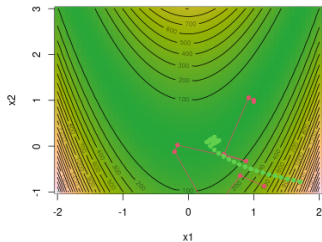


Optimization in Machine Learning

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 $\mathbf{A}^{[t]}$:

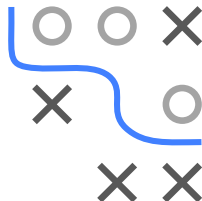
$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{[t]}) + \nabla^2 f(\mathbf{x}^{[t]})(\mathbf{x} - \mathbf{x}^{[t]}) = \mathbf{0} \quad \text{NR}$$

$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{[t]}) + \mathbf{A}^{[t]}(\mathbf{x} - \mathbf{x}^{[t]}) = \mathbf{0} \quad \text{QN}$$

The update direction:

$$\mathbf{d}^{[t]} = -\nabla^2 f(\mathbf{x}^{[t]})^{-1} \nabla f(\mathbf{x}^{[t]}) \quad \text{NR}$$

$$\mathbf{d}^{[t]} = -(\mathbf{A}^{[t]})^{-1} \nabla f(\mathbf{x}^{[t]}) \quad \text{QN}$$



QUASI-NEWTON: IDEA / 2

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

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

and set $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + \alpha^{[t]} \mathbf{d}^{[t]}$ (Step size through backtracking)

- 3 Calculate an efficient update $\mathbf{A}^{[t+1]}$, based on $\mathbf{x}^{[t]}$, $\mathbf{x}^{[t+1]}$, $\nabla f(\mathbf{x}^{[t]})$, $\nabla f(\mathbf{x}^{[t+1]})$ and $\mathbf{A}^{[t]}$.



QUASI-NEWTON: IDEA / 3

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

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

How $\mathbf{B}^{[t]}$ is constructed is shown on the next slides.

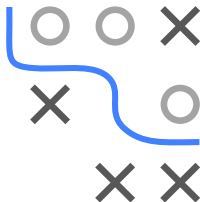
Requirements for the matrix sequence $\mathbf{A}^{[t]}$:

- 1 Symmetric pd, so that $\mathbf{d}^{[t]}$ are descent directions.
- 2 Low computational effort when solving LES

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

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

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



SYMMETRIC RANK 1 UPDATE (SR1)

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

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

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



SYMMETRIC RANK 1 UPDATE (SR1) / 2

Choice of $\mathbf{u}^{[t]}$:

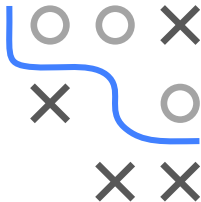
Vectors should be chosen so that the “modified” Taylor series corresponds to the gradient:

$$\begin{aligned}\nabla f(\mathbf{x}) &\stackrel{!}{=} \nabla f(\mathbf{x}^{[t+1]}) + \mathbf{A}^{[t+1]}(\mathbf{x} - \mathbf{x}^{[t+1]}) \\ \nabla f(\mathbf{x}) &= \nabla f(\mathbf{x}^{[t+1]}) + \left(\mathbf{A}^{[t]} + \beta \mathbf{u}^{[t]}(\mathbf{u}^{[t]})^\top \right) \underbrace{(\mathbf{x} - \mathbf{x}^{[t+1]})}_{:= \mathbf{s}^{[t+1]}}\end{aligned}$$

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

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

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



SYMMETRIC RANK 1 UPDATE (SR1) / 3

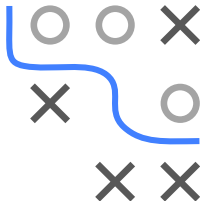
Advantage

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

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

Disadvantage

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

$$\mathbf{A}^{[t]} + \beta \mathbf{u}^{[t]} (\mathbf{u}^{[t]})^\top + \beta \mathbf{v}^{[t]} (\mathbf{v}^{[t]})^\top$$

with $\mathbf{s}^{[t]} := \mathbf{x}^{[t+1]} - \mathbf{x}^{[t]}$

- $\mathbf{u}^{[t]} = \nabla f(\mathbf{x}^{[t+1]}) - \nabla f(\mathbf{x}^{[t]})$
- $\mathbf{v}^{[t]} = \mathbf{A}^{[t]} \mathbf{s}^{[t]}$
- $\beta = \frac{1}{(\mathbf{u}^{[t]})^\top (\mathbf{s}^{[t]})}$
- $\beta = -\frac{1}{(\mathbf{s}^{[t]})^\top \mathbf{A}^{[t]} \mathbf{s}^{[t]}}$

The resulting matrices $\mathbf{A}^{[t]}$ are positive definite and the corresponding quasi-newton update directions $\mathbf{d}^{[t]}$ are actual descent directions.

