Trust-Region Sequential Quadratic Programming (SQP) Methods

Cory Cutsail

May 9, 2016

Trust Region Methods

- Trust region methods optimize over a region of an objective function. [4]
- Add a constraint to unconstrained objective
- General form of a trust region problem:

$$\min_p f(x_k + p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2 f(x_k + tp)p$$

$$\text{s.t.}$$

$$t \in (0, 1)$$

$$||p|| \leq \Delta$$

- Similar to QP

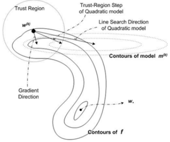

Figure: An example of the Trust-Region method, Nocedal and Wright [3], 67

Quadratic Programming

- Quadratic Programming methods optimize a quadratic objective function in $\mathbb{R}^n$.

$$\min_x P(x) = x^T Q x - c^T x$$

$$\text{s.t. } Ax \leq b$$

- Recall solving KKT System via Null-Space Method [3]

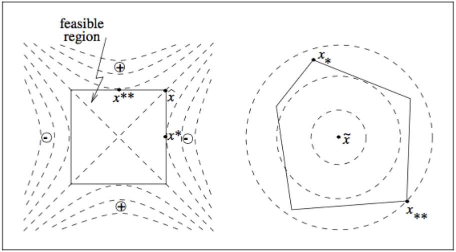

Figure: Nonconvex QPs, [3], 466

Sequential Quadratic Programming

- Use quadratic programming methods to optimize a nonlinear function.
- Each iterate approximates an objective function as a quadratic
- Methods shown here use linearized constraint approximations
- Don't need $\nabla^2_{xx}\mathcal{L}$ PSD; can be indefinite for TR methods
- Ensure global convergence

A Cookbook Approach

1. Convert NLP to QP at a point near min $\rightarrow$ Second-Order Taylor Approximation
2. Linearize constraints at same point $\rightarrow$ First-Order Taylor Approximation
3. Minimize model function using SQP method of choice (SLQP shown here)
4. New iterate is the point that minimized model over TR
5. Start over with new iterate

Restating the objective function

- Convert the original objective function, $f(x)$, to the quadratic approximation given by

$$f(x) \approx f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k) p$$

- This works locally, about $f(x_k)$.
- Error is $\mathcal{O}(||p||_2^2)$

Sequential Linear-Quadratic Programming (SLQP)

- The SLQP approach computes step in two "steps"
  1. Solve an LP to determine $\mathcal{W}$, $\mathcal{V}$.
  2. Solve an EQP, $\phi_1(m_k; \mu \cdot \mathcal{V})$ s.t. $\mathcal{W}$
- A Cauchy step also falls out of the LP scheme

The LP Step

1. Linearize everything (First-Order Taylor Approx, Maybe Log-Linear)
2. Constraints may not be L.I.; minimize penalty function and denote $\mathcal{V}$, $\mathcal{W}$.
   - $\min_p \phi_1(x_k;\mu) = f_k + \nabla f_k^T p +$
     $\mu \left(\sum_{i \in \mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| + \sum_{i \in \mathcal{I}} [c_i(x_k) + \nabla c_i(x_k)^T p]^- \right)$
     s.t.
     $||p||_\infty \leq \Delta_k^{LP}$
3. Yields p^{LP}, and the sets $\mathcal{V}$ and $\mathcal{W}$
4. Let $p^C = \alpha^{LP} p^{LP}$

The QP Step

- The QP is given by

$$\min_p f_k + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L} p + \left(\nabla f_k + \mu_k \sum_{i \in \mathcal{V}_k} \gamma_i \nabla c_i(x_k) \right)^T p$$

subject to $c_i(x_k) + \nabla c_i(x_k)^T p = 0, \ i \in \mathcal{W}_k$[1]
and $||p||_2 \leq \Delta_k$

- Footnote: Nocedal has $i \in \mathcal{E} \cap \mathcal{W}$
- Solution to this gives p^Q. Next iterate;
$p_k = p^C + \alpha^Q (p^Q - P^C)$
- Step is scaled, so that the worst possible step taken is the Cauchy step.

Algorithm

Algorithm 2 Step Computation and Penalty Update for SLQP and Sℓ_1QP Methods

1: Take as given x_k, μ_{k-1},$\Delta_k > 0$, and parameters $\varepsilon_1, \varepsilon_2 \in (0,1)$.

2: First solve

$$\min_p m_\mu(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p + \mu \left(\sum_{i \in \mathcal{E}} z_i + \sum_{i \in \mathcal{I}} (t_i)^- \right) \text{ s.t. } \begin{cases} \nabla c_i(x_k)^T p + c_i(x_k) = z_i, \ i \in \mathcal{E} \\ \nabla c_i(x_k)^T p + c_i(x_k) \geq -t_i, \ i \in \mathcal{I} \\ z, t \geq 0 \\ ||p||_\infty < \Delta \end{cases}$$

with $\mu = \mu_{k-1}$ for $p(\mu_{k-1})$.

3: Denote the penalized portion of the penalty function at iterate k by $\xi_k(\cdot)$.

4: **if** $\xi_k(p(\mu_{k-1})) = 0$ **then** Let $\mu_k = \mu_{k-1}$

5: **else** Compute the infinity norm of the step, p_∞

6: **if** $\xi_k(p_\infty) = 0$ **then** find $\mu_{k+1} \geq \mu_{k-1}$ s.t. $\xi_k(p(\mu_{k+1})) = 0$

7: **else** Find $\mu_{k+1} \geq \mu_{k-1}$ such that

$$\xi_k(0) - \xi_k(p(\mu_{k+1})) \geq \varepsilon_1 [\xi_k(0) - \xi_k(p_\infty)]$$

8: If μ_{k+1} is not such that

$$m_{\mu_{k+1}}(0) - m_{\mu_{k+1}}(p(\mu_{k+1}) \geq \varepsilon_2 \mu_{k+1} [\xi_k(0) - \xi_k(p(\mu_{k+1}),$$

increase μ_{k+1} until the above is satisfied

9: μ_{k+1} becomes the new penalty parameter, and $p_{k+1} = p(\mu_{k+1})$ becomes the new step.

Applications

1. PDEs (Gerdt, Wave Equation [2] and Dennis, Heinkenschloss, Vicente [1], Heat Equation)
2. Minimize time to stationary equilibria via policy parameter - think optimal taxation for stationary capital
3. Optimal control of PDEs, dynamical systems of ODEs, DAEs [2]

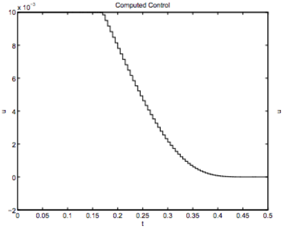

Figure: Numerical Heat Equation Results, [1], 43

J. Dennis, M. Heinkenschloss, and L. Vicente.
Trust-region interior point sqp algorithms for a class of nonlinear programming problems.
SIAM Journal for Control and Optimization, pages 1750–1794, 1998.

Matthias Gerdts.
Optimal Control of ODEs and DAEs.
Graduate Texts in Mathematics. De Gruyter, 2012.

Jorge Nocedal and Stephen J. Wright.
Numerical Optimization.
Springer Series in Operations Research. 2006.

Ya-Xiang Yuan.
A review of trust-region algorithms for optimization, 2000.