Outline

In this lecture:

- Newton’s method, interpretations, and properties
- Convergence analysis
- Equality-constrained Newton
Newton’s Method

Consider the unconstrained, smooth convex optimization problem:

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

where $f$ is convex, twice differentiable, and $\text{dom}(f) = \mathbb{R}^n$.

**Newton’s method (pure):** Choose initial $\mathbf{x}_0 \in \mathbb{R}^n$, and

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k), \quad k = 0, 1, 2, \ldots$$

where $\mathbf{H}(\mathbf{x}_k)$ is the Hessian matrix of $f$ at $\mathbf{x}_k$.

**Compared to gradient descent:** Choose initial $\mathbf{x}_0 \in \mathbb{R}^n$, and

$$\mathbf{x}_{k+1} = \mathbf{x}_k - s_k \nabla f(\mathbf{x}_k), \quad k = 0, 1, 2, \ldots$$
Interpretations of Newton’s Method

- **Interpretation 1:** Minimizer of the second-order Taylor approximation

  The second-order Taylor approximation of $f$ is:

  $$ f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2} (y - x)^\top \mathbf{H}(x) (y - x) $$

  i.e., quadratic approximation matching the curvature at a given point $x$

- Recall that, by contrast, gradient descent ($x_{k+1} = x_k - s_k \nabla$) can be obtained by minimizing over

  $$ f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2s} \|y - x\|^2 $$
Geometric Interpretation

\[(x, f(x))\]

\[(x + \Delta x, f(x + \Delta x))\]
Newton’s Method: Numerical Example

- For $f(x) = (10x_1^2 + x_2^2)/2 + 5 \log(1 + e^{-x_1-x_2})$
- Gradient descent (black) and Newton’s method (blue) with same step-sizes
History Perspective

- Newton’s method was first developed for root finding (Newton-Raphson).

- Assume \( F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a differentiable vector-valued function and consider the system of equations:

\[
F(x) = 0
\]

- Newton’s method for root finding: Choose initial \( x_0 \in \mathbb{R}^n \) and:

\[
x_k = x_{k-1} - F'(x)^{-1} f(x_{k-1}), \quad k = 1, 2, 3, \ldots,
\]

where \( F'(x) \) denotes the Jacobian matrix of \( F \) at \( x \)

- Here, the Newton step \( x^+ = x - F^{-1}(x)F(x) \) can be obtained by solving \( y \) over the linear approximation of \( F \):

\[
F(y) \approx F(x) + F'(x)(y - x) = 0
\]

- Newton (1685) & Raphson (1690) first used it to find roots of polynomials. Simpson (1740) extended it to general nonlinear equations and minimization
Geometrical Interpretation
Limitations of Newton’s Method

Despite being a highly efficient algorithm (more on its convergence speed later), Newton’s method has two flaws:

- The Jacobian matrix $F'(x)$ could be near singular at some iterations, causing numerical instability.

- More problematically, Newton’s method could be divergent if the starting point is too far away!
A Divergent Example of Newton’s Method

• Let’s use Newton’s method to find the roots of

\[ f(x) = \frac{x}{\sqrt{1 + x^2}} = 0 \]

Clearly, the root is \( x^* = 0 \).

• Note that \( f'(x) = \frac{1}{(1+x^2)^{\frac{3}{2}}} \). So the Newton’s method becomes:

\[
x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k}{\sqrt{1 + x_k^2}}(1 + x_k^2)^{\frac{3}{2}} = -x_k^3
\]

Thus, if \( |x_0| < 1 \), then \( x_k \to 0 \) as \( k \to \infty \). \( x_0 = \pm 1 \) are the oscillation points. If \( |x_0| > 1 \), then the method diverges.

• Therefore, Newton’s method can only claim local convergence!
Graphically

\[ f(x) = \frac{x}{\sqrt{1 + x^2}} \]
Theorem 1

Assume that $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and $x^* \in \mathbb{R}^n$ is a root of $F$, that is, $F(x^*) = 0$ and $F'(x^*)$ is non-singular. Then:

a) There exists $\delta > 0$ such that if $\|x_0 - x^*\| < \delta$, then the Newton’s method is well-defined and

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0 \quad (\text{superlinear}).$$

b) If $F'$ is Lipschitz continuous in a neighborhood of $x^*$, then there exists a constant $K > 0$ such that

$$\|x_{k+1} - x_*\| \leq K \|x_k - x_*\|^2 \quad (\text{2nd-order convergence}).$$
Newton Decrement

- Consider again Newton's method for $\min_{x \in \mathbb{R}^n} f(x)$

- At each $x_k$, define the Newton decrement as:

$$\lambda(x_k) = \left( \nabla f(x_k)^\top H^{-1}(x_k) \nabla f(x_k) \right)^{\frac{1}{2}}$$

- Related to difference btwn $f(x_k)$ and the min of its quadratic approximation:

$$f(x_k) - \min_y (f(x_k)) + \nabla f(x_k)^\top (y - x_k) + \frac{1}{2} (y - x_k)^\top H(x_k) (y - x_k)$$

$$= \frac{1}{2} \nabla f(x_k)^\top H^{-1}(x_k) \nabla f(x_k) = \frac{1}{2} \lambda(x_k)^2$$

- $\frac{1}{2} \lambda(x_k)^2$ can be thought of as an approximation bound on optimality gap $f(x_k) - f^*$, thus usually used as a stopping criterion
We have seen pure Newton’s method might not converge. To address limitation, damped Newton’s method (or “Levenberg-Marquardt” method) is used in practice:

\[ x_{k+1} = x_k - s_k \mathbf{H}^{-1}(x_k) \nabla f(x_k), \]

where \( s_k \in (0, 1] \) is the step-size (\( s_k = 1 \) implies pure Newton’s method)

- \( s_k \) typically chosen by backtracking with parameters \( \alpha \in (0, \frac{1}{2}] \), \( \beta \in (0, 1) \)

In each iteration, we start with \( s_k = 1 \) and while

\[
f(x_k + s_k \mathbf{d}) > f(x_k) + \alpha s_k \nabla f(x_k)^\top \mathbf{d},
\]

we shrink \( s_k = \beta s_k \). Otherwise, we perform Newton update.

Here, \( \mathbf{d} = -\mathbf{H}^{-1}(x_k) f(x_k) \), and hence \( \nabla f(x_k)^\top \mathbf{d} = -\lambda(x_k)^2 \).
Example: Logistic Regression

Logistic regression example, with $n = 500$, $p = 100$: Compare gradient descent and Newton’s method, both with backtracking.
Convergence Result for Damped Newton’s Method

\[ \| H(x) - H(y) \| \leq M \| x - y \| \]

Assume that \( f \in S^{2,1}_{\mu,L} \) and furthermore \( H(x) \) is \( M \)-Lipschitz continuous.

**Theorem 2**

Newton’s method with backtracking line search satisfies the following **two-stage** convergence rate bounds:

\[
 f(x_k) - f^* \leq \begin{cases} 
 (f(x_0) - f^*) - \gamma k, & \text{if } k \leq k_0 \\
 \frac{2\mu^3}{M^2} \left( \frac{1}{2} \right)^{2k-k_0+1}, & \text{if } k > k_0.
\end{cases}
\]

where \( \gamma = \alpha \beta^2 \eta^2 \mu / L^2 \), \( \eta = \min\{1, 3(1 - 2\alpha)\} \mu^2 / M \), and \( k_0 \) is the number of steps until \( \| f(x_{k_0+1}) \|_2 \leq \eta \).

[**BV**, Ch 9.5]
Understanding the Convergence Result

Convergence analysis reveals $\gamma > 0$, $0 < \eta \leq \mu^2 / M$ such that the convergence follows two stages

- **Damped phase:** \[ \| \nabla f(x_k) \|_2 \geq \eta, \text{ and} \]
  \[ f(x_{k+1}) - f(x_k) \leq -\gamma \]

- **Pure phase:** \[ \| \nabla f(x_k) \|_2 < \eta, \text{ backtracking selects } s_k = 1 \text{ and} \]
  \[ \frac{M}{2\mu^2} \| \nabla f(x_{k+1}) \|_2 \leq \left( \frac{M}{2\mu^2} \| \nabla f(x_k) \|_2 \right)^2 \]

Note that once entering pure phase, we won't leave because

\[ \frac{2\mu^2}{M} \left( \frac{M}{2\mu^2} \| f(x_k) \|_2 \right)^2 \leq \eta \]

when $\eta \leq \frac{\mu^2}{M}$

\[ \| f(x_{k+1}) \|_2 \leq \frac{2\mu^2}{M} \left( \frac{M}{2\mu^2} \| f(x_k) \|_2 \right)^2 \leq \frac{2\mu^2}{M} \left( \frac{M}{2\mu^2} \gamma \right)^2 \]

\[ \leq \frac{\mu^2}{M} \gamma^2 \leq \eta. \]
Understanding the Convergence Result

To reach $f(x_k) - f^* \leq \epsilon$, we need at most

$$\frac{f(x_0) - f^*}{\gamma} + \log \log(\epsilon_0/\epsilon)$$

iterations, where $\epsilon \triangleq 2\mu^2/M^2$

- This is called quadratic convergence, in contrast to linear convergence (what gradient descent achieves under strong convexity)

- The above result is a local convergence rate, i.e., we are only guaranteed quadratic convergence after some number of steps $k_0$, where $k_0 \leq \frac{f(x_0) - f^*}{\gamma}$

- The bound depends on $L$, $\mu$, $M$. But the algorithm itself does not
Self-Concordance

A scale-free analysis is possible for self-concordance functions: A convex function \( f : (a, b) \subset \mathbb{R} \to \mathbb{R} \) is called self-concordant if

\[ |f'''(x)| \leq 2f''(x)^{3/2}, \quad \forall x \]

On an open convex domain of \( \mathbb{R}^n \), \( f \) is self-concordant if its restriction to every line in its domain is self-concordant

Examples of self-concordant functions:

- \( f : \mathbb{R}^n_{++} \to \mathbb{R} \) defined by:

\[ f(x) = - \sum_{i=1}^{n} \log(x_i) \]

- \( f : S^n_{++} \to \mathbb{R} \) defined by:

\[ f(X) = - \log(\det(X)) \]
Convergence Analysis for Self-Concordant Functions

Theorem 3 (Nesterov and Nemirovski)

Newton’s method with backtracking line search requires at most

\[ C(\alpha, \beta)(f(x_0) - f^*) + \log \log(1/\epsilon) \]

iterations to reach \( f(x_k) - f^* \leq \epsilon \), where \( C(\alpha, \beta) \) is a constant that only depends on \( \alpha, \beta \)
Comparisons to First-Order Methods

- **Memory**: Each Newton iteration requires $O(n^2)$ storage ($n \times n$ Hessian); Each gradient iteration requires $O(n)$ storage ($n$-dimensional gradient)

- **Computation**: $O(n^3)$ steps per Newton iteration (solving $n \times n$ linear system); $O(n)$ steps per GD iteration (scaling/adding $n$-dim. vectors)

- **Backtracking**: Backtracking line search has roughly the same cost, but uses $O(n)$ steps per inner backtracking step

- **Conditioning**: Newton’s method is not affected by problem’s conditioning, but GD is highly sensitive

- **Fragility**: Newton’s method may be empirically more sensitive to bugs/numerical errors, GD is more robust

- **Sparse, structured problems**: Newton’s method is more advantageous because the inner linear systems (in Hessian) can be solved efficiently and reliably (e.g., diagonal, banded, etc.)
$x$-axis is in terms of time (each GD step is $O(n)$, but each Newton step is $O(n^3)$)
Equality-Constrained Newton’s Method

Consider new a problem with equality constraints

\[
\min_x f(x), \quad \text{subject to} \quad Ax = b \quad \forall \in \mathbb{R}^n
\]

Start with \( x_0 \in \mathbb{R}^n \), repeat the following update:

\[
x_{k+1} = x_k + s_k d,
\]

where

\[
d = \arg \min_{A(x_k+z)=b} \left( f(x_k) + \nabla f(x_k)^\top z + \frac{1}{2}z^\top H(x_k)z \right)
\]

From KKT conditions, it follows that for some dual variables \( w, z \) satisfy

\[
A^T(Az) = b \quad \forall \in \mathbb{R}^n
\]

\[
(\text{OT})(\text{CS}): \text{implied by (PF)}
\]

\[
[H(x_k) \quad A^\top] [z \quad w]^\top = -[\nabla f(x_k) \quad 0]^\top
\]

Let: Use Newton's method to solve a primal-dual method.
Next Class

Quasi-Newton and Interior-Point Methods