Johns Hopkins Engineering

625.464 Computational Statistics

Multivariate Optimization Problems

Module 3 Lecture 3A

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Multivariate Optimization Problems

- Seek to max/min a real valued function g of a p-dim vector

$$x = (x_1, \dots, x_p)^T$$

- notate iteration t, $x^{(t)} = (x_1^{(t)}, \dots, x_p^{(t)})^T$
- steps based on linearization of $g'(x)$
- convergence criteria in the same spirit

Convergence Criteria

Need $D(u,v)$ a distance measure for p-dim vectors.

~~Ex~~ $D(u,v) = \sum_{i=1}^{p} |u_i - v_i|$ $\quad D(u,v) = \sqrt{\sum_{i=1}^{p} (u_i - v_i)^2}$

Absolute convergence

$$D(x^{(t+1)}, x^{(t)}) < \varepsilon$$

relative convergence

$$\frac{D(x^{(t+1)}, x^{(t)})}{D(x^{(t)}, 0)} < \varepsilon$$

Definition of Hessian and Gradient

Recall that for a multivariate function f, the gradient of f at x is

$$f'(x) = \left(\frac{df(x)}{dx_1}, \frac{df(x)}{dx_2}, \dots, \frac{df(x)}{dx_p} \right)^T$$

the Hessian is the matrix

$$f''(x) = \left[\frac{d^2 f(x)}{dx_i \, dx_j} \right]_{ij}$$

Newton's Method and Fisher Scoring

For Newton's Method: we approximate $g(x^*)$ by the Taylor Series

$$g(x^*) = g(x^{(t)}) + (x^* - x^{(t)})^T g'(x^{(t)}) + (x^* - x^{(t)})^T g''(x^{(t)}) (x^* - x^{(t)}) / 2$$

set grad. of RHS = 0.

$$0 = g'(x^{(t)}) + g''(x^{(t)}) (x^* - x^{(t)})$$

Newton's Method and Fisher Scoring

Newton's Method Algorithm:

$$x^{(t+1)} = x^{(t)} - g''(x^{(t)})^{-1} g'(x^{(t)})$$

Fisher Scoring Algorithm:

$$\theta^{(t+1)} = \theta^{(t)} + I(\theta^{(t)})^{-1} l'(\theta^{(t)})$$

Newton-like Methods

Computation of the Hessian

$$g''(x^{(t)}) = \left[\frac{d^2 f(x)}{dx_i dx_j} \right]_{ij}$$

Can be expensive.

$$x^{(t+1)} = x^{(t)} - \left(M^{(t)}\right)^{-1} g'(x^{(t)})$$

where $M^{(t)}$ is a $p \times p$ approx.

Ascent Algorithms

With Newton's method the steps are not necess. up hill.

ie $g(x^{(t+1)}) > g(x^{(t)})$

Method of Steepest Ascent: $M^{(t)} = -I$

$$x^{(t+1)} = x^{(t)} + g'(x^{(t)})$$

- Scaled steps

$$x^{(t+1)} = x^{(t)} + \alpha^{(t)} g'(x^{(t)})$$

for $\alpha^{(t)} > 0$

- General Ascent Alg

$$x^{(t+1)} = x^{(t)} - \alpha^{(t)} (M^{(t)})^{-1} g'(x^{(t)})$$

Backtracking in an Ascent Algorithm

Backtracking:

- Start each step w/ $\alpha^{(t)} = 1$
- if step is downhill $\left(g(x^{(t+1)}) < g(x^{(t)})\right)$

  let $\alpha^{(t)} = \frac{1}{2}\alpha^{(t)}$ & try again

- Repeat until step is uphill

Fixed-Point Method

If $M^{(t)} = M$ $\forall t$ we have a fixed pt. method.

$$x^{(t+1)} = x^{(t)} - M^{-1} g'(x^{(t)})$$

A reasonable choice is $M = g''(x^{(0)})$.

If M is a diagonal matrix, then this is eq to applying the univariate scaled fixed pt alg to each component.

Secant-Like Methods

We replace $g''(x^{(t)})$ with a matrix $M^{(t)}$ of finite discrete difference quotients.

~~Ex~~ Let $g_i'(x) = \frac{dg(x)}{dx_i}$ ith element of $g'(x)$ & $e_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow$ jth position

$$M_{ij}^{(t)} = \frac{g_i'(x^{(t)} + h_{ij}^{(t)} e_j) - g_i'(x^{(t)})}{h_{ij}^{(t)}}$$ for constants $h_{ij}^{(t)}$.

If $h_{ij}^{(t)} = h$ we get a convergence order 1.

If $h_{ij}^{(t)} = x_j^{(t)} - x_j^{(t-1)}$ $\forall i$ convergence order sim. to secant method in uni. case.