Newton-Type Methods

Exploring the Interplay Between Inner and Outer Iterations

Part I

Fred Roosta

School of Mathematics and Physics
University of Queensland

$$\min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d} \quad f(\mathbf{x})$$

- f is twice (continuously) differentiable and lower bounded.
- High-dimensional: $d \gg 1$.

$$\min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

- f is twice (continuously) differentiable and lower bounded.
- High-dimensional: $d \gg 1$.
- "Big data": $n \gg 1$

In machine learning:

$$f(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\, \ell(\, h(\mathbf{x}, \mathbf{a})\,, b) \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

In machine learning:

$$f(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\, \ell(\overbrace{h(\mathbf{x}, \mathbf{a})}^{\text{NN}}, b) \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

In machine learning:

$$\color{blue}{f}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\, \overbrace{\color{red}{\ell}\big(\overbrace{h(\mathbf{x}, \mathbf{a})}^{\text{NN}}, b\big)}^{\textit{Loss}} \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

In machine learning:

$$f(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathcal{D}}\,\overbrace{\ell(\,\overbrace{h(\mathbf{x},\mathbf{a})}^{\text{NN}}\,, b)}^{\textit{Loss}}}_{\textit{Risk}} \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

In machine learning:

$$f(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathcal{D}}\, \overbrace{\ell(\, \overbrace{h(\mathbf{x}, \mathbf{a})}^{\text{NN}}\, , b)}^{Loss}}_{Risk} \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

Empirical average using samples $\{(\mathbf{a}_i, b)\}_{i=1}^{n}$ gives

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{x}, \mathbf{a}_i), b_i)$$

In machine learning:

$$f(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathcal{D}}\, \overbrace{\ell(\overbrace{h(\mathbf{x}, \mathbf{a})}^{\text{NN}}, b)}^{\textit{Loss}}}_{\textit{Risk}} \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

Empirical average using samples $\{(\mathbf{a}_i, b)\}_{i=1}^{n}$ gives

$$f(\mathbf{x}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}, \mathbf{a}_i), b_i)}_{\text{empirical risk}}$$

In machine learning:

$$f(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathcal{D}}\, \overbrace{\ell(\, \overbrace{h(\mathbf{x}, \mathbf{a})}^{\text{NN}}\, , b)}^{\textit{Loss}}}_{\textit{Risk}} \quad \text{where} \quad (\mathbf{a}, b) \sim \mathcal{D}$$

Empirical average using samples $\{(\mathbf{a}_i, b)\}_{i=1}^{n}$ gives

$$f(\mathbf{x}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}, \mathbf{a}_i), b_i)}_{\text{empirical risk}} = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

Notation

- scalars: lower case, e.g., α
- Vectors: bold lower case, e.g., $\mathbf{x}$
- Matrices: bold upper case, e.g., $\mathbf{H}$
- $\mathbf{g}(\mathbf{x}) \triangleq \nabla f(\mathbf{x})$
- $\mathbf{H}(\mathbf{x}) \triangleq \nabla^2 f(\mathbf{x})$
- Outer iteration counter: subscript, e.g., $\mathbf{x}_k$, f_k, $\mathbf{g}_k$, $\mathbf{H}_k$
- Inner iteration counter: superscript, e.g., $\mathbf{p}_k^{(t)}$, $\mathbf{p}^{(t)}$
- Inner product of $\mathbf{v}$ and $\mathbf{w}$ is denoted by $\langle \mathbf{v}, \mathbf{w} \rangle$

Loosely speaking, there are two classes of algorithms:

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$
- 2^{nd}-order algorithms

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \, \mathbf{H}_k^{-1} \mathbf{g}_k$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \overbrace{\mathbf{H}_k^{-1}\mathbf{g}_k}^{\text{Linear System}}$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \overbrace{\mathbf{H}_k^{-1}\mathbf{g}_k}^{\text{Linear System}}$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}(\mathbf{x}_k - \alpha_k \mathbf{g}_k)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \overbrace{\mathbf{H}_k^{-1}\mathbf{g}_k}^{\text{Linear System}}$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1}\mathbf{g}_k\right)$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \overbrace{\mathbf{H}_k^{-1}\mathbf{g}_k}^{\text{Linear System}}$
  - Constrained: $\mathbf{x}_{k+1} = \cancel{\mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1}\mathbf{g}_k\right)}$

Loosely speaking, there are two classes of algorithms:

- 1^{st}-order algorithms, e.g., (projected) gradient descent
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
  - Constrained: $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_k - \alpha_k \mathbf{g}_k\right)$

- 2^{nd}-order algorithms, e.g., (projected) Newton's method
  - Unconstrained ($\mathcal{X} = \mathbb{R}^d$): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \overbrace{\mathbf{H}_k^{-1}\mathbf{g}_k}^{\text{Linear System}}$
  - Constrained: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k(\mathbf{y}_k - \mathbf{x}_k)$, where

$$\mathbf{y}_k = \underset{\mathbf{y} \in \mathcal{X}}{\arg\min} \; \langle \mathbf{g}_k, \mathbf{y} - \mathbf{x}_k \rangle + \frac{1}{2} \langle \mathbf{y} - \mathbf{x}_k, \mathbf{H}_k(\mathbf{y} - \mathbf{x}_k) \rangle$$

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**

- **Inner iterations**

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
- **Inner iterations**

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
  - Formulate the subproblem
- **Inner iterations**

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
  - Formulate the subproblem
  - Update the iterate
- **Inner iterations**

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
  - Formulate the subproblem
  - Update the iterate
  - Check for convergence
- **Inner iterations**

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
  - Formulate the subproblem
  - Update the iterate
  - Check for convergence
- **Inner iterations**
  - Iteratively solve the subproblem (approximately)

Ingredients for almost all Newton-type-algorithms:

- **Outer Iterations**
  - Evaluate the function and its derivatives
  - Formulate the subproblem
  - Update the iterate
  - Check for convergence
- **Inner iterations**
  - Iteratively solve the subproblem (approximately)

Treating the subproblem solver as a black box often necessitates

- unnecessary assumptions,
- unnecessary safeguards,
- complex analysis, and
- complicated algorithms.

Treating the subproblem solver as a black box often necessitates

- unnecessary assumptions,
- unnecessary safeguards,
- complex analysis, and
- complicated algorithms.

Leveraging the properties of a suitable solver can

- reduce unnecessary assumptions,
- remove unnecessary safeguards,
- simplify analysis, and
- simplify algorithms.

Outline:

Outline:

1. Consequences of treating subproblem solvers as "black box"

Outline:

1. Consequences of treating subproblem solvers as “black box”

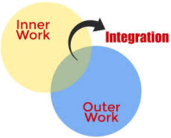
2. Open the the box and derive the properties of the solvers

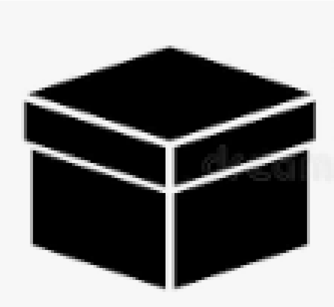
Outline:

1. Consequences of treating subproblem solvers as “black box”

2. Open the the box and derive the properties of the solvers

3. Integrate the inner and outer iterations





$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- f has Lipschitz continuous gradient

Algorithm Generic 2$^{\text{nd}}$-order Method

Start from $\mathbf{x}_0$

Andrew R Conn, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM; Jorge Nocedal and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media; Coralia Cartis, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM

Algorithm Generic 2nd-order Method

Start from $\mathbf{x}_0$

for $k = 1, 2, \ldots$ **do**

$$\mathbf{p}_k = \begin{cases} \alpha_k \mathbf{p} \quad \text{where} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k & \text{(Line Search)} \\ \\ \\ \\ \end{cases}$$

end for

Andrew R Conn, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM; Jorge Nocedal and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media; Coralia Cartis, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM

Algorithm Generic 2[nd]-order Method

Start from $\mathbf{x}_0$

for $k = 1, 2, \ldots$ **do**

$$\mathbf{p}_k = \begin{cases} \alpha_k \mathbf{p} \quad \text{where} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k & \text{(Line Search)} \\ \\ \underset{\|\mathbf{p}\| \le \Delta_k}{\arg\min} \; \langle \mathbf{p}, \mathbf{g}_k \rangle + \dfrac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} & \text{(Trust Region)} \end{cases}$$

end for

Andrew R Conn, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM; Jorge Nocedal and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media; Coralia Cartis, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM

Algorithm Generic 2nd-order Method

Start from $\mathbf{x}_0$

for $k = 1, 2, \ldots$ **do**

$$\mathbf{p}_k = \begin{cases} \alpha_k \mathbf{p} \quad \text{where} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k & \text{(Line Search)} \\ \underset{\|\mathbf{p}\| \leq \Delta_k}{\arg\min} \ \langle \mathbf{p}, \mathbf{g}_k \rangle + \dfrac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} & \text{(Trust Region)} \\ \underset{\|\mathbf{p}\| \in \mathbb{R}^d}{\arg\min} \ \langle \mathbf{p}, \mathbf{g}_k \rangle + \dfrac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} + \dfrac{\sigma_k}{3} \|\mathbf{p}_k\|^3 & \text{(Cubic Regularization)} \end{cases}$$

end for

Andrew R Conn, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM; Jorge Nocedal and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media; Coralia Cartis, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM

Algorithm Generic 2^{nd}-order Method

Start from $\mathbf{x}_0$

for $k = 1, 2, \ldots$ **do**

$$\mathbf{p}_k = \begin{cases} \alpha_k \mathbf{p} \quad \text{where} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k & \text{(Line Search)} \\ \\ \underset{\|\mathbf{p}\| \leq \Delta_k}{\arg\min} \; \langle \mathbf{p}, \mathbf{g}_k \rangle + \dfrac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} & \text{(Trust Region)} \\ \\ \underset{\|\mathbf{p}\| \in \mathbb{R}^d}{\arg\min} \; \langle \mathbf{p}, \mathbf{g}_k \rangle + \dfrac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} + \dfrac{\sigma_k}{3} \|\mathbf{p}_k\|^3 & \text{(Cubic Regularization)} \end{cases}$$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$

end for

Andrew R Conn, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM; Jorge Nocedal and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media; Coralia Cartis, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM

We only explore the line search framework, but the essence of what is to come can applied to other frameworks as well.

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Sufficient Condition for Descent

If $\langle \mathbf{p}_k, \mathbf{g}_k \rangle < 0$, then $\mathbf{p}_k$ is a descent direction for f at $\mathbf{x}_k$.

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Sufficient Condition for Descent

If $\langle \mathbf{p}_k, \mathbf{g}_k \rangle < 0$, then $\mathbf{p}_k$ is a descent direction for f at $\mathbf{x}_k$.

In line search framework, the exact solution is $\mathbf{p}_k = -\mathbf{H}_k^{-1} \mathbf{g}_k$.

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Sufficient Condition for Descent

If $\langle \mathbf{p}_k, \mathbf{g}_k \rangle < 0$, then $\mathbf{p}_k$ is a descent direction for f at $\mathbf{x}_k$.

In line search framework, the exact solution is $\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$. So to have a descent direction, we need $\left\langle \mathbf{g}_k, \mathbf{H}_k^{-1}\mathbf{g}_k \right\rangle > 0, \quad \forall k \geq 0$.

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Sufficient Condition for Descent

If $\langle \mathbf{p}_k, \mathbf{g}_k \rangle < 0$, then $\mathbf{p}_k$ is a descent direction for f at $\mathbf{x}_k$.

In line search framework, the exact solution is $\mathbf{p}_k = -\mathbf{H}_k^{-1} \mathbf{g}_k$. So to have a descent direction, we need $\left\langle \mathbf{g}_k, \mathbf{H}_k^{-1} \mathbf{g}_k \right\rangle > 0, \quad \forall k \geq 0$. Without any other information, this can be guaranteed if $\mathbf{H}_k \succ \mathbf{0}$ for all k,

Descent Direction

A direction $\mathbf{p}_k \in \mathbb{R}^d$ is a descent direction for f at $\mathbf{x}_k$ if $\exists \bar{\alpha} > 0$, such that

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

In words, there is a line segment from $\mathbf{x}$ along which the function has smaller values than $f(\mathbf{x})$.

Sufficient Condition for Descent

If $\langle \mathbf{p}_k, \mathbf{g}_k \rangle < 0$, then $\mathbf{p}_k$ is a descent direction for f at $\mathbf{x}_k$.

In line search framework, the exact solution is $\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$. So to have a descent direction, we need $\left\langle \mathbf{g}_k, \mathbf{H}_k^{-1}\mathbf{g}_k \right\rangle > 0, \quad \forall k \geq 0$. Without any other information, this can be guaranteed if $\mathbf{H}_k \succ \mathbf{0}$ for all k, i.e., if we assume f is **strongly convex**.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- f has Lipschitz continuous gradient
- f is strongly convex

In "big data" regime, i.e., $n \gg 1$, Hessian evaluations can be very expensive...

In "big data" regime, i.e., $n \gg 1$, Hessian evaluations can be very expensive...We can sub-sample Hessian:

Hessian Sub-Sampling

$$\widehat{\mathbf{H}} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x}), \quad \text{where} \quad \mathcal{S} \subset \{1, 2, \ldots, n\}.$$

In "big data" regime, i.e., $n \gg 1$, Hessian evaluations can be very expensive...We can sub-sample Hessian:

Hessian Sub-Sampling

$$\widehat{\mathbf{H}} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x}), \quad \text{where} \quad \mathcal{S} \subset \{1, 2, \ldots, n\}.$$

Now, the exact Newton's direction becomes $\mathbf{p}_k = -[\widehat{\mathbf{H}}_k]^{-1} \mathbf{g}_k$.

In "big data" regime, i.e., $n \gg 1$, Hessian evaluations can be very expensive...We can sub-sample Hessian:

Hessian Sub-Sampling

$$\widehat{\mathbf{H}} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x}), \quad \text{where} \quad \mathcal{S} \subset \{1, 2, \ldots, n\}.$$

Now, the exact Newton's direction becomes $\mathbf{p}_k = -[\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k$. So to have a descent direction, we need

$$\left\langle \mathbf{g}_k, [\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k \right\rangle > 0, \quad \forall\, k \geq 0 \quad \text{and} \quad \forall\, |\mathcal{S}| \geq 1.$$

In "big data" regime, i.e., $n \gg 1$, Hessian evaluations can be very expensive...We can sub-sample Hessian:

Hessian Sub-Sampling

$$\widehat{\mathbf{H}} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x}), \quad \text{where} \quad \mathcal{S} \subset \{1, 2, \ldots, n\}.$$

Now, the exact Newton's direction becomes $\mathbf{p}_k = -[\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k$. So to have a descent direction, we need

$$\left\langle \mathbf{g}_k, [\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k \right\rangle > 0, \quad \forall\, k \geq 0 \quad \text{and} \quad \forall\, |\mathcal{S}| \geq 1.$$

Without any other information, this can be guaranteed if each f_i is **strongly convex**.

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- Each f_i is strongly convex

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\underline{\text{Each } f_i}$ has Lipschitz continuous gradient
- $\underline{\text{Each } f_i}$ is strongly convex

Byrd et al. (2011) gives

$$\lim_{k\to\infty} \mathbf{g}_k = \mathbf{0},$$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- Each f_i is strongly convex

Byrd et al. (2011) gives

$$\lim_{k\to\infty} \mathbf{g}_k = \mathbf{0},$$

while with some extra variance assumption, Bollapragada, Byrd, and Nocedal (2018) gives

$$\mathbb{E}\left(f_k - f^\star\right) \le \rho^k \left(f_0 - f^\star\right) \quad \text{for some} \quad 0 \le \rho < 1.$$

What if f is strongly convex, but each f_i is only convex?

What if f is strongly convex, but each f_i is only convex?

Example

Suppose $f_i(\mathbf{x}) = \ell_i(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)$, where $\ell_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and $\ell_i'' \geq \gamma > 0$.

What if f is strongly convex, but each f_i is only convex?

Example

Suppose $f_i(\mathbf{x}) = \ell_i(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)$, where $\ell_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and $\ell_i'' \geq \gamma > 0$. Also, suppose

$$\text{Range}(\{\mathbf{a}_i\}_{i=1}^{n}) = \mathbb{R}^d,$$

and in particular $n \geq d$.

What if f is strongly convex, but each f_i is only convex?

Example

Suppose $f_i(\mathbf{x}) = \ell_i(\langle \mathbf{a}_i, \mathbf{x}\rangle, b_i)$, where $\ell_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and $\ell_i'' \geq \gamma > 0$. Also, suppose

$$\mathrm{Range}(\{\mathbf{a}_i\}_{i=1}^n) = \mathbb{R}^d,$$

and in particular $n \geq d$.

Each $\nabla^2 f_i(\mathbf{x}) = \ell_i''(\langle \mathbf{a}_i, \mathbf{x}\rangle, b_i)\mathbf{a}_i\mathbf{a}_i^T$ is *rank one*!

What if f is strongly convex, but each f_i is only convex?

Example

Suppose $f_i(\mathbf{x}) = \ell_i(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)$, where $\ell_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and $\ell_i'' \geq \gamma > 0$. Also, suppose

$$\text{Range}(\{\mathbf{a}_i\}_{i=1}^n) = \mathbb{R}^d,$$

and in particular $n \geq d$.

Each $\nabla^2 f_i(\mathbf{x}) = \ell_i''(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)\mathbf{a}_i\mathbf{a}_i^\intercal$ is *rank one*!

But $\nabla^2 f(\mathbf{x}) \succeq \gamma \cdot \lambda_{\min}\left(\sum_{i=1}^n \mathbf{a}_i\mathbf{a}_i^\intercal\right)$

What if f is strongly convex, but each f_i is only convex?

Example

Suppose $f_i(\mathbf{x}) = \ell_i(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)$, where $\ell_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and $\ell_i'' \geq \gamma > 0$.
Also, suppose

$$\text{Range}(\{\mathbf{a}_i\}_{i=1}^n) = \mathbb{R}^d,$$

and in particular $n \geq d$.

Each $\nabla^2 f_i(\mathbf{x}) = \ell_i''(\langle \mathbf{a}_i, \mathbf{x} \rangle, b_i)\mathbf{a}_i\mathbf{a}_i^\intercal$ is *rank one*!

But $\nabla^2 f(\mathbf{x}) \succeq \gamma \cdot \underbrace{\lambda_{\min}\left(\sum_{i=1}^n \mathbf{a}_i\mathbf{a}_i^\intercal\right)}_{>0}$

Lemma (Roosta and Mahoney, 2019)

Suppose $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ *and* $0 \preceq \nabla^2 f_i(\mathbf{x}) \preceq L_{\mathbf{g}} \mathbf{I}$.

Lemma (Roosta and Mahoney, 2019)

Suppose $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ *and* $0 \preceq \nabla^2 f_i(\mathbf{x}) \preceq L_{\mathbf{g}} \mathbf{I}$. *Given any* $0 < \epsilon < 1$, $0 < \delta < 1$

Lemma (Roosta and Mahoney, 2019)

Suppose $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ *and* $0 \preceq \nabla^2 f_i(\mathbf{x}) \preceq L_{\mathbf{g}} \mathbf{I}$. *Given any* $0 < \epsilon < 1$, $0 < \delta < 1$, *if Hessian is uniformly sub-sampled with*

$$|\mathcal{S}| \geq \frac{2\kappa \log(d/\delta)}{\epsilon^2},$$

Lemma (Roosta and Mahoney, 2019)

Suppose $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ *and* $0 \preceq \nabla^2 f_i(\mathbf{x}) \preceq L_{\mathbf{g}} \mathbf{I}$. *Given any* $0 < \epsilon < 1$, $0 < \delta < 1$, *if Hessian is uniformly sub-sampled with*

$$|\mathcal{S}| \geq \frac{2\kappa \log(d/\delta)}{\epsilon^2},$$

then

$$\mathbb{P}\left(\widehat{\mathbf{H}} \succeq (1-\epsilon)\mu \mathbf{I}\right) \geq 1 - \delta.$$

where $\kappa = L_{\mathbf{g}}/\mu$.

Lemma (Roosta and Mahoney, 2019)

Suppose $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ *and* $0 \preceq \nabla^2 f_i(\mathbf{x}) \preceq L_{\mathbf{g}} \mathbf{I}$*. Given any* $0 < \epsilon < 1$, $0 < \delta < 1$*, if Hessian is uniformly sub-sampled with*

$$|\mathcal{S}| \geq \frac{2\kappa \log(d/\delta)}{\epsilon^2},$$

then

$$\mathbb{P}\left(\widehat{\mathbf{H}} \succeq (1-\epsilon)\mu \mathbf{I}\right) \geq 1 - \delta.$$

where $\kappa = L_{\mathbf{g}}/\mu$.

Proof.

Follows from Matrix Chernoff (Tropp, 2011; Tropp, 2012) bound for sampling with or without replacement. □

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- <u>Each f_i</u> has Lipschitz continuous gradient
- f is strongly convex but <u>each f_i</u> is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

Proof.

$\square$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\left\langle \mathbf{p}_k, \mathbf{g}_k\right\rangle + \alpha^2 L_{\mathbf{g}} \left\|\mathbf{p}_k\right\|^2/2$.

$\square$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\langle\mathbf{p}_k, \mathbf{g}_k\rangle + \alpha^2 L_{\mathbf{g}}\|\mathbf{p}_k\|^2/2$. If $\alpha \leq 2(1-\beta)(1-\epsilon)/\kappa$, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\beta\langle\mathbf{p}_k, \mathbf{g}_k\rangle$. $\square$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\left\langle \mathbf{p}_k, \mathbf{g}_k\right\rangle + \alpha^2 L_{\mathbf{g}} \left\|\mathbf{p}_k\right\|^2/2$. If $\alpha \leq 2(1-\beta)(1-\epsilon)/\kappa$, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\beta\left\langle \mathbf{p}_k, \mathbf{g}_k\right\rangle$. Since, $(1-\epsilon)\mu \prec \widehat{\mathbf{H}}_k \prec L_{\mathbf{g}}\mathbf{I}$,

$\square$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\,(f_k - f^\star)$ *for some* $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq f_k + \alpha \langle \mathbf{p}_k, \mathbf{g}_k \rangle + \alpha^2 L_{\mathbf{g}} \|\mathbf{p}_k\|^2 / 2$. If $\alpha \leq 2(1-\beta)(1-\epsilon)/\kappa$, we get $f(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq f_k + \alpha\beta \langle \mathbf{p}_k, \mathbf{g}_k \rangle$. Since, $(1-\epsilon)\mu \prec \widehat{\mathbf{H}}_k \prec L_{\mathbf{g}}\mathbf{I}$, we have $\langle \mathbf{p}_k, \mathbf{g}_k \rangle = -\left\langle \mathbf{p}_k, \widehat{\mathbf{H}}_k \mathbf{p}_k \right\rangle \leq -(1-\epsilon)\mu \|\mathbf{p}_k\|^2 < 0$,

□

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ for some $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\left\langle \mathbf{p}_k, \mathbf{g}_k \right\rangle + \alpha^2 L_{\mathbf{g}}\left\|\mathbf{p}_k\right\|^2/2$. If $\alpha \leq 2(1-\beta)(1-\epsilon)/\kappa$, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\beta\left\langle \mathbf{p}_k, \mathbf{g}_k \right\rangle$. Since, $(1-\epsilon)\mu \prec \widehat{\mathbf{H}}_k \prec L_{\mathbf{g}}\mathbf{I}$, we have $\left\langle \mathbf{p}_k, \mathbf{g}_k \right\rangle = -\left\langle \mathbf{p}_k, \widehat{\mathbf{H}}_k\mathbf{p}_k \right\rangle \leq -(1-\epsilon)\mu\left\|\mathbf{p}_k\right\|^2 < 0,$ and $\left\|\mathbf{p}_k\right\| = \left\|[\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k\right\| \geq \left\|\mathbf{g}_k\right\|/L_{\mathbf{g}}$. $\square$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho\left(f_k - f^\star\right)$ *for some* $0 \leq \rho < 1$.

Proof.

From Lipschitz continuity, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\langle\mathbf{p}_k, \mathbf{g}_k\rangle + \alpha^2 L_{\mathbf{g}}\|\mathbf{p}_k\|^2/2$. If $\alpha \leq 2(1-\beta)(1-\epsilon)/\kappa$, we get $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f_k + \alpha\beta\langle\mathbf{p}_k, \mathbf{g}_k\rangle$. Since, $(1-\epsilon)\mu \prec \widehat{\mathbf{H}}_k \prec L_{\mathbf{g}}\mathbf{I}$, we have $\langle\mathbf{p}_k, \mathbf{g}_k\rangle = -\left\langle\mathbf{p}_k, \widehat{\mathbf{H}}_k\mathbf{p}_k\right\rangle \leq -(1-\epsilon)\mu\|\mathbf{p}_k\|^2 < 0$, and $\|\mathbf{p}_k\| = \|[\widehat{\mathbf{H}}_k]^{-1}\mathbf{g}_k\| \geq \|\mathbf{g}_k\|/L_{\mathbf{g}}$. Now, μ-strong convexity of f, gives the result. □

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use?

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods.

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,
- Optimal rate for positive definite settings, and

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,
- Optimal rate for positive definite settings, and
- Every iteration of CG is a descent direction

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,
- Optimal rate for positive definite settings, and
- Every iteration of CG is a descent direction

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \ \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2}$$

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,
- Optimal rate for positive definite settings, and
- Every iteration of CG is a descent direction

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k + \mathbf{H}_k \mathbf{p}_k^{(t)} \right\rangle = 0$$

In high-dimensional problems, i.e., $d \gg 1$, inverting the Hessian can be impractical...we can perform inexact update:

Inexact Update

$$\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\| \quad \text{for some} \quad \theta < 1.$$

What solver to use? **Conjugate gradient (CG)** (Björck, 2015) is the most widely used, giving rise to **Newton-CG** methods. But why CG?

- Simple
- Extensively covered in textbooks,
- Many available software libraries,
- Optimal rate for positive definite settings, and
- Every iteration of CG is a descent direction

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k + \mathbf{H}_k \mathbf{p}_k^{(t)} \right\rangle = 0$$

$$\implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \right\rangle = -\left\langle \mathbf{p}_k^{(t)}, \mathbf{H}_k \mathbf{p}_k^{(t)} \right\rangle < 0$$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \left\| \mathbf{g}_k \right\|$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \left\| \mathbf{g}_k \right\|$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho \left(f_k - f^\star \right)$ *for some* $0 \leq \rho < 1$.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \left\| \mathbf{g}_k \right\|$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho \left(f_k - f^\star \right)$ *for some* $0 \leq \rho < 1$.

Proof.

We have $\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \rangle = -\langle \mathbf{p}_k^{(t)}, \widehat{\mathbf{H}}_k \mathbf{p}_k^{(t)} \rangle \leq -(1-\epsilon)\mu \|\mathbf{p}_k^{(t)}\|^2$.

□

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \left\| \mathbf{g}_k \right\|$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho \left(f_k - f^\star \right)$ *for some* $0 \leq \rho < 1$.

Proof.

We have $\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \rangle = -\langle \mathbf{p}_k^{(t)}, \widehat{\mathbf{H}}_k \mathbf{p}_k^{(t)} \rangle \leq -(1-\epsilon)\mu \|\mathbf{p}_k^{(t)}\|^2$. This coupled with $\|\widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\|$ gives $\langle \mathbf{p}, \mathbf{g}_k \rangle \leq -(1-\theta)^2 \mu \|\mathbf{g}_k\|^2 / L_{\mathbf{g}}{}^2$. □

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- Each f_i has Lipschitz continuous gradient
- f is strongly convex but each f_i is ~~strongly~~ convex
- $|S| \in \Omega(\kappa)$
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \left\| \mathbf{g}_k \right\|$

Theorem (Roosta and Mahoney, 2019)

With high probability, $f_{k+1} - f^\star \leq \rho \left(f_k - f^\star \right)$ *for some* $0 \leq \rho < 1$*.*

Proof.

We have $\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \rangle = -\langle \mathbf{p}_k^{(t)}, \widehat{\mathbf{H}}_k \mathbf{p}_k^{(t)} \rangle \leq -(1-\epsilon)\mu \|\mathbf{p}_k^{(t)}\|^2$. This coupled with $\|\widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k\| \leq \theta \|\mathbf{g}_k\|$ gives $\langle \mathbf{p}, \mathbf{g}_k \rangle \leq -(1-\theta)^2 \mu \|\mathbf{g}_k\|^2 / {L_\mathbf{g}}^2$. The proof then follows a very similar line of reasoning as the exact case. □

What if $f(\mathbf{x})$ is convex but not strongly so!

What if $f(\mathbf{x})$ is convex but not strongly so!

In this case, $\mathbf{H}_k$ can become singular

What if $f(\mathbf{x})$ is convex but not strongly so!

In this case, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent

What if $f(\mathbf{x})$ is convex but not strongly so!

In this case, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent, i.e.,

$$\nexists \mathbf{p} \quad \text{such that} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k.$$

What if $f(\mathbf{x})$ is convex but not strongly so!

In this case, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent, i.e.,

$$\nexists \mathbf{p} \quad \text{such that} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k.$$

So, there might be no inverse $\mathbf{H}_k^{-1}$

What if $f(\mathbf{x})$ is convex but not strongly so!

In this case, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent, i.e.,

$$\nexists \mathbf{p} \quad \text{such that} \quad \mathbf{H}_k \mathbf{p} = -\mathbf{g}_k.$$

So, there might be no inverse $\mathbf{H}_k^{-1}$.... but there is always pseudo-inverse $\mathbf{H}_k^{\dagger}$

$$\min_{\mathbf{x} \in \mathbb{R}^d} \; f(\mathbf{x})$$

Assumptions:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x})$$

Assumptions:

- f is relatively smoothness

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L_{\mathbf{g}}}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x})$$

Assumptions:

- f is relatively smoothness

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L_{\mathbf{g}}}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

- f is ~~strongly~~ relatively convex

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$$

Assumptions:

- f is relatively smoothness

$$f(\mathbf{y}) \le f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L_{\mathbf{g}}}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

- f is ~~strongly~~ relatively convex

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

Theorem (Karimireddy, Stich, and Jaggi, 2018)

With $\mathbf{p}_k = -\alpha \mathbf{H}_k^{\dagger}\mathbf{g}_k$ and $\alpha < 1/L_{\mathbf{g}}$, we have $f_{k+1} - f^\star \le \rho\left(f_k - f^\star\right)$ for some $0 \le \rho < 1$.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x})$$

Assumptions:

- f is relatively smoothness

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L_{\mathbf{g}}}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

- f is ~~strongly~~ relatively convex

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{H}(\mathbf{x})}$$

Theorem (Karimireddy, Stich, and Jaggi, 2018)

With $\mathbf{p}_k = -\alpha \mathbf{H}_k^{\dagger}\mathbf{g}_k$ and $\alpha < 1/L_{\mathbf{g}}$, we have $f_{k+1} - f^\star \leq \rho(f_k - f^\star)$ for some $0 \leq \rho < 1$.

Note: No results for finite sum problems or inexact CG variant (AFAIK)

What if $f(\mathbf{x})$ is non-convex!

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown, i.e., if $\exists \mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)$, $\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle \leq 0$,

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown, i.e., if $\exists \mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)$, $\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle \leq 0$, then

$$\inf_{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} = -\infty$$

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown, i.e., if $\exists \mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)$, $\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle \leq 0$, then

$$\inf_{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} = -\infty$$

- Even if CG does not breakdown, many of its directions may be ascent directions

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown, i.e., if $\exists \mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)$, $\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle \leq 0$, then

$$\inf_{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} = -\infty$$

- Even if CG does not breakdown, many of its directions may be ascent directions, i.e., $\left\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \right\rangle > 0$,

What if $f(\mathbf{x})$ is non-convex!

- Again, $\mathbf{H}_k$ can become singular, and if $\mathbf{g}_k \notin \text{Range}(\mathbf{H}_k)$, the system is inconsistent
- In addition, CG can breakdown, i.e., if $\exists \mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)$, $\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle \leq 0$, then

$$\inf_{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)} \langle \mathbf{p}, \mathbf{g}_k \rangle + \frac{\langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle}{2} = -\infty$$

- Even if CG does not breakdown, many of its directions may be ascent directions, i.e., $\left\langle \mathbf{p}_k^{(t)}, \mathbf{g}_k \right\rangle > 0$, for example, $\mathbf{p}_k = -[\mathbf{H}_k]^{-1} \mathbf{g}_k$ where $\mathbf{H}_k$ invertible but indefinite

Safeguards/Strategies

Safeguards/Strategies

- Goldstein-Price Method (Goldstein and Price, 1967):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{p} = -\mathbf{g}$$

Safeguards/Strategies

- Goldstein-Price Method (Goldstein and Price, 1967):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{p} = -\mathbf{g}$$

- Modify the spectrum of the Hessian:

Safeguards/Strategies

- Goldstein-Price Method (Goldstein and Price, 1967):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{p} = -\mathbf{g}$$

- Modify the spectrum of the Hessian:
  - Goldfeld et al. Method (Goldfeld, Quandt, and Trotter, 1966):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{H} \leftarrow \mathbf{H} + \lambda \mathbf{I}$$

Safeguards/Strategies

- Goldstein-Price Method (Goldstein and Price, 1967):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{p} = -\mathbf{g}$$

- Modify the spectrum of the Hessian:
  - Goldfeld et al. Method (Goldfeld, Quandt, and Trotter, 1966):

$$\mathbf{H} \not\succ \mathbf{0} \implies \mathbf{H} \leftarrow \mathbf{H} + \lambda \mathbf{I}$$

  - Gill-Murray's modified Cholesky (Gill, Murray, and Wright, 2019):

$$\mathbf{H} + \mathbf{E} = \mathbf{L}\mathbf{D}\mathbf{L}^{\intercal} \quad \text{where} \quad \mathbf{D} \succ \mathbf{0}$$

Safeguards/Strategies (continued...)

- Negative curvature direction methods, i.e., find $\mathbf{p}$ s.t. $\langle \mathbf{p}, \mathbf{H}\mathbf{p} \rangle < 0$:

Safeguards/Strategies (continued...)

- Negative curvature direction methods, i.e., find $\mathbf{p}$ s.t. $\langle \mathbf{p}, \mathbf{H}\mathbf{p} \rangle < 0$:
    - Gill-Murray Stable Newton's Method: construct $\mathbf{p}$ using $\mathbf{E}$, $\mathbf{D}$ and $\mathbf{L}$ from the modified Cholesky

Safeguards/Strategies (continued...)

- Negative curvature direction methods, i.e., find $\mathbf{p}$ s.t. $\langle \mathbf{p}, \mathbf{Hp} \rangle < 0$:
    - Gill-Murray Stable Newton's Method: construct **p** using **E**, **D** and **L** from the modified Cholesky
    - Fiacco-McCormick Method (Fiacco and McCormick, 1990): construct **p** using **D** and **L** from "LU-factorization", $\mathbf{LDL}^\intercal$

Safeguards/Strategies (continued...)

- Negative curvature direction methods, i.e., find $\mathbf{p}$ s.t. $\langle \mathbf{p}, \mathbf{Hp} \rangle < 0$:
  - Gill-Murray Stable Newton's Method: construct $\mathbf{p}$ using $\mathbf{E}$, $\mathbf{D}$ and $\mathbf{L}$ from the modified Cholesky
  - Fiacco-McCormick Method (Fiacco and McCormick, 1990): construct $\mathbf{p}$ using $\mathbf{D}$ and $\mathbf{L}$ from "LU-factorization", $\mathbf{LDL}^\intercal$
  - Fletcher-Freeman Method (Fletcher and Freeman, 1977): construct $\mathbf{p}$ based on stable symmetric indefinite factorization due to Bunch and Parlett (1971)

Safeguards/Strategies (continued...)

- Negative curvature direction methods, i.e., find $\mathbf{p}$ s.t. $\langle \mathbf{p}, \mathbf{Hp} \rangle < 0$:
  - Gill-Murray Stable Newton's Method: construct **p** using **E**, **D** and **L** from the modified Cholesky
  - Fiacco-McCormick Method (Fiacco and McCormick, 1990): construct **p** using **D** and **L** from "LU-factorization", $\mathbf{LDL}^\intercal$
  - Fletcher-Freeman Method (Fletcher and Freeman, 1977): construct **p** based on stable symmetric indefinite factorization due to Bunch and Parlett (1971)
- Line search Newton-CG with a safeguard...

Algorithm 7.1 (Line Search Newton–CG).

Given initial point x_0;

for $k = 0, 1, 2, \ldots$

  Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|})\|\nabla f_k\|$;

  Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;

  for $j = 0, 1, 2, \ldots$

    if $d_j^T B_k d_j \leq 0$

      if $j = 0$

        return $p_k = -\nabla f_k$;

      else

        return $p_k = z_j$;

    Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;

    Set $z_{j+1} = z_j + \alpha_j d_j$;

    Set $r_{j+1} = r_j + \alpha_j B_k d_j$;

    if $\|r_{j+1}\| < \epsilon_k$

      return $p_k = z_{j+1}$;

    Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;

    Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;

  end (for)

  Set $x_{k+1} = x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or

    Armijo backtracking conditions (using $\alpha_k = 1$ if possible);

end

Algorithm 7.1 (Line Search Newton–CG).

Given initial point x_0;
for $k = 0, 1, 2, \ldots$
    Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|})\|\nabla f_k\|$;
    Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
    for $j = 0, 1, 2, \ldots$
        if $d_j^T B_k d_j \leq 0$
            if $j = 0$
                return $p_k = -\nabla f_k$;
            else
                return $p_k = z_j$;
        Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
        Set $z_{j+1} = z_j + \alpha_j d_j$;
        Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
        if $\|r_{j+1}\| < \epsilon_k$
            return $p_k = z_{j+1}$;
        Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
        Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
    end (for)
    Set $x_{k+1} = x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or
        Armijo backtracking conditions (using $\alpha_k = 1$ if possible);
end

"Algorithm 7.1 is well suited for large problems, but it has a weakness. When the Hessian is nearly singular, the line search Newton-CG direction can be long and of poor quality, requiring many function evaluations in the line search and giving only a small reduction in the function."

(Nocedal and Wright, 2006)

Is there an optimal solver for symmetric but potentially indefinite/singular system?

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \|\mathbf{g}_k + \mathbf{H}_k \mathbf{p}\|^2$$

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \|\mathbf{g}_k + \mathbf{H}_k \mathbf{p}\|^2 \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{H}_k \left(\mathbf{g}_k + \mathbf{H}_k \mathbf{p}_k^{(t)} \right) \right\rangle = 0$$

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p}\in\mathcal{K}_t(\mathbf{H}_k,\mathbf{g}_k)}{\arg\min} \|\mathbf{g}_k + \mathbf{H}_k\mathbf{p}\|^2 \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{H}_k\left(\mathbf{g}_k + \mathbf{H}_k\mathbf{p}_k^{(t)}\right)\right\rangle = 0$$

$$\implies \langle \mathbf{p}_k^{(t)},\ \mathbf{H}_k\mathbf{g}_k \rangle = -\left\|\mathbf{H}_k\mathbf{p}_k^{(t)}\right\|^2 < 0$$

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p}\in\mathcal{K}_t(\mathbf{H}_k,\mathbf{g}_k)}{\arg\min} \|\mathbf{g}_k + \mathbf{H}_k\mathbf{p}\|^2 \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{H}_k\left(\mathbf{g}_k + \mathbf{H}_k\mathbf{p}_k^{(t)}\right)\right\rangle = 0$$

$$\implies \langle \mathbf{p}_k^{(t)}, \underbrace{\mathbf{H}_k\mathbf{g}_k}_{\nabla(\|\mathbf{g}_k\|^2/2)} \rangle = -\left\|\mathbf{H}_k\mathbf{p}_k^{(t)}\right\|^2 < 0$$

Is there an optimal solver for symmetric but potentially indefinite/singular system? Yes, the Minimum Residual (MINRES) method of Paige and Saunders (1975).

- More complex than CG
- Much less covered in textbook
- Far fewer software libraries
- Optimal rate for all symmetric systems
- Can easily handle inconsistent/indefinite systems
- Every iteration of MINRES is a descent direction for $\|\mathbf{g}\|^2$

$$\mathbf{p}_k^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t(\mathbf{H}_k, \mathbf{g}_k)}{\arg\min} \ \|\mathbf{g}_k + \mathbf{H}_k \mathbf{p}\|^2 \implies \left\langle \mathbf{p}_k^{(t)}, \mathbf{H}_k \left(\mathbf{g}_k + \mathbf{H}_k \mathbf{p}_k^{(t)} \right) \right\rangle = 0$$

$$\implies \langle \mathbf{p}_k^{(t)}, \underbrace{\mathbf{H}_k \mathbf{g}_k}_{\nabla(\|\mathbf{g}_k\|^2/2)} \rangle = - \left\| \mathbf{H}_k \mathbf{p}_k^{(t)} \right\|^2 < 0$$

This category of methods will be referred to as **Newton-MR** methods.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\|\widehat{\mathbf{H}}_k\mathbf{p} + \mathbf{g}_k\right\| \leq \theta \|\mathbf{g}_k\|$

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\| \widehat{\mathbf{H}}_k \mathbf{p} + \mathbf{g}_k \right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex, i.e., $\exists \boldsymbol{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \boldsymbol{\eta}(\mathbf{x},\mathbf{y}), \nabla f(\mathbf{x}) \rangle, \quad \forall \mathbf{x}, \mathbf{y}.$$

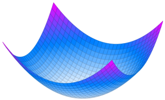
$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\|\widehat{\mathbf{H}}_k\mathbf{p} + \mathbf{g}_k\right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex, i.e., $\exists \boldsymbol{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \boldsymbol{\eta}(\mathbf{x},\mathbf{y}), \nabla f(\mathbf{x}) \rangle, \quad \forall \mathbf{x}, \mathbf{y}.$$

Convex



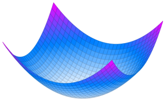
$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

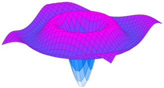
- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\|\widehat{\mathbf{H}}_k\mathbf{p} + \mathbf{g}_k\right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex, i.e., $\exists \boldsymbol{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \boldsymbol{\eta}(\mathbf{x},\mathbf{y}), \nabla f(\mathbf{x})\rangle, \quad \forall \mathbf{x}, \mathbf{y}.$$

Convex



Non-Convex



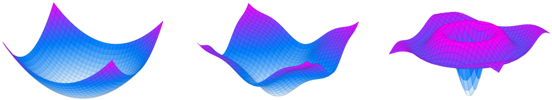
$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\|\widehat{\mathbf{H}}_k\mathbf{p} + \mathbf{g}_k\right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex, i.e., $\exists \boldsymbol{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \boldsymbol{\eta}(\mathbf{x},\mathbf{y}), \nabla f(\mathbf{x})\rangle, \quad \forall \mathbf{x}, \mathbf{y}.$$

Convex Invex Non-Convex

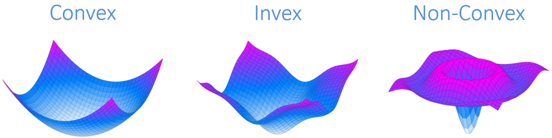


$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right\}$$

Assumptions:

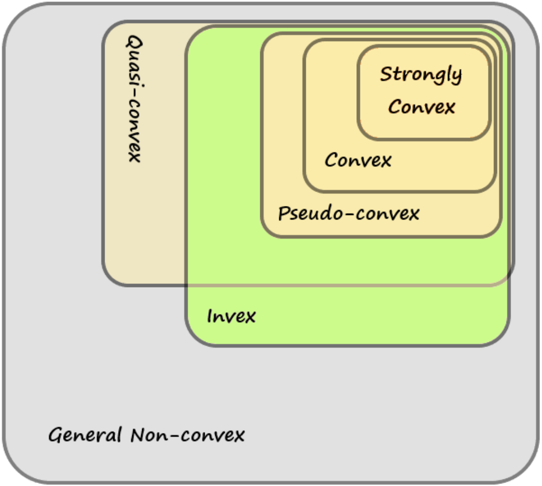
- $\|\mathbf{g}\|^2$ has Lipschitz continuous gradient, i.e., moral smoothness
- $|S|$ is large enough
- $\left\|\widehat{\mathbf{H}}_k\mathbf{p} + \mathbf{g}_k\right\| \leq \theta \|\mathbf{g}_k\|$
- f is ~~strongly convex~~ invex, i.e., $\exists \boldsymbol{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \boldsymbol{\eta}(\mathbf{x},\mathbf{y}), \nabla f(\mathbf{x}) \rangle, \quad \forall \mathbf{x}, \mathbf{y}.$$



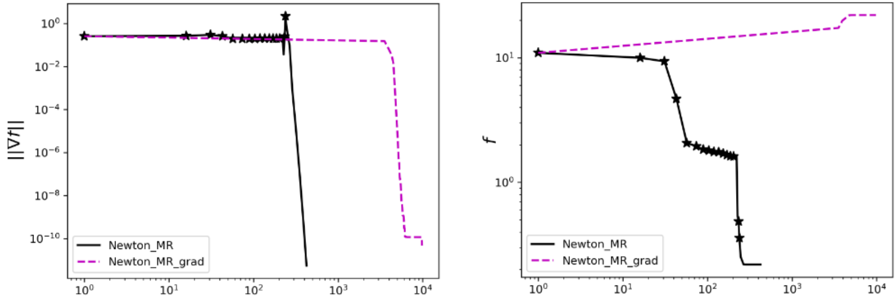
Theorem (Liu and Roosta, 2021)

With high probability, $\|\mathbf{g}_{k+1}\| \leq \rho \|\mathbf{g}_k\|$ *for some* $0 \leq \rho < 1$.


Strongly
Convex
Convex
Pseudo-convex
Quasi-convex
Invex
General Non-convex

What if $f(\mathbf{x})$ is non-convex but non-invex!

What if $f(\mathbf{x})$ is non-convex but non-invex!



A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2$$

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\right\|^2 + \phi \left\|\mathbf{p}\right\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\|\mathbf{g}_k\right\|^2 \quad (?)$$

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\right\|^2 + \phi\left\|\mathbf{p}\right\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta\left\|\mathbf{g}_k\right\|^2 \quad (?)$$

If ✓

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \left\| \mathbf{H}_k \mathbf{p} + \mathbf{g}_k \right\|^2 + \phi \left\| \mathbf{p} \right\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\| \mathbf{g}_k \right\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

A potential “black-box” remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \|\mathbf{g}_k\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

If ✗

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \|\mathbf{g}_k\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

If ✗, $\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2$ s.t. $\langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \|\mathbf{g}_k\|^2$

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \|\mathbf{g}_k\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

If ✗, $\mathbf{p}_k \approx \min_{\mathbf{p}} \|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\|^2 + \phi \|\mathbf{p}\|^2 \text{ s.t. } \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \|\mathbf{g}_k\|^2$

It can be shown that when ✗,

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\right\|^2 + \phi \left\|\mathbf{p}\right\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\|\mathbf{g}_k\right\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

If ✗, $\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\right\|^2 + \phi \left\|\mathbf{p}\right\|^2$ s.t. $\langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\|\mathbf{g}_k\right\|^2$

It can be shown that when ✗,

$$\mathbf{p}_k = -\tilde{\mathbf{H}}_k^{\dagger}\tilde{\mathbf{g}}_k - \lambda_k (\tilde{\mathbf{H}}_{t,i}^{\intercal}\tilde{\mathbf{H}}_k)^{-1}\mathbf{g}_k,$$

$$\lambda_k = \frac{-\langle \tilde{\mathbf{H}}_k^{\dagger}\tilde{\mathbf{g}}_k, \mathbf{g}_k \rangle + \theta \|\mathbf{g}_k\|^2}{\langle (\tilde{\mathbf{H}}_k^{\intercal}\tilde{\mathbf{H}}_k)^{-1}\mathbf{g}_k, \mathbf{g}_k \rangle} > 0.$$

A potential "black-box" remedy (Crane and Roosta, 2020):

$$\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\right\|^2 + \phi \left\|\mathbf{p}\right\|^2 \implies \langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\|\mathbf{g}_k\right\|^2 \quad (?)$$

If ✓, we use $\mathbf{p}_k$

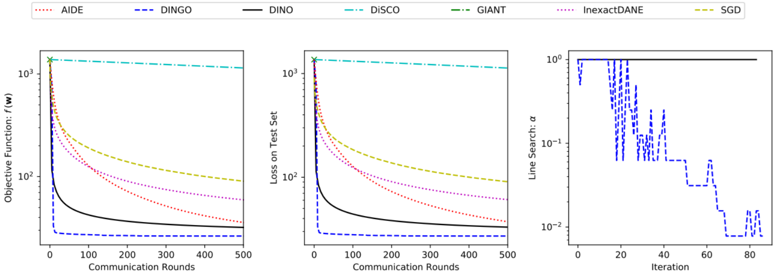
If ✗, $\mathbf{p}_k \approx \min_{\mathbf{p}} \left\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\right\|^2 + \phi \left\|\mathbf{p}\right\|^2$ s.t. $\langle \mathbf{p}, \mathbf{g}_k \rangle \leq -\theta \left\|\mathbf{g}_k\right\|^2$

It can be shown that when ✗,

$$\begin{aligned} \mathbf{p}_k &= -\tilde{\mathbf{H}}_k^{\dagger}\tilde{\mathbf{g}}_k - \lambda_k(\tilde{\mathbf{H}}_{t,i}^{\intercal}\tilde{\mathbf{H}}_k)^{-1}\mathbf{g}_k, \\ \lambda_k &= \frac{-\langle \tilde{\mathbf{H}}_k^{\dagger}\tilde{\mathbf{g}}_k, \mathbf{g}_k \rangle + \theta\|\mathbf{g}_k\|^2}{\langle (\tilde{\mathbf{H}}_k^{\intercal}\tilde{\mathbf{H}}_k)^{-1}\mathbf{g}_k, \mathbf{g}_k \rangle} > 0. \end{aligned}$$

where $\tilde{\mathbf{H}} \triangleq \begin{bmatrix} \mathbf{H} \\ \sqrt{\phi}\mathbf{I} \end{bmatrix}$ and $\tilde{\mathbf{g}} \triangleq \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}$.

Unfortunately, these steps can be of poor quality and the performance of the algorithm may not be competitive in many cases.



Goldfeld, Stephen M, Richard E Quandt, and Hale F Trotter (1966). "Maximization by quadratic hill-climbing". In: *Econometrica: Journal of the Econometric Society*, pp. 541–551.

Goldstein, AA and JF Price (1967). "An effective algorithm for minimization". In: *Numerische Mathematik* 10, pp. 184–189.

Bunch, James R and Beresford N Parlett (1971). "Direct methods for solving symmetric indefinite systems of linear equations". In: *SIAM Journal on Numerical Analysis* 8.4, pp. 639–655.

Paige, Christopher C and Michael A Saunders (1975). "Solution of sparse indefinite systems of linear equations". In: *SIAM journal on numerical analysis* 12.4, pp. 617–629.

Fletcher, Roger and Thomas Leonard Freeman (1977). "A modified Newton method for minimization". In: *Journal of Optimization Theory and Applications* 23, pp. 357–372.

Fiacco, Anthony V and Garth P McCormick (1990). *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM.

Conn, Andrew R, Nicholas IM Gould, and Ph L Toint (2000). *Trust region methods*. Vol. 1. SIAM.

Nocedal, Jorge and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media.

Byrd, Richard H. et al. (2011). "On the use of stochastic Hessian information in optimization methods for machine learning". In: *SIAM Journal on Optimization* 21.3, pp. 977–995.

Tropp, Joel A. (2011). "Improved analysis of the subsampled randomized Hadamard transform". In: *Advances in Adaptive Data Analysis* 3.01n02, pp. 115–126.

— (2012). "User-friendly tail bounds for sums of random matrices". In: *Foundations of Computational Mathematics* 12.4, pp. 389–434.

Björck, Ake (2015). *Numerical methods in matrix computations*. Springer.

Bollapragada, Raghu, Richard H Byrd, and Jorge Nocedal (2018). "Exact and inexact subsampled Newton methods for optimization". In: *IMA Journal of Numerical Analysis* 39.2, pp. 545–578.

Karimireddy, Sai Praneeth, Sebastian U Stich, and Martin Jaggi (2018). "Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients". In: *arXiv preprint arXiv:1806.00413*.

- Gill, Philip E, Walter Murray, and Margaret H Wright (2019). *Practical Optimization*. SIAM.
- Roosta, Fred and Michael W Mahoney (2019). "Sub-sampled Newton methods". In: *Mathematical Programming* 174.1-2, pp. 293–326.
- Crane, Rixon and Fred Roosta (2020). "DINO: Distributed Newton-Type Optimization Method". In: *Proceedings of the International Conference on Machine Learning (ICML-20)*. Vol. 119, pp. 2174–2184.
- Liu, Yang and Fred Roosta (2021). "Convergence of Newton-MR under Inexact Hessian Information". In: *SIAM Journal on Optimization* 31.1, pp. 59–90.
- Cartis, Coralia, Nicholas IM Gould, and Philippe L Toint (2022). *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM.