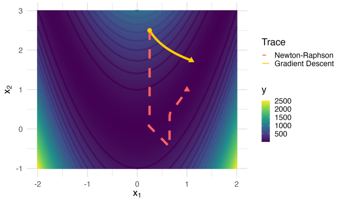
Optimization in Machine Learning

Second order methods

Quasi-Newton



Learning goals

- Newton-Raphson vs. Quasi-Newton
- SR1
- BFGS

QUASI-NEWTON: IDEA

- Start point of **QN method** is (as with NR) a Taylor approximation of the gradient, except that H is replaced by a **pd** matrix $\mathbf{A}^{[t]}$:

$$
\begin{aligned}
\nabla f(\boldsymbol{x}) &\approx \nabla f(\boldsymbol{x}^{[t]}) + \nabla^2 f(\boldsymbol{x}^{[t]})(\boldsymbol{x} - \boldsymbol{x}^{[t]}) = \mathbf{0} && \text{NR} \\
\nabla f(\boldsymbol{x}) &\approx \nabla f(\boldsymbol{x}^{[t]}) + \mathbf{A}^{[t]}(\boldsymbol{x} - \boldsymbol{x}^{[t]}) \qquad\quad = \mathbf{0} && \text{QN}
\end{aligned}
$$

- The update direction:

$$
\begin{aligned}
\boldsymbol{d}^{[t]} &= -\nabla^2 f(\boldsymbol{x}^{[t]})^{-1} \nabla f(\boldsymbol{x}^{[t]}) && \text{NR} \\
\boldsymbol{d}^{[t]} &= -(\mathbf{A}^{[t]})^{-1} \nabla f(\boldsymbol{x}^{[t]}) && \text{QN}
\end{aligned}
$$

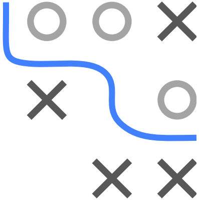
QUASI-NEWTON: IDEA

- Select a starting point $\boldsymbol{x}^{[0]}$ and initialize pd matrix $\mathbf{A}^{[0]}$ (can also be a diagonal matrix - a very rough approximation of Hessian)
- Calculate update direction by solving

$$\mathbf{A}^{[t]}\boldsymbol{d}^{[t]} = -\nabla f(\boldsymbol{x}^{[t]})$$

  and set $\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} + \alpha^{[t]}\boldsymbol{d}^{[t]}$ (Step size through backtracking)
- Calculate an efficient update $\mathbf{A}^{[t+1]}$, based on $\boldsymbol{x}^{[t]}$, $\boldsymbol{x}^{[t+1]}$, $\nabla f(\boldsymbol{x}^{[t]})$, $\nabla f(\boldsymbol{x}^{[t+1]})$ and $\mathbf{A}^{[t]}$

QUASI-NEWTON: IDEA



- Usually the matrices $\mathbf{A}^{[t]}$ are calculated recursively by performing an additive update

$$\mathbf{A}^{[t+1]} = \mathbf{A}^{[t]} + \boldsymbol{B}^{[t]}$$

- How $\boldsymbol{B}^{[t]}$ is constructed is shown on the next slides
- **Requirements** for the matrix sequence $\mathbf{A}^{[t]}$:
  - Symmetric pd, so that $\boldsymbol{d}^{[t]}$ are descent directions
  - Low computational effort when solving LES

$$\mathbf{A}^{[t]}\boldsymbol{d}^{[t]} = -\nabla f(\boldsymbol{x}^{[t]})$$

  - Good approximation of Hessian: The “modified” Taylor series for $\nabla f(\boldsymbol{x})$ (especially for $t \to \infty$) should provide a good approximation

$$\nabla f(\boldsymbol{x}) \approx \nabla f(\boldsymbol{x}^{[t]}) + \mathbf{A}^{[t]}(\boldsymbol{x} - \boldsymbol{x}^{[t]})$$

SYMMETRIC RANK 1 UPDATE (SR1)

- Simplest approach: symmetric rank 1 updates (**SR1**) of form

$$\mathbf{A}^{[t+1]} \leftarrow \mathbf{A}^{[t]} + \boldsymbol{B}^{[t]} = \mathbf{A}^{[t]} + \beta \boldsymbol{u}^{[t]}(\boldsymbol{u}^{[t]})^T$$

  with appropriate vector $\boldsymbol{u}^{[t]} \in \mathbb{R}^n$, $\beta \in \mathbb{R}$

SYMMETRIC RANK 1 UPDATE (SR1)

- **Choice of $\boldsymbol{u}^{[t]}$:** Vectors should be chosen so that the “modified” Taylor series corresponds to the gradient:

$$\nabla f(\boldsymbol{x}) \overset{!}{=} \nabla f(\boldsymbol{x}^{[t+1]}) + \mathbf{A}^{[t+1]}(\boldsymbol{x} - \boldsymbol{x}^{[t+1]})$$

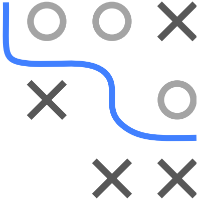
$$\nabla f(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^{[t+1]}) + \left(\mathbf{A}^{[t]} + \beta \boldsymbol{u}^{[t]}(\boldsymbol{u}^{[t]})^T\right) \underbrace{(\boldsymbol{x} - \boldsymbol{x}^{[t+1]})}_{:=\boldsymbol{s}^{[t+1]}}$$

$$\underbrace{\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}^{[t+1]})}_{\boldsymbol{y}^{[t+1]}} = \left(\mathbf{A}^{[t]} + \beta \boldsymbol{u}^{[t]}(\boldsymbol{u}^{[t]})^T\right) \boldsymbol{s}^{[t+1]}$$

$$\boldsymbol{y}^{[t+1]} - \mathbf{A}^{[t]}\boldsymbol{s}^{[t+1]} = \left(\beta (\boldsymbol{u}^{[t]})^T \boldsymbol{s}^{[t+1]}\right) \boldsymbol{u}^{[t]}$$

- For $\boldsymbol{u}^{[t]} = \boldsymbol{y}^{[t+1]} - \mathbf{A}^{[t]}\boldsymbol{s}^{[t+1]}$ and $\beta = \frac{1}{(\boldsymbol{y}^{[t+1]} - \mathbf{A}^{[t]}\boldsymbol{s}^{[t+1]})^T \boldsymbol{s}^{[t+1]}}$ the equation is satisfied

SYMMETRIC RANK 1 UPDATE (SR1)



Advantage

- Provides a sequence of **symmetric pd** matrices
- Matrices can be inverted efficiently and stable using Sherman-Morrison:

$$(\mathbf{A} + \beta \boldsymbol{u}\boldsymbol{u}^T)^{-1} = \mathbf{A} + \beta \frac{\boldsymbol{u}\boldsymbol{u}^T}{1 + \beta \boldsymbol{u}^T \boldsymbol{u}}$$

Disadvantage

- The constructed matrices are not necessarily pd, and the update directions $\boldsymbol{d}^{[t]}$ are therefore not necessarily descent directions

BFGS ALGORITHM

- Instead of Rank 1 updates, the **BFGS** procedure (published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno) uses rank 2 modifications of the form

$$\mathbf{A}^{[t]} + \beta_1 \boldsymbol{u}^{[t]}(\boldsymbol{u}^{[t]})^T + \beta_2 \boldsymbol{v}^{[t]}(\boldsymbol{v}^{[t]})^T$$

  with $\boldsymbol{s}^{[t]} := \boldsymbol{x}^{[t+1]} - \boldsymbol{x}^{[t]}$
- $\boldsymbol{u}^{[t]} = \nabla f(\boldsymbol{x}^{[t+1]}) - \nabla f(\boldsymbol{x}^{[t]})$
- $\boldsymbol{v}^{[t]} = \mathbf{A}^{[t]}\boldsymbol{s}^{[t]}$
- $\beta_1 = \frac{1}{(\boldsymbol{u}^{[t]})^T(\boldsymbol{s}^{[t]})}$
- $\beta_2 = -\frac{1}{(\boldsymbol{s}^{[t]})^T\mathbf{A}^{[t]}\boldsymbol{s}^{[t]}}$
- The resulting matrices $\mathbf{A}^{[t]}$ are positive definite and the corresponding quasi-Newton update directions $\boldsymbol{d}^{[t]}$ are actual descent directions