OPER 627: Nonlinear Optimization
Lecture 9: Trust-region methods

Department of Statistical Sciences and Operations Research
Virginia Commonwealth University

Sept 25, 2013
True or False:
- From any arbitrary initial point, BFGS algorithm guarantees convergence to a stationary point
- For a general nonlinear smooth function, BFGS algorithm has superlinear local convergence under some mild conditions
- Quasi-Newton methods are good at sparse large-scale optimization problems

What is the idea of LBFGS method?
Today’s Outline

- Introduction to trust region methods
- Trust region algorithm
- Trust region subproblems
An alternative perspective

Model function to approximate $f(x_k + p)$:

$$m_k(p) = f(x_k) + \nabla f(x_k)^\top p + \frac{1}{2} p^\top B_k p$$

This model is only “locally” reliable: $p$ cannot be too large

Let’s go local

1. Step 1: Choose a trust region (a compact convex neighborhood around the current point)
2. Step 2: Find a minimizer of some model function in the trust region

Q: Why do we need the trust region?
Trust region subproblem

1. Model function: $m_k(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p$
2. Trust region radius: $\Delta, \|p\| \leq \Delta$ (2-norm by default)
   - $B_k$ could be the true Hessian
   - $B_k$ could be a quasi-Newton approximate matrix

Trust region subproblem (TRP):

$$\min_p m(p)$$

s.t. $\|p\| \leq \Delta$

Make $p_k$ as large as possible (Why?), as long as $m(p)$ agrees “closely” with $f(x + p)$

How close?

- $A-Red_k$: actual reduction in $f$, $\Delta f_k := f(x_k) - f(x_k + p_k)$
- $P-Red_k$: predicted reduction in $f$, $\Delta m_k := f(x_k) - m_k(p_k)$
- Ratio: $\rho_k = \frac{\Delta f_k}{\Delta m_k}$, if close to 1, then we have a good agreement
Trust region: simple idea

If $\rho$ is far from 1:

If $\rho$ is close to 1, and $\rho$ hit the boundary: $\|\rho\| = \delta$: 
Trust region algorithm

Algorithm 1 Trust region ($\bar{\Delta} > 0, \Delta_0 \in (0, \bar{\Delta}), \eta \in [0, \frac{1}{4})$)

1: Given $x_k$, $\Delta_k > 0$, calculate $g_k = \nabla f(x_k)$ and $B_k$
2: (Key) Solve $TRP_k$ for $p_k$
3: Evaluate $f(x_k + p_k)$, and calculate $\Delta f_k$, $\rho_k$
4: if $\rho_k < \frac{1}{4}$ (not a good model, reduce the trust region radius) then
5: $\Delta_{k+1} = \frac{\Delta_k}{4}$
6: else if $\|p_k\| = \Delta_k$ (hit the boundary) then
7: $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
8: else
9: $\Delta_{k+1} = \Delta_k$
10: end if
11: if $\rho_k > \eta$ (so there are some reduction in the model) then
12: $x_{k+1} = x_k + p_k$
13: else
14: $x_{k+1} = x_k$ (null step)
15: end if
Example: \( \min_x (x^2 - 1)^2 \)

1. Line search at 0: cannot get over saddle point
2. Trust region:

\[
\min_d -2d^2 \quad \text{s.t.} \quad |d| \leq \Delta
\]

Escape the saddle point!
Solution of TRP

Trust region subproblem (TRP):

$$\begin{align*}
\min_{p} & \quad m(p) \\
\text{s.t.} & \quad \|p\| \leq \Delta
\end{align*}$$

Theorem (More and Sorensen 1983)

Vector $p^*$ is a global minimizer of TRP if and only if $\|p^*\| \leq \Delta$ and

$$\exists \lambda \geq 0 \text{ that:}$$

1. $(B + \lambda I)p^* = -g$
2. $\lambda(\Delta - \|p^*\|) = 0$
3. $B + \lambda I$ is PSD
Two cases

1. \( \lambda = 0 \), this happens only if \( B \) is PSD, and \( p^* = -B^{-1}g \) satisfies
\[ \|p^*\| \leq \Delta \]

2. \( \lambda > 0 \), then we look for a number \( \lambda \) that is big enough so that
\( B + \lambda I \) is PSD, and \( p^* = -(B + \lambda I)^{-1}g \) satisfies \( \|p^*\| = \Delta \)

Let \( p(\lambda) = -(B + \lambda I)^{-1}g \), we “just” need to solve \( \|p(\lambda)\| = \Delta \), which is
a single-variable root finding problem!
How to solve? No explicit form $p(\lambda)$!

Let’s do some derivation!
Algorithm for solving TRP

Algorithm 2 Trust region subproblem

1: Given $\lambda_0, \Delta > 0$
2: for $l = 0, 1, \cdots$ do
3: Factorize $B + \lambda_l I = R^\top R$ (safeguard here to ensure Cholesky factorization)
4: Solve $R^\top Rp_l = -g$
5: Solve $R^\top q_l = p_l$
6: Set $\lambda_{l+1} = \lambda_l + \frac{\|p_l\|^2 \|p_l\| - \Delta}{\|q_l\|^2 \Delta}$
7: end for

Good news: typically just 2 or 3 steps of Newton iteration!