boldsymbol{H}_{t+1}^{-1}$, based on $\boldsymbol{H}_t^{-1}$ can then be formally written as

$$\boldsymbol{H}_{t+1}^{-1} = \boldsymbol{H}_t^{-1} + \frac{\boldsymbol{y}_t \boldsymbol{y}_t^T}{\boldsymbol{y}_t^T \boldsymbol{s}_t} - \frac{\boldsymbol{H}_t^{-1} \boldsymbol{s}_t \boldsymbol{s}_t^T \boldsymbol{H}_t^{-1}}{\boldsymbol{s}_t^T \boldsymbol{H}_t^{-1} \boldsymbol{s}_t} \qquad (*)$$

- The updating is rank-2 and hence can be executed efficiently
- This updating is known as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

BFGS

Formalizing the Selection (Cont'd)

- We can now update $\boldsymbol{H}_{t+1}^{-1}$ based on $\boldsymbol{H}_t^{-1}$

$$\boldsymbol{H}_{t+1}^{-1} = \boldsymbol{H}_t^{-1} + \frac{\boldsymbol{y}_t \boldsymbol{y}_t^T}{\boldsymbol{y}_t^T \boldsymbol{s}_t} - \frac{\boldsymbol{H}_t^{-1} \boldsymbol{s}_t \boldsymbol{s}_t^T \boldsymbol{H}_t^{-1}}{\boldsymbol{s}_t^T \boldsymbol{H}_t^{-1} \boldsymbol{s}_t} \qquad (*)$$

- But the algorithm updates using the following form

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{H}_t \nabla f(\boldsymbol{x}_t)$$

- We need $\boldsymbol{H}_t$ rather than $\boldsymbol{H}_t^{-1}$
  - Q: can we construct a similar update formula?

Formalizing the Selection (Cont'd)

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- Sherman-Morrison-Woodbury formula

$$(\boldsymbol{A}+\boldsymbol{UDV})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{D}^{-1}+\boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}$$

- Homework: By setting

$$\boldsymbol{U}=\boldsymbol{V}^{T}=[\boldsymbol{H}_{t}^{-1}\boldsymbol{s}_{t}\ \ \boldsymbol{y}_{t}]$$

$$\boldsymbol{D}=\begin{bmatrix}-1/(\boldsymbol{s}_{t}{}^{T}\boldsymbol{H}_{t}^{-1}\boldsymbol{s}_{t}) & \boldsymbol{0}\\ \boldsymbol{0} & 1/(\boldsymbol{y}_{t}{}^{T}\boldsymbol{s}_{t})\end{bmatrix}$$

- The BFGS is equivalent to the following

$$\boldsymbol{H}_{t+1}=(\boldsymbol{I}-\rho_{t}\boldsymbol{s}_{t}\boldsymbol{y}_{t}^{T})\boldsymbol{H}_{t}(\boldsymbol{I}-\rho_{t}\boldsymbol{y}_{t}\boldsymbol{s}_{t}^{T})+\rho_{t}\boldsymbol{s}_{t}\boldsymbol{s}_{t}^{T}$$

where $\rho_{t}=\frac{1}{\boldsymbol{y}_{t}{}^{T}\boldsymbol{s}_{t}}$

An Alternative Interpretation

- The update of $\boldsymbol{H}_{t+1}$ based on $\boldsymbol{H}_t$ is also equivalent to the following

$$\min_{\boldsymbol{H}} \langle \boldsymbol{H}_t, \boldsymbol{H}^{-1} \rangle - \log\det(\boldsymbol{H}_t \boldsymbol{H}^{-1})$$

$$\text{s.t.} \quad \boldsymbol{H} = \boldsymbol{H}^T \qquad \ldots\ldots \qquad \text{(symmetric condition)}$$

$$\boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{s}_t \qquad \ldots\ldots \qquad \text{(secant equation)}$$

- In essence, minimizing KL divergence with constraints on the secant equation

Quasi Newton Method

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- Suppose we have an unconstrained optimization problem,

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in R^n$$

  - Start with an initial point $\boldsymbol{x}_0$ and update as follows

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{H}_t \nabla f(\boldsymbol{x}_t)$$

    where η_t is determined by line search

  - Update $\boldsymbol{H}_{t+1}$ by the following

$$\boldsymbol{H}_{t+1} = (\boldsymbol{I} - \rho_t \boldsymbol{s}_t \boldsymbol{y}_t^T)\boldsymbol{H}_t(\boldsymbol{I} - \rho_t \boldsymbol{y}_t \boldsymbol{s}_t^T) + \rho_t \boldsymbol{s}_t \boldsymbol{s}_t^T$$

where $\boldsymbol{y}_t = \nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)$, $\boldsymbol{s}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t$, and $\rho_t = \frac{1}{\boldsymbol{y}_t{}^T \boldsymbol{s}_t}$

Convergence Rate

- Assume f is strongly convex and has Lipschitz-continuous Hessian.
- **Theorem**: BFGS method converges as

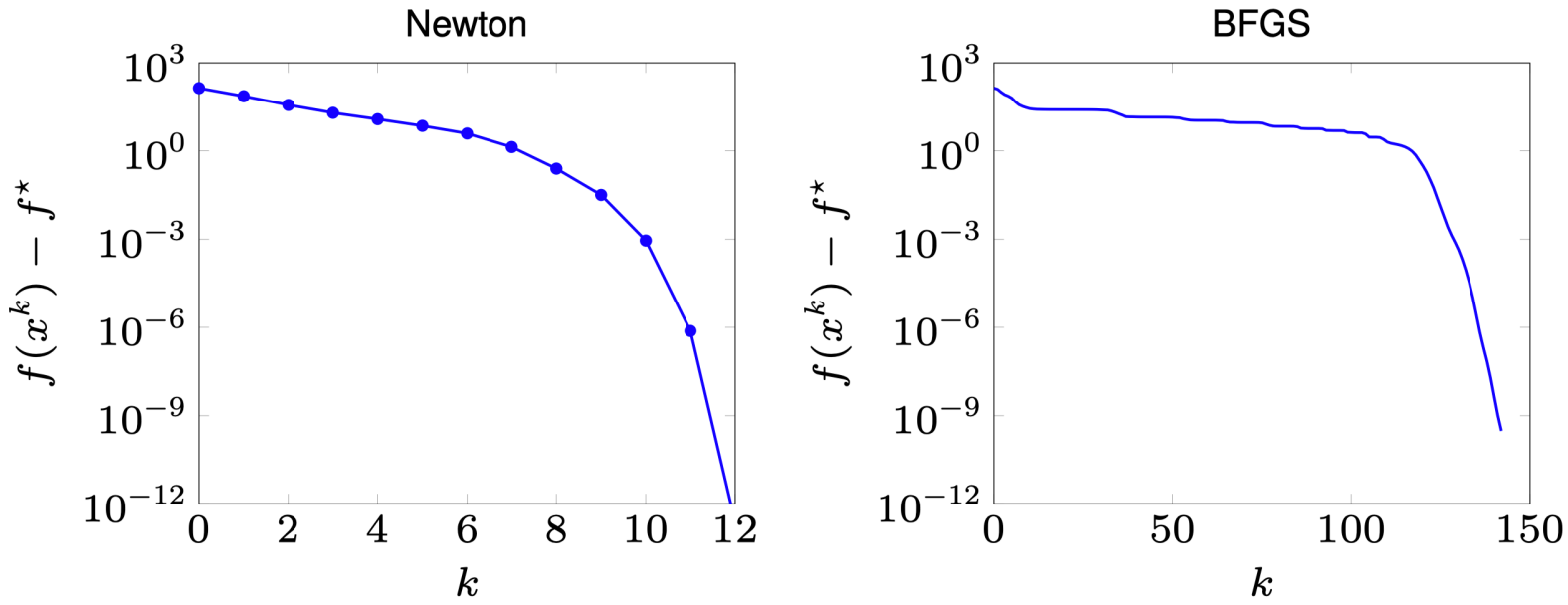
$$\lim_{t\to\infty} \frac{\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|_2}{\|\boldsymbol{x}_t - \boldsymbol{x}^*\|_2} = 0$$

  - Superlinear convergence rate (why): number of required iterations to achieve a certain accuracy is larger than Newton methods but smaller than Gradient descent

Numerical Example

- Exam p le from Vandenberghe's lecture notes: Newton versus BFGS on LP barrier problem, for n=100, m=500

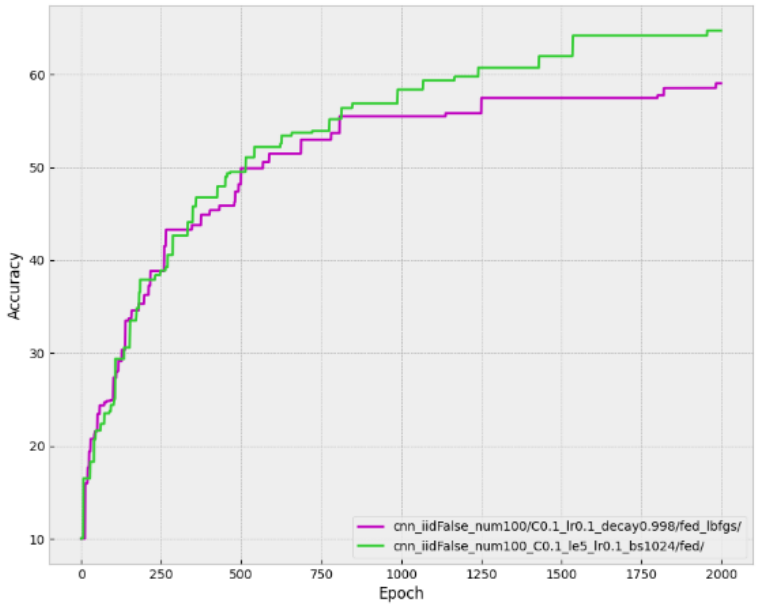
$$\min_x \; c^T x - \sum_{i=1}^{m} \log(b_i - a_i^T x)$$



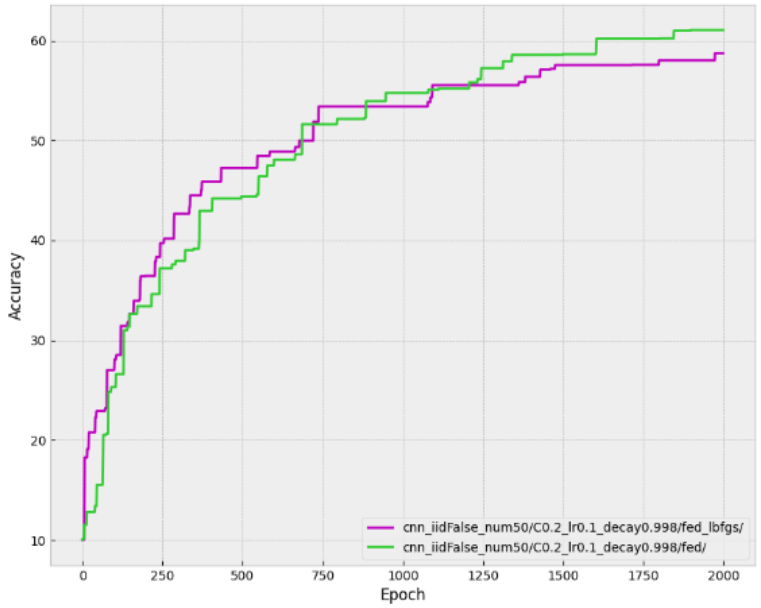
Stochastic Quasi Newton Method

- Can we use the stochastic gradient in BFGS?

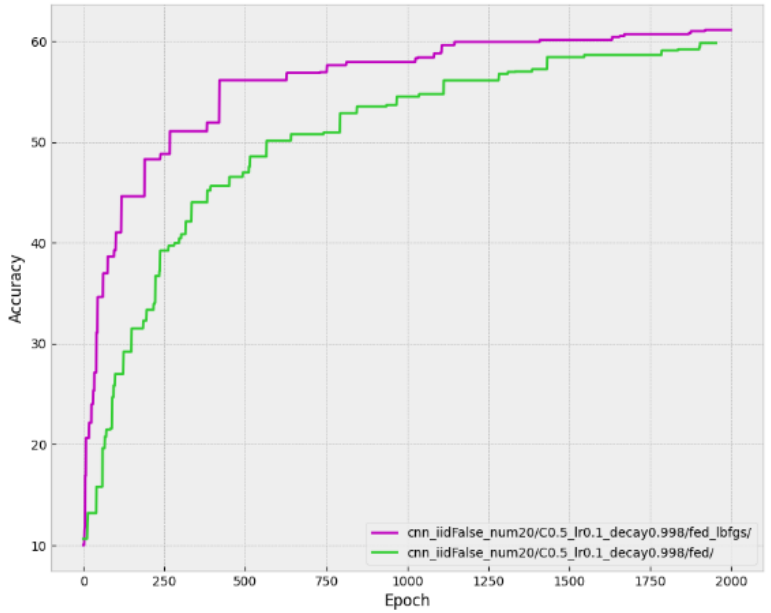
Second-order
First-order



Batch size = 500



Batch size = 1000



Batch size = 2500

Learning Objectives

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- Gradient descent and accelerating methods
- Variants of (stochastic) gradient descent—adaptive optimization methods
- Quasi Newton method