Nonlinear Programming

January 1, 2023

Nonlinear Programming

General Formulation

$$\min_{x \in \mathbb{R}^n} \psi(\mathbf{x}), \ \text{s.t.} \quad \begin{array}{c} \mathbf{c}_l \leq c(\mathbf{x}) \leq \mathbf{c}_u \\ \mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array} \tag{1}$$

- $\psi(\mathbf{x})$ is a general smooth (twice continuously differentiable) function with gradient $\mathbf{g}(\mathbf{x})$ and Hessian $\mathbf{H}(\mathbf{x})$.
- $c(\mathbf{x})$ are general nonlinear constraint functions, assume $c(\mathbf{x})$ are twice differentiable as well.
- $\mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u$ are general linear constraints.
- $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ are simple bound constraints.

Nonlinear Programming

Feasible Direction

- for linearly constrained problems, active set method searches a null space direction $\mathbf{p}$ wrt working set $\mathcal{W}$ such that $\mathbf{A}_{\mathcal{W}}(\mathbf{x}_k + \alpha\mathbf{p}) = \mathbf{A}_{\mathcal{W}}\mathbf{x}_k$.
- for nonlinear constraints, in order to retain the equality $c_{\mathcal{W}}(\mathbf{x}) = c_{\mathcal{W}}(\mathbf{x}_k)$, we need move along a feasible arc $\mathbf{x}(t) = (x_0(t), x_1(t), \cdots, x_{n-1}(t))$.
- let $\mathbf{p}$ be the tangent vector to the arc, $\frac{d}{dt}c_{\mathcal{W}}(\mathbf{x}_k) = \mathbf{J}_{\mathcal{W}}(\mathbf{x}_k)\mathbf{p} = 0$, in other words $\mathbf{p}$ is a null space direction wrt the Jacobian $\mathbf{J}_{\mathcal{W}}(\mathbf{x}_k)$ of active constraints $c_{\mathcal{W}}(\mathbf{x}_k)$.

Nonlinear Programming

Simplified Formulation using Active Set Method

- Mathwrist directly solves the general formulation (1), implementation based on SNOPT method [4].
- For brevity of discussion and without loss of generality, here we are looking at a simplified form

$$\min_{x \in \mathbb{R}^n} \psi(\mathbf{x}), \text{ s.t. } c(\mathbf{x}) \geq 0 \tag{2}$$

- we can include linear constraints as a part of $c(\mathbf{x})$, since the Jacobian of $\mathbf{Ax}$ is just $\mathbf{A}$.
- please refer to our linear programming (LP) and quadratic programming (QP) documentation for the details of handling constraint upper bounds and simple bound constraints.

Nonlinear Programming

Equality Constrained QP Problem

Moving along a feasible arc $\mathbf{x}(t)$ wrt a fixed working set $\mathcal{W}$, we can approximate the objective function as

$$\psi(\mathbf{x}(t)) \approx \psi(\mathbf{x}_k) + t\mathbf{g}^T(\mathbf{x}_k)\mathbf{p} + \frac{1}{2}t^2\mathbf{p}^T \bigtriangledown_{xx} \mathcal{L}(\mathbf{x}_k)\mathbf{p} \tag{3}$$

, where $\bigtriangledown_{xx}\mathcal{L}(\mathbf{x}_k)$ is the Hessian matrix of the Lagrangian function $\mathcal{L}(\mathbf{x}, \lambda)$ wrt $\mathbf{x}$,

$$\mathcal{L}(\mathbf{x}, \lambda) = \psi(\mathbf{x}) - \lambda^T c_{\mathcal{W}}(\mathbf{x}) \tag{4}$$

Define vector $\mathbf{d} = t\mathbf{p}$. Minimizing (3) is to solve an equality constrained QP

$$\min_{d \in \mathbb{R}^n} \mathbf{g}^T(\mathbf{x}_k)\mathbf{d} + \frac{1}{2}\mathbf{d}^T \bigtriangledown_{xx} \mathcal{L}(\mathbf{x}_k)\mathbf{d} \text{ s.t. } \mathbf{J}_{\mathcal{W}}(\mathbf{x}_k)\mathbf{d} = 0 \tag{5}$$

Nonlinear Programming

Sequential Quadratic Programming (SQP)

At the k-th major iteration, define

$$\begin{aligned}\mathbf{J}_k &:= \mathbf{J}(\mathbf{x}_k)\\ \mathbf{c}_k &:= c(\mathbf{x}_k)\\ \mathbf{g}_k &:= \mathbf{g}(\mathbf{x}_k)\\ \bigtriangledown_{xx}\mathcal{L}_k &:= \bigtriangledown_{xx}\mathcal{L}(\mathbf{x}_k)\end{aligned}$$

- linearize all nonlinear constraints $c(\mathbf{x})$ to $\hat{c}(\mathbf{x}) = \mathbf{c}_k + \mathbf{J}_k(\mathbf{x} - \mathbf{x}_k)$.
- approximate $c(\mathbf{x}) \geq 0$ by $\hat{c}(\mathbf{x}) \geq 0$, equivalently $\mathbf{J}_k\mathbf{d} \geq -\mathbf{c}_k$ for $\mathbf{d} = \mathbf{x} - \mathbf{x}_k$.

Nonlinear Programming

Sequential Quadratic Programming (SQP continued)

- formulate and solve sub QP problems in minor iterations,

$$\min_{d\in\mathbb{R}^n} \mathbf{g}_k^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T \bigtriangledown_{xx} \mathcal{L}_k\mathbf{d} \text{ s.t. } \mathbf{J}_k\mathbf{d} \geq -\mathbf{c}_k \tag{6}$$

- compute a step length α of moving along $\mathbf{d}$.
- update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{d}$, and recompute $\mathbf{J}_{k+1}$, $\mathbf{g}_{k+1}$, $\mathbf{c}_{k+1}$ and $\bigtriangledown_{xx}\mathcal{L}_{k+1}$ and continue to the $(k+1)$-th iteration.

Nonlinear Programming

Primal-dual Solution

- equality constrained sub QP (5) is equivalent to

$$\min_{x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m} \mathcal{L}(x, \lambda) \tag{7}$$

- working on an augmented unknown space $(\mathbf{x}, \lambda)$, solve for both $\mathbf{x}$ and λ simultaneously.
- upon the termination of a sub QP (6),
  - obtain $\mathbf{d}$ and estimate of Lagrange multipliers μ.
  - by strict complementary condition, $\mu_i = 0, \forall i \notin \mathcal{W}$.

Quasi-Newton Approximation

- replace the Hessian matrix $\bigtriangledown_{xx}\mathcal{L}_k$ in sub QP (6) by a Quasi-Newton approximation matrix $\mathbf{B}_k$.
- between SQP major iterations, $\mathbf{B}_k$ is updated by BFGS method on a modified Lagrange function,

$$\mathcal{L}_m(\mathbf{x}, \lambda) = \psi(\mathbf{x}) - \lambda^T \left(c(\mathbf{x}) - \hat{c}(\mathbf{x})\right) \tag{8}$$

- $\mathcal{L}_m(\mathbf{x}, \lambda)$ has same Hessian as $\mathcal{L}(\mathbf{x}, \lambda)$.

Nonlinear Programming

Quasi-Newton Approximation, BFGS Update

Define,

$$
\begin{aligned}
\delta &= \mathbf{x}_{k+1} - \mathbf{x}_k \\
\mathbf{y} &= \nabla \mathcal{L}_m(\mathbf{x}_{k+1}, \lambda_{k+1}) - \nabla \mathcal{L}_m(\mathbf{x}_k, \lambda_{k+1}) \\
&= \mathbf{g}_{k+1} - \mathbf{g}_k - (\mathbf{J}_{k+1} - \mathbf{J}_k)^T \lambda_{k+1}
\end{aligned}
$$

- theoretically, for the positive definiteness of $\mathbf{B}_{k+1}$, we need $\mathbf{y}^T\delta > 0$.
- if $\mathbf{y}^T\delta < \sigma$, where $\sigma = \alpha(1-\eta)\mathbf{d}^T\mathbf{B}_k\mathbf{d}$, for a constant $0 < \eta < 1$, two trial modifications are attempted, details in [4].
- if both trials fail to remedy the definitness of $\mathbf{B}_{k+1}$, the Hessian approximation is not updated.

Merit Function

- as soon as we move away from $\mathbf{x}_k$, $c(\mathbf{x}) - \hat{c}(\mathbf{x}) \neq 0$, feasibility could be broken.
- need a merit function to balance the reduction of $\psi(\mathbf{x})$ and the violation of $c(\mathbf{x})$.
- slack variables $\mathbf{s}$ and penalty factor ρ are introduced to incorporate the violation components in the merit function.

$$\mathcal{M}_\rho(\mathbf{x}, \lambda, \mathbf{s}) = \psi(\mathbf{x}) - \lambda^T(c(\mathbf{x}) - \mathbf{s}) + \frac{1}{2}\sum_{i=1}^{m} \rho_i \left(c_i(\mathbf{x}) - \mathbf{s}_i\right)^2 \tag{9}$$

Nonlinear Programming

Merit Function in Line Search

- upon the termination of a sub QP, define the augmented search direction as

$$\begin{pmatrix} \mathbf{d} \\ \xi \\ \mathbf{q} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mu - \lambda_k \\ \hat{\mathbf{s}}_k - \mathbf{s}_k \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mu - \lambda_k \\ \mathbf{J}_k \mathbf{d} + \mathbf{c}_k - \mathbf{s}_k \end{pmatrix} \tag{10}$$

- use a line search method with $\mathcal{M}_\rho(\mathbf{x}, \lambda, \mathbf{s})$ as the objective function to determine a step length α along this augmented direction.
- update the primal-dual variables to start the $(k+1)$-th major iteration,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha \mathbf{d} \\ \lambda_{k+1} &= \lambda_k + \alpha \xi \end{aligned}$$

Nonlinear Programming

Penalty ρ in Merit Function

- fix the search direction (10) and write the merit function (9) as a univariate function $\phi_\rho(\alpha)$ wrt step length α,

$$
\begin{aligned}
v(\alpha) &= \begin{pmatrix} \mathbf{x}_k \\ \lambda_k \\ \mathbf{s}_k \end{pmatrix} + \alpha \begin{pmatrix} \mathbf{d} \\ \xi \\ \mathbf{q} \end{pmatrix} \\
\phi_\rho(\alpha) &= \mathcal{M}_\rho(v(\alpha))
\end{aligned} \tag{11}
$$

- to make descent step move, we need $\phi'(0)$ be significantly negative. The choice in SNOPT method [4] is to find ρ such that

$$
\phi'_\rho(0) < -\frac{1}{2}\mathbf{d}^T \mathbf{B}_k \mathbf{d} \tag{12}
$$

Nonlinear Programming

Constraint Feasibility

- use phase-I active set method to find an initial feasible point $\mathbf{x}_0$ wrt only general linear constraints and simple bounds in formulation (1).
- if $\mathbf{x}_0$ is found, the active set method ensures linear constraints are satisfied in all subsequent iterations; otherwise, declare the problem has no feasible solution.
- at each major iteration to start a sub QP problem, it is possible that a feasible point does not exist wrt all linear constraints and the linearization $\hat{c}(\mathbf{x}_k)$ of nonlinear constraints $c(\mathbf{x})$.

Nonlinear Programming

Elastic Programming Mode

- if a sub QP cannot find a starting feasible point, we introduce surplus variables $\mathbf{w}$ and enter elastic programming (EP) mode to solve,

$$\min_{\mathbf{x}\in\mathbb{R}^n,\mathbf{w}\in\mathbb{R}^m} \psi^{(e)}(\mathbf{x},\mathbf{w}) \text{ s.t. } c^{(e)}(\mathbf{x},\mathbf{w}) \geq 0, \mathbf{w} \geq 0$$

, where

$$\psi^{(e)}(\mathbf{x},\mathbf{w}) = \psi(\mathbf{x}) + \gamma \mathbf{e}^T \mathbf{w}$$
$$c^{(e)}(\mathbf{x},\mathbf{w}) = c(\mathbf{x}) + \mathbf{w}$$

- if the new linearization $\hat{c}(\mathbf{x}_{k+1})$ has a starting feasible point, we quit from the EP mode and solve the original problem (2) again.
- if $\hat{c}(\mathbf{x}_{k+1})$ is still infeasible, we increase penalty factor γ until a max penalty value is reached.

References I

[1] Jorge Nocedal and Stephen J. Wright: Numerical Optimization, Springer, 1999

[2] Philip E. Gill, Walter Murray and Margaret H. Wright: Practical Optimization, Academic Press, 1981

[3] Philip. E. Gill and Elizabeth Wong: Sequential Quadratic Programming Methods, UCSD Department of Mathematics, Technical Report NA-10-03

[4] Philip. E. Gill, Walter Murray and Michael A. Saunders: SNOPT: An SQP Algorithm for Large-Scale Constrainted Optimization. SIAM Review, Volumn 47, Number 1, pp.99-131

References II

[5] Philip. E. Gill, Walter Murray and Michael A. Saunders: Some Theoretical Properties of an Augmented Lagrangian Merit Function, Advances in Optimization and Parallel Computing, P.M. Pardalos, ed., North-Holland, Amsterdam, 1992, pp. 101-128

[6] M. J. D. Powell, The Convergence of variable metric methods for nonlinearly constrainted optimization calculation, in Nonlinear Programming, 3 (Proc. Sympos., Special Interest Group Math. Programming, University of Wisconsin, WI, 1977), Academic Press, New York 1978, pp 27-63

[7] Samuel K. Eldersveld: Large-Scale Sequential Quadratic Programming Algorithms, Technical Report SOL 92-4, September 1992.