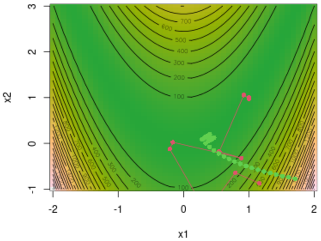
Optimization in Machine Learning

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

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

The update direction:

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

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

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

   and set $\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} + \alpha^{[t]}\boldsymbol{d}^{[t]}$ (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 $\boldsymbol{A}^{[t]}$ are calculated recursively by performing an additive update

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

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

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

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

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

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

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

SYMMETRIC RANK 1 UPDATE (SR1)

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

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

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

SYMMETRIC RANK 1 UPDATE (SR1) / 2

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

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

For $\boldsymbol{u}^{[t]} = \boldsymbol{y}^{[t+1]} - \boldsymbol{A}^{[t]}\boldsymbol{s}^{[t+1]}$ and $\beta = \frac{1}{\left(\boldsymbol{y}^{[t+1]} - \boldsymbol{A}^{[t]}\boldsymbol{s}^{[t+1]}\right)^\top \boldsymbol{s}^{[t+1]}}$ 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:

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

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

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

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

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

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