Newton-Type Optimization with Constraints

Moritz Diehl

Systems Control and Optimization Laboratory
Faculty of Engineering & Faculty of Mathematics and Physics
University of Freiburg, Germany

July 17, 2017

Overview

- Equality Constrained Optimization
- Optimality Conditions and Multipliers
- Newton's Method
- Constrained Gauss Newton Method
- Inequality Constraints
- Interior Point Methods
- Sequential Quadratic Programming

General Nonlinear Program (NLP)

Nonlinear Program (NLP)

$$\boxed{\min_{x} \quad f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{lcl} g(x) & = & 0 \\ h(x) & \leq & 0 \end{array} \right.}$$

First treat case without inequalities:

$$\boxed{\min_{x} \quad f(x) \quad \text{s.t.} \quad g(x) \quad = \quad 0}$$

Lagrange Function and Optimality Conditions

$$\boxed{\min_{x} \quad f(x) \quad \text{s.t.} \quad g(x) = 0}$$

Lagrangian function:

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x)$$

KKT conditions: for optimal x^* exist multipliers λ^* such that

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*) &= 0 \\ g(x^*) &= 0 \end{aligned}$$

Simple Example (quadratic)

$$\min_{x_1,x_2} \quad x_1^2 + 6x_2^2 \quad \text{s.t.} \quad 3x_1 + x_2 + 1 \quad = \quad 0$$

Lagrangian function:

$$\mathcal{L}(x,\lambda) = x_1^2 + 6x_2^2 + \lambda(3x_1 + x_2 + 1)$$

KKT conditions:

$$\nabla_x \mathcal{L}(x,\lambda) = \quad \begin{bmatrix} 2x_1 \\ 12x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 1 \end{bmatrix} \lambda \quad = 0$$

$$g(x) = \qquad 3x_1 + x_2 + 1 \qquad = 0$$

linear system, can e.g. be solved by elimination: $\lambda = (2/3)x_1$, $x_2 = (1/12)\lambda = (1/18)x_1$, $3x_1 + (1/18)x_1 = 1$, i.e., $x_1 = (18/55)$.

General "Quadratic Programs" (QP)

$$\min_{x} \quad c^\top x + \frac{1}{2} x^\top B x \quad \text{s.t.} \quad Ax + b \; = \; 0$$

Lagrangian function:

$$\mathcal{L}(x, \lambda) = c^\top x + \frac{1}{2} x^\top B x + \lambda^\top (Ax + b)$$

KKT conditions:

$$\nabla_x \mathcal{L}(x, \lambda) = \quad c + Bx + A^\top \lambda \quad = 0$$

$$g(x) = \qquad Ax + b \qquad = 0$$

equivalent linear system

$$\begin{bmatrix} c \\ b \end{bmatrix} + \begin{bmatrix} B & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = 0$$

can be solved easily

Simple Example (nonlinear)

$$\boxed{\min_{x_1,x_2} \quad x_1^2 + 6x_2^2 \quad \text{s.t.} \quad 3x_1 + x_2^3 + 1 \quad = \quad 0}$$

Lagrangian function:

$$\mathcal{L}(x,\lambda) = x_1^2 + 6x_2^2 + \lambda(3x_1 + x_2^3 + 1)$$

KKT conditions:

$$\nabla_x \mathcal{L}(x,\lambda) = \quad \begin{bmatrix} 2x_1 \\ 12x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 3x_2^2 \end{bmatrix} \lambda \quad = 0$$

$$g(x) = \qquad 3x_1 + x_2^3 + 1 \qquad = 0$$

nonlinear system, can be solved by Newton-type method

Newton's Method on Optimality Conditions

To solve nonlinear KKT equations

$$\begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ g(x) \end{bmatrix} = 0$$

linearize them at a guess $(x_{[k]}, \lambda_{[k]})$

$$\begin{bmatrix} \nabla_x \mathcal{L}(x_{[k]},\lambda_{[k]}) \\ g(x_{[k]}) \end{bmatrix} + \begin{bmatrix} B_{[k]} & A_{[k]}^\top \\ A_{[k]} & 0 \end{bmatrix} \begin{bmatrix} x - x_{[k]} \\ \lambda - \lambda_{[k]} \end{bmatrix} = 0$$

with $B_{[k]} = \nabla_x^2 \mathcal{L}(x_{[k]},\lambda_{[k]})$ and $A_{[k]} = \nabla_x g(x_{[k]})^\top$

Solution is used to obtain next linearization point $(x_{[k+1]}, \lambda_{[k+1]})$

Newton Step = Quadratic Program

Linear system equivalent to a quadratic program (QP):

$$\begin{array}{ll} \min\limits_{x} & \nabla f(x_{[k]})^\top x + \frac{1}{2}(x - x_{[k]})^\top B_{[k]} (x - x_{[k]}) \\ \text{s.t.} & g(x_{[k]}) + A_{[k]}(x - x_{[k]}) \quad = \quad 0, \end{array}$$

Newton iterate $(x_{[k+1]}, \lambda_{[k+1]})$ is solution of above QP.

Newton-Type Optimization Methods

Newton-Type Optimization Methods replace the exact Hessian of the Lagrangian by an approximation $B_{[k]} \approx \nabla_x^2 \mathcal{L}(x_{[k]}, \lambda_{[k]})$

Examples: BFGS, Gauss-Newton, ...

Constrained Gauss-Newton Method

In special case of least squares objectives

$$\begin{array}{ll} \min\limits_{x} & \frac{1}{2}\|R(x)\|_2^2 \\ \text{s.t.} & g(x) \;=\; 0 \end{array}$$

can approximate Hessian by

$$B_{[k]} = \nabla R(x_{[k]}) \nabla R(x_{[k]})^\top$$

or equivalently, solve the following QP in each Newton-type iteration

$$\begin{array}{ll} \min\limits_{x} & \frac{1}{2}\|R(x_{[k]}) + \nabla R(x_{[k]})^\top (x - x_{[k]})\|_2^2 \\ \text{s.t.} & g(x_{[k]}) + \nabla g(x_{[k]})^\top (x - x_{[k]}) \;=\; 0 \end{array}$$

Linear convergence rate. Fast convergence if $\|R(x^*)\|$ small.

NLP with Inequalities

Regard again NLP with both, equalities and inequalities:

$$\min_{x} \quad f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{lcl} g(x) & = & 0, \\ h(x) & \leq & 0. \end{array} \right.$$

Introduce Lagrangian function

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x)$$

Optimality Conditions with Inequalities

THEOREM(Karush-Kuhn-Tucker (KKT) conditions)
For an optimal solution x^* exist multipliers λ^* and μ^* such that

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) &= 0 \\ g(x^*) &= 0 \\ h(x^*) &\leq 0 \\ \mu^* &\geq 0 \\ h(x^*)^\top \mu^* &= 0 \end{aligned}$$

Last three conditions are called "complementarity conditions", the make the system non-smooth and difficult to solve.
Two important classes of Newton-type methods are:

- Interior Point (IP) methods
- Sequential Quadratic Programming (SQP) methods

Interior Point Methods

Step 1: Problem reformulation using slack variables s

$$\min_{x,s} \quad f(x) \quad \text{s.t.} \quad \begin{cases} g(x) &= 0 \\ h(x) + s &= 0 \\ s &\geq 0 \end{cases}$$

Step 2: Approximate problem by *barrier problem*

$$\min_{x,s} \quad f(x) + \phi(s,\tau) \quad \text{s.t.} \quad \begin{cases} g(x) &= 0 \\ h(x) + s &= 0 \end{cases}$$

with $\tau > 0$ and *logarithmic barrier function* $\phi(s,\tau) = -\tau \sum_{i=1}^{n_h} \log(s_i)$

Interior Point Methods

Barrier problem

$$\min_{x,s} \quad f(x) + \phi(s,\tau) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} g(x) & = & 0 \\ h(x) + s & = & 0 \end{array} \right.$$

KKT conditions of barrier problem equivalent to

$$\begin{array}{rcl} \nabla_x \mathcal{L}(x,\lambda,\mu) & = & 0 \\ g(x) & = & 0 \\ h(x) + s & = & 0 \\ s_i \mu_i - \tau & = & 0 \quad \text{for} \quad i = 1, \ldots, n_h \end{array}$$

Smooth root finding problem, can be solved by Newton's method.
If barrier parameter τ shrinks to zero, original solution is recovered.

Interior Point Algorithm (Sketch)

1. Choose large $\tau_{[0]}$ and initial guess $(x_{[0]}, \lambda_{[0]}, \mu_{[0]}, s_{[0]})$
2. Loop for $k = 0, 1, 2, \ldots$
   - Starting at $(x_{[k]}, \lambda_{[k]}, \mu_{[k]}, s_{[k]})$, perform one or multiple Newton-type optimization iterations to solve

$$\boxed{\min_{x,s} \quad f(x) + \phi(s, \tau_{[k]}) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} g(x) & = & 0 \\ h(x) + s & = & 0 \end{array} \right.}$$

   - store solution as next initial guess $(x_{[k+1]}, \lambda_{[k+1]}, \mu_{[k+1]}, s_{[k+1]})$
   - if $\tau_{[k]}$ small enough stop iterations
     otherwise decrease barrier to $\tau_{[k+1]} < \tau_{[k]}$ and continue the loop
3. Result: high accuracy solution of original NLP

Note: an algorithm of this type is implemented in the solver `ipopt`

Alternative: Sequential Quadratic Programming (SQP)

Linearizing all functions of the NLP, one obtains a QP:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & \nabla f(x_{[k]})^\top x + \frac{1}{2}(x - x_{[k]})^\top A_{[k]} (x - x_{[k]}) \\ \text{s.t.} & \left\{ \begin{array}{rcl} g(x_{[k]}) + \nabla g(x_{[k]})^\top (x - x_{[k]}) & = & 0, \\ h(x_{[k]}) + \nabla h(x_{[k]})^\top (x - x_{[k]}) & \leq & 0, \end{array} \right. \end{array}$$

with

$$A_{[k]} = \nabla_x^2 \mathcal{L}(x_{[k]}, \lambda_{[k]}, \mu_{[k]})$$

The solution of the QP delivers the next SQP iterate

$$x_{[k+1]}, \quad \lambda_{[k+1]}, \quad \mu_{[k+1]}$$

SQP is implemented e.g. in the code `SNOPT`. SQP is better at warmstarting, but often still slower than IP methods.

How to solve QP subprojects within SQP?

- Interior point methods - as explained, but tailored to QP (e.g. `ooqp`, `hpmpc`, `hpipm`, `FORCES PRO`)
- Active Set Methods - similar to simplex for Linear Programming (e.g. `qpOASES`)
- First order methods (e.g. `FiOrdOs`)
- ...

Summary Newton-type optimization

- Newton-type optimization solves the necessary optimality conditions
- Newton's method linearizes the nonlinear system in each iteration
- for constraints, need Lagrangian function, and KKT conditions
- for equalities KKT conditions are smooth, can apply Newton's method
- for inequalities KKT conditions are non-smooth, can apply Interior Point (IP) or Sequential Quadratic Programming (SQP) methods