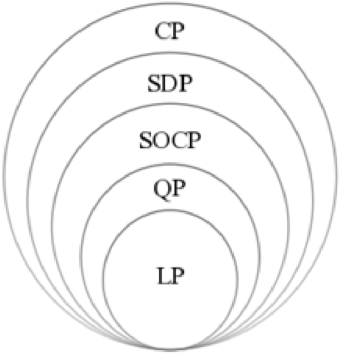
Optimization in Machine Learning

Nonlinear programs
Solvers



Learning goals

- Definition
- Max. Likelihood
- Normal regression
- Risk Minimization

SEQUENTIAL QUADRATIC PROGRAMMING

For simplification, we consider only equality constraints, thus problems of the form

$$\min f(\mathbf{x}) \qquad \text{s.t.} \quad h(\mathbf{x}) = 0.$$

Idea:

- Instead of f we optimize the 2nd order Taylor approximation in a point $\tilde{\boldsymbol{x}}$

$$\tilde{f}(\mathbf{x}) = f(\tilde{\boldsymbol{x}}) + \nabla_x f(\tilde{\boldsymbol{x}})^T(\mathbf{x} - \tilde{\boldsymbol{x}}) + \frac{1}{2}(\mathbf{x} - \tilde{\boldsymbol{x}})^T \nabla^2_{xx} f(\tilde{\boldsymbol{x}})(\mathbf{x} - \tilde{\boldsymbol{x}})$$

- h is also replaced by its linear approximation in $\tilde{\boldsymbol{x}}$.

$$\tilde{h}(\mathbf{x}) = h(\tilde{\boldsymbol{x}}) + \nabla h(\tilde{\boldsymbol{x}})^T(\mathbf{x} - \tilde{\boldsymbol{x}}).$$

SEQUENTIAL QUADRATIC PROGRAMMING / 2

With $\boldsymbol{d} := (\mathbf{x} - \tilde{\boldsymbol{x}})$ we formulate the **quadratic auxiliary problem**

$$
\begin{aligned}
\min_{\boldsymbol{d}} \quad & \tilde{f}(\boldsymbol{d}) := f(\tilde{\boldsymbol{x}}) + \boldsymbol{d}^T \nabla_x f(\tilde{\boldsymbol{x}}) + \frac{1}{2}\boldsymbol{d}^T \nabla^2_{xx} f(\tilde{\boldsymbol{x}})\boldsymbol{d} \\
\text{s.t.} \quad & \tilde{h}(\boldsymbol{d}) := h(\tilde{\boldsymbol{x}}) + \nabla h(\tilde{\boldsymbol{x}})^T \boldsymbol{d} = 0.
\end{aligned}
$$

Even if no conditions for optimality can be formulated for the actual optimization problem, the KKT conditions apply in an optimum of this problem necessarily.

If the matrix $\nabla^2_{xx} f(\mathbf{x})$ is positive semidefinite, it is a **convex optimization problem**.

SEQUENTIAL QUADRATIC PROGRAMMING / 3

Using the Lagrange function

$$L(\boldsymbol{d}, \beta) = \boldsymbol{d}^T \nabla_x f(\tilde{\boldsymbol{x}}) + \frac{1}{2}\boldsymbol{d}^T \nabla^2_{xx} f(\tilde{\boldsymbol{x}})\boldsymbol{d} + \beta^T (h(\tilde{\boldsymbol{x}}) + \nabla h(\tilde{\boldsymbol{x}})^T \boldsymbol{d})$$

we formulate the KKT conditions

- $\nabla_{\boldsymbol{d}} L(\boldsymbol{d}, \beta) = \nabla_x f(\tilde{\boldsymbol{x}}) + \nabla^2_{xx} f(\tilde{\boldsymbol{x}})\boldsymbol{d} + \nabla h(\tilde{\boldsymbol{x}})^T \beta = 0$
- $h(\tilde{\boldsymbol{x}}) + \nabla h(\tilde{\boldsymbol{x}})^T \boldsymbol{d} = 0$

or in matrix notation

$$\begin{pmatrix} \nabla^2_{xx} f(\tilde{\boldsymbol{x}}) & \nabla h(\tilde{\boldsymbol{x}})^T \\ \nabla h(\tilde{\boldsymbol{x}}) & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ \beta \end{pmatrix} = - \begin{pmatrix} \nabla_x f(\tilde{\boldsymbol{x}}) \\ h(\tilde{\boldsymbol{x}}) \end{pmatrix}$$

The solution of the **quadratic subproblem** can thus be traced back to the solution of a linear system of equations.

SEQUENTIAL QUADRATIC PROGRAMMING / 4

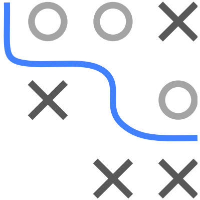
Algorithm SQP for problems with equality constraints

Select a feasible starting point $\mathbf{x}^{(0)} \in \mathbb{R}^n$
while Stop criterion not fulfilled **do**
    Solve quadratic subproblem by solving the equation

$$\begin{pmatrix} \nabla^2_{xx} L(\mathbf{x}, \boldsymbol{\mu}) & \nabla h(\mathbf{x})^T \\ \nabla h(\mathbf{x}) & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{\beta} \end{pmatrix} = - \begin{pmatrix} \nabla_x L(\mathbf{x}, \boldsymbol{\mu}) \\ h(\mathbf{x}) \end{pmatrix}$$

    Set $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \boldsymbol{d}$
end while

PENALTY METHODS



Idea: Replace the constrained Optimization problem with a sequence of unconstrained optimization problems using a **penalty function**.

Instead of looking at

$$\min f(\mathbf{x}) \qquad \text{s.t.} \quad h(\mathbf{x}) = 0.$$

we look at the unconstrained optimization problem

$$\min_{\mathbf{x}} p(\mathbf{x}) = f(\mathbf{x}) + \rho \frac{\|h(\mathbf{x})\|^2}{2}.$$

Under appropriate conditions it can be shown that the solutions of the problem for $\rho \to \infty$ converge against the solution of the initial problem.

BARRIER METHOD

Idea: Establish a “barrier” that penalizes if $\mathbf{x}$ comes too close to the edge of the allowed set $\boldsymbol{S}$. For the problem

$$\min f(\mathbf{x}) \qquad \text{s.t.} \quad g(\mathbf{x}) \leq 0$$

a common **Barrier function** is

$$B_\rho = f(\mathbf{x}) - \rho \sum_{i=1}^{m} \ln(-g_i(\mathbf{x}))$$

The penalty term becomes larger, the closer $\mathbf{x}$ comes to 0, i.e. the limit of the feasible set. Under certain conditions, the solutions of $\min B_\rho$ for $\rho \to 0$ converge against the optimum of the original problem.

The procedure is also called **interior-point method**.

Constrained Optimization in R

CONSTRAINED OPTIMIZATION IN R

- The function **optim(..., method = "L-BFGS-B")** uses quasi-newton methods and can handle box constraints.
- The function **nlminb()** uses trust-region procedures and can also handle box constraints.
- **constrOptim()** can be used for optimization problems with linear inequality conditions and is based on interior-point methods.
- **nloptr** is an interface to **NLopt**, an open-source library for nonlinear optimization (`https://nlopt.readthedocs.io/en/latest/`)