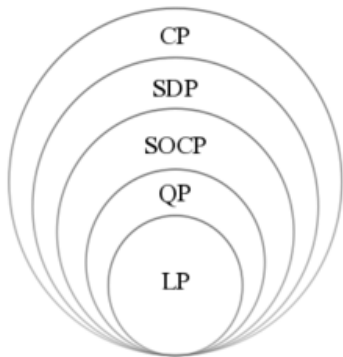


Nonlinear programs

Solvers



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



SEQUENTIAL QUADRATIC PROGRAMMING

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

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

Idea:

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

$$\tilde{f}(\mathbf{x}) = f(\tilde{\mathbf{x}}) + \nabla_{\mathbf{x}} f(\tilde{\mathbf{x}})^T (\mathbf{x} - \tilde{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \tilde{\mathbf{x}})^T \nabla_{\mathbf{xx}}^2 f(\tilde{\mathbf{x}}) (\mathbf{x} - \tilde{\mathbf{x}})$$

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

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



SEQUENTIAL QUADRATIC PROGRAMMING / 2

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

$$\begin{array}{ll}\min_{\mathbf{d}} & \tilde{f}(\mathbf{d}) := f(\tilde{\mathbf{x}}) + \mathbf{d}^T \nabla_{\mathbf{x}} f(\tilde{\mathbf{x}}) + \frac{1}{2} \mathbf{d}^T \nabla_{\mathbf{xx}}^2 f(\tilde{\mathbf{x}}) \mathbf{d} \\ \text{s.t.} & \tilde{h}(\mathbf{d}) := h(\tilde{\mathbf{x}}) + \nabla h(\tilde{\mathbf{x}})^T \mathbf{d} = 0.\end{array}$$

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 $\nabla_{\mathbf{xx}}^2 f(\mathbf{x})$ is positive semidefinite, it is a **convex optimization problem**.



SEQUENTIAL QUADRATIC PROGRAMMING / 3

Using the Lagrange function

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

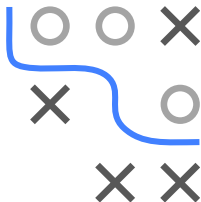
we formulate the KKT conditions

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

or in matrix notation

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

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

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

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

- 4: Set $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \mathbf{d}$
 - 5: **end while**
-



PENALTY METHODS

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

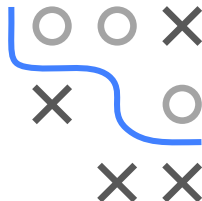
Instead of looking at

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

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 $\rho \rightarrow \infty$ 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 \mathbf{S} . For the problem

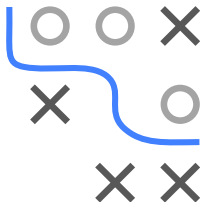
$$\min f(\mathbf{x}) \quad \text{s.t.} \quad g(\mathbf{x}) \leq 0$$

a common **Barrier function** is

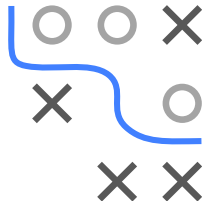
$$B_\rho = f(\mathbf{x}) - \rho \sum_{i=1}^m \ln(-g_i(\mathbf{x}))$$

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 $\rho \rightarrow 0$ 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 **nlinb()** 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/>)

