Optimization in Machine Learning

Second order methods: Newton-Raphson

Learning goals

- First vs. Second order methods
- Newton-Raphson

FIRST AND SECOND ORDER PROCEDURES

So far we have considered algorithms based on the gradient (1st derivative). These methods are called **first-order methods**.

In the following we consider procedures based on the Hessian matrix (2nd derivative). These are called **second-order methods**.

NEWTON-RAPHSON METHOD

Assumption: Function f is twice differentiable, i.e. the Hessian matrix $\nabla^2 f(\boldsymbol{x})$ can be calculated.

Aim: Find stationary point

$$\nabla f(\boldsymbol{x}) \quad = \quad \mathbf{0}$$

Idea: Find root of Taylor approximation (1st order) of $\nabla f(\boldsymbol{x})$:

$$\begin{aligned}
\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{[t]}) + \nabla^2 f(\mathbf{x}^{[t]})(\mathbf{x} - \mathbf{x}^{[t]}) \quad &= \quad \mathbf{0} \\
\nabla^2 f(\mathbf{x}^{[t]})(\mathbf{x} - \mathbf{x}^{[t]}) \quad &= \quad -\nabla f(\mathbf{x}^{[t]}) \\
\mathbf{x}^{[t+1]} \quad &= \quad \mathbf{x}^{[t]} \underbrace{- \left(\nabla^2 f(\mathbf{x}^{[t]})\right)^{-1} \nabla f(\mathbf{x}^{[t]})}_{:=d^{[t]}}
\end{aligned}$$

This motivates the update $\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} + \boldsymbol{d}^{[t]}$ with update direction $\boldsymbol{d}^{[t]} = -\left(\nabla^2 f(\mathbf{x}^{[t]})\right)^{-1} \nabla f(\mathbf{x}^{[t]})$.

NEWTON-RAPHSON METHOD

Example:

$$f(x, y) = \left(x^2 + \frac{y^2}{2}\right)$$

Update direction:

$$\mathbf{d}^{[t]} = -\left(\nabla^2 f(x^{[t]}, y^{[t]})\right)^{-1} \nabla f(x^{[t]}, y^{[t]})$$

$$\begin{aligned} \nabla f(x, y) &= \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} 2x \\ y \end{pmatrix} \\ \nabla^2 f(x, y) &= \begin{pmatrix} \frac{\partial^2 f(x,y)}{\partial^2 x} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial y \partial x} & \frac{\partial^2 f(x,y)}{\partial^2 y} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

NEWTON-RAPHSON METHOD

`t = 1`:

$$
\begin{aligned}
\begin{pmatrix} x^{[1]} \\ y^{[1]} \end{pmatrix} &= \begin{pmatrix} x^{[0]} \\ y^{[0]} \end{pmatrix} + \mathbf{d}^{[0]} = \begin{pmatrix} x^{[0]} \\ y^{[0]} \end{pmatrix} - \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2x^{[0]} \\ y^{[0]} \end{pmatrix} \\
&= \begin{pmatrix} x^{[0]} \\ y^{[0]} \end{pmatrix} + \begin{pmatrix} -x^{[0]} \\ -y^{[0]} \end{pmatrix} \\
&= \mathbf{0}
\end{aligned}
$$

Newton-Raphson only needs one iteration to solve the problem!

NEWTON-RAPHSON METHOD

Advantages:

- If the function is sufficiently smooth, the procedure converges quadratically locally (i.e. if the starting point is close enough to optimum)

Disadvantages

- At “bad” starting points the procedure may not converge at all
- The Hessian must be calculated and the direction of descent determined by solving a system of equations

NEWTON-RAPHSON METHOD

Problem 1: The update direction

$$\mathbf{d}^{[t]} = -\left(\nabla^2 f(\mathbf{x}^{[t]})\right)^{-1} \nabla f(\mathbf{x}^{[t]})$$

is generally not a direction of descent.

But: If the Hessian matrix is positive definite, it is necessarily a descent direction:

$$(\mathbf{d}^{[t]})^\top \nabla f(\mathbf{x}^{[t]}) = -\left(\nabla f(\mathbf{x}^{[t]})\right)^\top \left(\nabla^2 f(\mathbf{x}^{[t]})\right)^{-1} \nabla f(\mathbf{x}^{[t]}) < 0.$$

Near the minimum, the Hessian matrix is positive definite. But especially at the beginning the Hessian is often not positive definite and the Newton-Raphson update direction is not sensible.

NEWTON-RAPHSON METHOD

Problem 2: The calculation of the Hessian can be **expensive** and the calculation of the descent direction by solving the system of equations

$$\left(\nabla^2 f(\mathbf{x}^{[t]})\right) \mathbf{d}^{[t]} = -\nabla f(\mathbf{x}^{[t]})$$

can be numerically unstable.

Aim: Find methods that can be applied without the Hessian matrix

- Quasi-Newton method.
- Gauss-Newton algorithm (for least squares).