Newton's method

For suitably differentiable V:

$$V(x_j+\delta x) \cong V(x_j) + \nabla V(x_j)^T \delta x + \sum_{p=1}^{n}\sum_{q=1}^{n} \frac{\partial^2 V(x)}{\partial x_p \partial x_q}\delta x_p \delta x_q$$

$$= V(x_j) + \nabla V(x_j)^T \delta x + \tfrac{1}{2}\delta x^T V_{xx}(x_j)\delta x$$

where

(i) $V_{xx}(x) \in \mathbb{R}^{n\times n}$

(ii) $[V_{xx}(x)]_{p,q} \triangleq \dfrac{\partial^2 V(x)}{\partial x_p \partial x_q}$

(iii) $V_{xx}(x)$ is symmetric since

$$[V_{xx}(x)]_{p,q} = \frac{\partial^2 V(x)}{\partial x_p \partial x_q} = \frac{\partial^2 V(x)}{\partial x_q \partial x_p} = [V_{xx}(x)]_{q,p}$$

eg. $V: \mathbb{R}^2 \to \mathbb{R}$ then

$$V_{xx}(x) = \begin{bmatrix} \dfrac{\partial^2 V(x)}{\partial x_1 \partial x_1} & \dfrac{\partial^2 V(x)}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 V(x)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 V(x)}{\partial x_2 \partial x_2} \end{bmatrix}$$

So:

$$V(x_j+\delta x) \cong V(x_j) + \nabla V(x_j)^T \delta x + \tfrac{1}{2}\delta x^T V_{xx}(x_j)\delta x \quad (4.1)$$

for small $\|\delta x\|$

(4.2) For our standard quadratic on $\mathbb{R}^n$

$$V(x_j+\delta x) = V(x_j) + \nabla V(x_j)^T \delta x + \tfrac{1}{2}\delta x^T C \delta x$$

for all $\delta x \in \mathbb{R}^n$

$$\Rightarrow V_{xx}(x_j) = C > 0 \ , \ \forall\, x_j$$

How to get search directions from such quadratic approximations of V ?

Consider a slightly more general quadratic approximations

$$V(x) = V(x_j + \underbrace{[x-x_j]}_{\delta x})$$

$$\cong V(x_j) + \nabla V(x_j)^T \delta x + \tfrac{1}{2}\delta x^T P(x_j)\delta x \triangleq v_j^P(x)$$

with $P(x_j)^T = P(x_j)$

Standing for:
value at x of expansion about x_j with second order contribution $\frac{1}{2}\delta x^T P(x_j)\delta x$

(4.4) Th Consider

$$V_j^P(x) = V(x_j) + \nabla V(x_j)^T (x - x_j) + \tfrac{1}{2}(x-x_j)^T P(x_j)(x-x_j)$$

where $P(x_j)^T = P(x_j)$

Then: $\min_{x \in \mathbb{R}^n} V_j^P(x)$

(I) exists and occurs uniquely at:

$$\tilde{x}_j = x_j - P(x_j)^{-1} \nabla V(x_j) \quad \text{if } P(x_j) > 0$$

(II) exists and occurs non-uniquely at:

$$\tilde{x}_j = x_j - P(x_j)^{+} \nabla V(x_j)$$

the so called pseudo-inverse of $P(x_j)$; to be defined later

if $P(x_j) \geq 0$ and $\nabla V(x_j) \in \mathcal{R}[P(x_j)]$

i.e. $x^T P(x_j) x \geq 0$ for all $x \in \mathbb{R}^n$
i.e. $\lambda_i[P(x_j)] \geq 0, \forall i$

range of $P(x_j)$
$\{x \in \mathbb{R}^n : P(x_j) z = x, z \in \mathbb{R}^n\}$

(III) does not exist if any $\lambda_i[P(x_j)] < 0$

e.g. to see (I):

$$\nabla V_j^P(\tilde{x}_j) = \nabla V(x_j) + P(x_j)(\tilde{x}_j - x_j) = 0$$

necessary condition for a l.m. of V_j^P

$$\Rightarrow \tilde{x}_j - x_j = -P(x_j)^{-1} \nabla V(x_j)$$

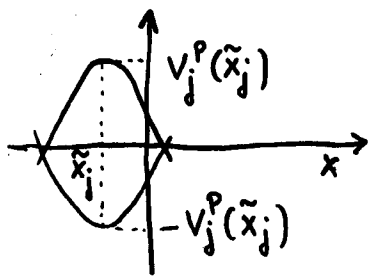
to see III:

suppose then that e.g. $P(x_j) < 0$

$\Rightarrow$ any $\tilde{x}_j$ s.t. $\nabla V_j^P(\tilde{x}_j) = 0$

is a maximizer (NOT a minimizer) of V_j^P since

$$\max_{x \in \mathbb{R}^n} V_j^P(x) = -\min_{x \in \mathbb{R}^n} [-V_j^P(x)]$$



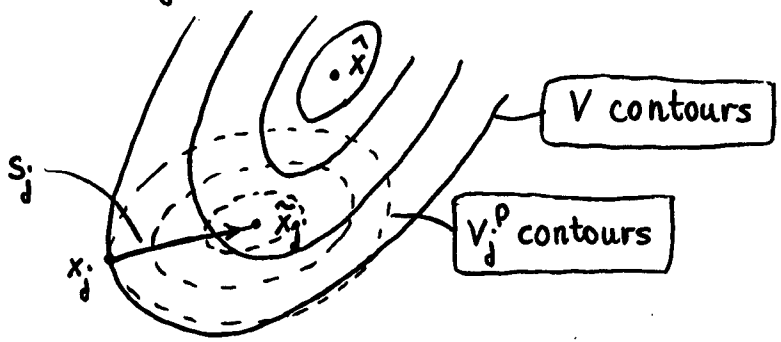
minimizer exists and is given by (I) since $-V_j^P$ involves $-P(x_j) > 0$

Hence minimization of V_j^P is simplest when $P(x_j) > 0$ and $P(x_j) = P(x_j)^T$. ← henceforth always assumed

For general V:

$V(x) \cong V_j^P(x)$ when $x \cong x_j$

and we might have



Suggests: use $s_j = \tilde{x}_j - x_j$

g.m. for V_j^P

Note: Better approximation V_j^P to V

$\Rightarrow$ $\tilde{x}_j$ nearer $\hat{x}$

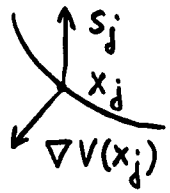
$\Rightarrow$ s_j better, because it points nearer to $\hat{x}$

(4.6) **Def^n** s_j is called a descent-direction if:

$V(x_j + \omega s_j) < V(x_j)$

for some $\omega > 0$

e.g. if $\nabla V(x_j)^T s_j < 0$ #



(4.7) **Th** If $P(x_j) > 0$ and $\nabla V(x_j) \neq 0$

and $s_j \triangleq \tilde{x}_j - x_j$

Then:

① $s_j = -P(x_j)^{-1} \nabla V(x_j)$ ← see Th. (4.4)(I)

② s_j = a descent direction

↑

to see this:

$\nabla V(x_j)^T s_j = -\nabla V(x_j)^T P(x_j)^{-1} \nabla V(x_j) < 0$

because $P(x_j) > 0 \Rightarrow P^{-1}(x_j) > 0$ for all x_j ($\not\in$)

---

as then $\nabla V(x_j)$ and s_j are at an obtuse angle to each other

Proof of (♯):

call $y \triangleq P^{-1}(x_j)\,x \Rightarrow x = P(x_j)\,y$, $\forall x$

$\Rightarrow x^T P^{-1}(x_j)\,x = y^T P(x_j) P^{-1}(x_j) P(x_j)\,y$

$= I$

$= y^T P(x_j)\,y > 0$, $\forall y \neq 0$

$\Downarrow$

$x^T P^{-1}(x_j)\,x > 0$, $\forall x \neq 0$

as $x = 0 \Leftrightarrow y = 0$

Hence we consider:

The P-algorithm = the s.d. algorithm with the above s_j, $\forall j$

different P's yield different algorithms

Some choices for $P(x_j)$

Requirements:

$P(x_j)^T = P(x_j) > 0$, $\forall x_j$

$V(x) \cong V_j^P(x)$ (*)

near as possible

$\Downarrow$

(I) If $V_{xx}(x_j) > 0$, $\forall j$:

choose $P(x_j) = V_{xx}(x_j)$ #

$\Rightarrow s_j = -V_{xx}(x_j)^{-1} \nabla V(x_j)$

↑

Newton s.d. & resulting P-alg. called Newton alg.

(II) If $V_{xx}(x_j) \not> 0$:

choose $P(x_j)$ = a positive definite approximation to $V_{xx}(x_j)$

$\Downarrow$

modified Newton algorithm

Choosing a positive definite approximation to $V_{xx}(x_j)$: $\Downarrow$

As then $P^T = P > 0$ and $V_j^P(x)$ is a good approximation to $V(x)$ when $\| x - x_j \|$ is small — so the requirements (*) on P are satisfied

Put $V_{xx}(x_j)$ in spectral form:

$$V_{xx}(x_j) = M \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} M^T$$

(M: $\perp$)

orthogonal matrix (indicated by $\perp$) meaning that $M^T M = M M^T = I$

(the columns of M are normalized eigenvectors of V)

Facts: ① $V_{xx} \not> 0$ iff any $\lambda_i \leq 0$

② $P > 0$ iff all $\lambda_i [P] > 0$

⇓

Obtain a pos. def. approximation P to V_{xx} by setting

$\bar{\lambda}_i = \lambda_i$ if $\lambda_i > 0$

$\bar{\lambda}_i = \varepsilon > 0$ if $\lambda_i \leq 0$

⇓

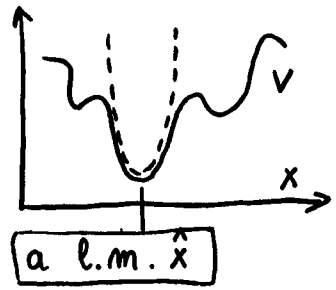
$$P(x_j) = M \begin{bmatrix} \bar{\lambda}_1 & & 0 \\ & \ddots & \\ 0 & & \bar{\lambda}_n \end{bmatrix} M^T$$

← pos. def. because all the $\bar{\lambda}_i > 0$

Remarks on Newton and modified-Newton algs.

① For our standard quadratic, Newton gives $x_1 = \hat{x}$, $\forall x_0$ ← very good

② Many nasty functions V look like a quadratic near a l.m.



Hence Newton often gives fast convergence to a l.m. $\hat{x}$ once it gets near $\hat{x}$

③ Say $n = 20$. Allowing for symmetry V_{xx} involves 210 different p.ds. $\frac{\partial^2 V}{\partial x_p \partial x_q}$

⇒ you have to find 210 different formulae
- too much like hard work

⇒ methods using V_{xx} not often practical for big problems ← Bad

⇓

Secant algorithms invented.

Secant Algorithms

↑ In these estimate $V_{xx}(x_j)^{-1}$ from changes in ∇V and x observed as the sd. algorithm iterates

↳ secant algorithms also known as quasi-Newton algorithms

Development of Secant Alg.

Consider the standard quadratic on $\mathbb{R}^n$ when:

- V & ∇V computable, somehow, $\forall x$
- C unknown (but $C^T = C > 0$)

corresponding to an unknown V_{xx} as $V_{xx}(x) = C$, $\forall x$, for quadratic V

Estimation of $V_{xx}(x)^{-1}$ = estimation of C^{-1}

For quadratic V:

$$\nabla V(x+\Delta x) = \nabla V(x) + C\,\Delta x \quad , \quad \forall \Delta x$$

$$\Rightarrow \quad C\Delta x = \nabla V(x+\Delta x) - \nabla V(x) \triangleq \Delta g$$

$$\Downarrow$$

$$\Delta x = C^{-1}\Delta g \qquad (4.11)$$

given observed pairs $(\Delta x, \Delta g)$ shall use to estimate C^{-1}

note that $\Delta x = C^{-1}\Delta g$ for one pair $(\Delta x, \Delta g)$ does not specify every entry of C^{-1} — n linearly independent pairs $(\Delta x_i, \Delta g_i)$ are actually needed for that

⇓ see next

At iteration j of the s.d. alg., WE KNOW:

past changes Δx_i , caused changes Δg_i

$i = 0, \dots, j-1$: $\Delta x_i = x_{i+1} - x_i$, $\Delta g_i = \nabla V(x_{i+1}) - \nabla V(x_i)$

$\Downarrow$

We know C^{-1} satisfies:

$$\Delta x_i = C^{-1} \Delta g_i \quad , i = 0, 1, \dots, j-1 \qquad (\#)$$

with $(C^{-1})^T = C^{-1} > 0$

Suppose we choose $H_j \in \mathbb{R}^{n \times n}$, so that:

$$\Delta x_i = H_j \Delta g_i \quad , \ \forall\, i = 0, 1, \dots, j-1$$

with $H_j^T = H_j > 0$

Secant condition

Then $H_j \cong$ the unknown C^{-1}

in some sense because H_j satisfies properties # of C^{-1}

Since $H_j \cong C^{-1}$

Hence use: $s_j = -H_j \nabla V(x_j) \cong -C^{-1} \nabla V(x_j)$

s_j: pseudo-Newton search direction

$-C^{-1} \nabla V(x_j)$: Newton search dir. ($\$$)
- would yield $x_{j+1} = \hat{x}$ (#)
- but unusable as C unknown

After optimizing along s_j (approximately?) we will have the extra information that

$$\Delta x_j = x_{j+1} - x_j = w_j s_j$$

caused by the gradient change

$$\Delta g_j = \nabla V(x_{j+1}) - \nabla V(x_j)$$

and we can build this new information into a potentially better estimate H_{j+1} of C^{-1} than H_j

---

($\$$) because $C = V_{xx}$ for our quadratic

(#) for our quadratic V

(4.12) Secant Algorithm

- Choose (i) $x_0 \in \mathbb{R}^n$

  (ii) symmetric, pos. def. $H_0 \in \mathbb{R}^{n \times n}$

  estimate of unknown C^{-1}, use $H_0 = I$ if no better guess available

- Set $j := 0$

1) [At iteration j]:

- Set $s_j := -H_j \nabla V(x_j)$ — pseudo-Newton s.dir
- Choose $\omega_j \geq 0$ so

  $$V(x_j + \omega_j s_j) < V(x_j)$$

  ideally ω_j chosen to minimize this
- Set $x_{j+1} := x_j + \omega_j s_j$
- Stop if $\|\nabla V(x_{j+1})\| < \varepsilon$ = small
- $\Delta x_j := x_{j+1} - x_j$, $\Delta g_j := \nabla V(x_{j+1}) - \nabla V(x_j)$
- Choose symm. pos.def. $H_{j+1} \in \mathbb{R}^{n \times n}$ so $H_{j+1} \cong C^{-1}$ – improved
- Set $j := j+1$, Go to 1)

There exist many ways for choosing suitable H_j. Most famous is:

The Davidon-Fletcher-Powell Alg. (DFP)

= the Secant Alg. when:

exact opt. used along each s_j

H_i chosen using:

$$H_{j+1} = H_j + \frac{\Delta x_j (\Delta x_j)^T}{(\Delta x_j)^T (\Delta g_j)} - \frac{H_j \Delta g_j (H_j \Delta g_j)^T}{(\Delta g_j)^T H_j \Delta g_j} \qquad (4.13)$$

If approximate minimization used along each s_j, we have to use other update schemes for finding H_{j+1} from $H_j, \Delta x_j, \Delta g_j$

e.g. – Symmetric Rank
– BFGS — probably the best
– Broyden

Ref: R. Fletcher: "Practical Methods of Optimization",

Properties of Secant Algs. for quadratics

(4.14) Th Consider

- Secant Alg. using exact minimization along each s_j
- standard quadratic V

Then:

① the s_j generated are C-conjugate

$\Rightarrow$ Secant Alg. = a C.D. Alg. – NICE
$\hat{x}$ achieved after at most n iterations

② if all n iterations needed to find $\hat{x}$, then $H_n = C^{-1}$

the estimate H_j of C^{-1} is eventually exact

(4.15) Th. For

- many Secant Algs. (including DFP) using exact minimization along each s_j and $H_0 = I$
- standard quadratic V:

If x_0 (Secant) $= x_0$ (CG)

then $x_j = x_j$ for all $j \geq 1$

Question:

Usually $H_0 = I$ so Secant & C.G. yield, for quadratics, the same x_j's

BUT:

Secant = more complex alg. than CG.

So: what is the use of Secant algs?

Answer:

Secant tends to be better than CG on non-quadratic V, when algs. applied properly to such V