Matrix Secant Methods

Equation Solving
$g(x) = 0$

Matrix Secant Methods

Newton-Like Iterations:

$$x^{k+1} := x^k - J_k g(x^k), \quad \text{where } J_k \approx g'(x^k).$$

Matrix Secant Methods

Newton-Like Iterations:

$$x^{k+1} := x^k - J_k g(x^k), \quad \text{where } J_k \approx g'(x^k).$$

3700 years ago the Babylonians used the *secant method* in 1D:

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}.$$

Matrix Secant Methods

Newton-Like Iterations:

$$x^{k+1} := x^k - J_k g(x^k), \quad \text{where } J_k \approx g'(x^k).$$

3700 years ago the Babylonians used the *secant method* in 1D:

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}.$$

This gives the error

$$g'(x^k)^{-1} - J_k = \frac{g(x^{k-1}) - [g(x^k) + g'(x^k)(x^{k-1} - x^k)]}{g'(x^k)[g(x^{k-1}) - g(x^k)]}.$$

Matrix Secant Methods

Newton-Like Iterations:

$$x^{k+1} := x^k - J_k g(x^k), \quad \text{where } J_k \approx g'(x^k).$$

3700 years ago the Babylonians used the *secant method* in 1D:

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}.$$

This gives the error

$$g'(x^k)^{-1} - J_k = \frac{g(x^{k-1}) - [g(x^k) + g'(x^k)(x^{k-1} - x^k)]}{g'(x^k)[g(x^{k-1}) - g(x^k)]}.$$

If $g'(x^*) \neq 0$, then $\exists\ \alpha > 0$ such that

$$\|g'(x^k)^{-1} - J_k\| \leq \frac{(L/2)\|x^{k-1} - x^k\|^2}{\alpha\|g'(x^k)\|\|x^{k-1} - x^k\|} \leq K\|x^{k-1} - x^k\|$$

by the Quadratic Bound Lemma, so $x^k \to \bar{x}$ 2-step quadratic rate.

Matrix Secant Methods

Can we apply the secant method to higher dimentions than 1?

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}$$

Matrix Secant Methods

Can we apply the secant method to higher dimentions than 1?

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}$$

Multiply on the rhs by $g(x^k) - g(x^{k-1})$ gives

$$J_k(g(x^k) - g(x^{k-1})) = x^k - x^{k-1}.$$

This is called the *matrix secant equation* (MSE), or quasi-Newton equation.

Matrix Secant Methods

Can we apply the secant method to higher dimentions than 1?

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}$$

Multiply on the rhs by $g(x^k) - g(x^{k-1})$ gives

$$J_k(g(x^k) - g(x^{k-1})) = x^k - x^{k-1}.$$

This is called the *matrix secant equation* (MSE), or quasi-Newton equation.
Here the matrix J_k is unknown (n^2 unknowns), and satisfies the n linear equations of the MSE.

Matrix Secant Methods

Can we apply the secant method to higher dimentions than 1?

$$J_k = \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})}$$

Multiply on the rhs by $g(x^k) - g(x^{k-1})$ gives

$$J_k(g(x^k) - g(x^{k-1})) = x^k - x^{k-1}.$$

This is called the *matrix secant equation* (MSE), or quasi-Newton equation.

Here the matrix J_k is unknown (n^2 unknowns), and satisfies the n linear equations of the MSE.

There are not enough equations to nail down all of the entries if J_k, so further conditions are required. What should they be?

Matrix Secant Methods

To better understand what further conditions on J_k are sensible, we revert to discussing the matrices $B_k = J_k^{-1}$, so the MSE becomes

$$B_k(x^k - x^{k-1}) = g(x^k) - g(x^{k-1}).$$

Matrix Secant Methods

To better understand what further conditions on J_k are sensible, we revert to discussing the matrices $B_k = J_k^{-1}$, so the MSE becomes

$$B_k(x^k - x^{k-1}) = g(x^k) - g(x^{k-1}).$$

Key Idea: Think of the J_k's as part of an iteration scheme

$$\{x^k, B_k\} \ .$$

Matrix Secant Methods

To better understand what further conditions on J_k are sensible, we revert to discussing the matrices $B_k = J_k^{-1}$, so the MSE becomes

$$B_k(x^k - x^{k-1}) = g(x^k) - g(x^{k-1}).$$

Key Idea: Think of the J_k's as part of an iteration scheme

$$\{x^k, B_k\} \; .$$

As the iteration proceeds, it is hoped that B_k becomes a better approximation to $g'(x^k)$. With this in mind, we assume that, in the long run, B_k is already close to $g'(x^k)$, so B_{k+1} should not differ from B_k by too much, i.e. B_k should be "close" to B_{k+1}.

Matrix Secant Methods

What do we mean by “close” in this context?
Should $\|B_k - B_{k+1}\|$ be small?

Matrix Secant Methods

What do we mean by “close” in this context?
Should $\|B_k - B_{k+1}\|$ be small?

We take “close” to mean algebraically close in the sense that B_{k+1} should be easy to compute from B_k and yet satisfy the MSE.

Matrix Secant Methods

What do we mean by "close" in this context?
Should $\|B_k - B_{k+1}\|$ be small?

We take "close" to mean algebraically close in the sense that B_{k+1} should be easy to compute from B_k and yet satisfy the MSE.

Let's start by assuming that B_{k+1} is a rank one update to B_k. That is, there exist $u, v \in \mathbb{R}^n$ such that

$$B_{k+1} = B_k + uv^T$$

Matrix Secant Methods

Set

$$s^k := x^{k+1} - x^k \qquad \text{and} \qquad y^k := g(x^{k+1}) - g(x^k)\ .$$

The MSE becomes $B_{k+1}s^k = y^k$.

Matrix Secant Methods

Set

$$s^k := x^{k+1} - x^k \qquad \text{and} \qquad y^k := g(x^{k+1}) - g(x^k)\ .$$

The MSE becomes $B_{k+1}s^k = y^k$.
Multiplying by s^k gives

$$y^k = B_{k+1}s^k = B_k s^k + uv^T s^k\ .$$

Hence, if $v^T s^k \neq 0$, we obtain

$$u = \frac{y^k - B_k s^k}{v^T s^k}$$

and

$$B_{k+1} = B_k + \frac{\left(y^k - B_k s^k\right) v^T}{v^T s^k}.$$

Matrix Secant Methods

$$B_{k+1} = B_k + \frac{\left(y^k - B_k s^k\right) v^T}{v^T s^k}$$

This equation determines a whole class of rank one updates that satisfy the MSE by choosing $v \in \mathbb{R}^n$ so that $v^T s^k \neq 0$.

Matrix Secant Methods

$$B_{k+1} = B_k + \frac{\left(y^k - B_k s^k\right) v^T}{v^T s^k}$$

This equation determines a whole class of rank one updates that satisfy the MSE by choosing $v \in \mathbb{R}^n$ so that $v^T s^k \neq 0$.

One choice is $v = s_k$ giving

$$B_{k+1} = B_k = \frac{\left(y^k - B_k s^k\right) {s^k}^T}{{s^k}^T s^k}.$$

This is known as Broyden's update. It turns out that the Broyden update is also analytically close to B_k.

Matrix Secant Methods

Let $A \in \mathbb{R}^{n\times n}$, s, $y \in \mathbb{R}^n$, $s \neq 0$. Then for any matrix norms $\|\cdot\|$ and $\|\cdot\|_e$ such that

$$\|AB\| \leq \|A\|\,\|B\|_e$$

and

$$\|\frac{vv^T}{v^Tv}\|_e \leq 1,$$

the solution to

$$\min_{B\in\mathbb{R}^{n\times n}} \{\|B-A\| : Bs = y\}$$

is

$$A_+ = A + \frac{(y-As)s^T}{s^Ts}.$$

In particular, A_+ solves the given optimization problem when $\|\cdot\|$ is the ℓ_2 matrix norm, and A_+ is the unique solution when $\|\cdot\|$ is the Frobenius norm.

Matrix Secant Methods

PROOF: Let $B \in \{B \in \mathbb{R}^{n\times n} : Bs = y\}$, then

$$\begin{aligned} \|A_+ - A\| &= \|\frac{(y - As)s^T}{s^Ts}\| = \|(B - A)\frac{ss^T}{s^Ts}\| \\ &\le \|B - A\|\,\|\frac{ss^T}{s^Ts}\|_e \le \|B - A\|. \end{aligned}$$

Broyden's Methods

Algorithm: Broyden's Method

Initialization: $x^0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n \times n}$

Having (x^k, B_k) compute (x^{k+1}, B_{x+1}) as follows:

Solve $B_k s^k = -g(x^k)$ for s^k and set

$$\begin{aligned} x^{k+1} : &= x^k + s^k \\ y^k : &= g(x^{k+1}) - g(x^k) \\ B_{k+1} : &= B_k + \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k}. \end{aligned}$$

Sherman-Morrison-Woodbury Formula for Matrix Inversion

Suppose $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times k}$, $V \in \mathbb{R}^{n\times k}$ are such that both A^{-1} and $(I + V^T A^{-1} U)^{-1}$ exist, then

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}$$

Inverse Broyden Update

If $B_k^{-1} = J_k$ exists and ${s^k}^T J_k y^k = {s^k}^T B_k^{-1} y^k \neq 0$, then

$$\begin{aligned} J_{k+1} &= \left[B_k + \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k} \right]^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k){s^k}^T B_k^{-1}}{{s^k}^T B_k^{-1} y} \\ &= J_k + \frac{(s^k - J_k y^k){s^k}^T J_k}{{s^k}^T J_k y}. \end{aligned}$$

Inverse Broyden Update

If $B_k^{-1} = J_k$ exists and ${s^k}^T J_k y^k = {s^k}^T B_k^{-1} y^k \neq 0$, then

$$\begin{aligned} J_{k+1} &= \left[B_k + \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k} \right]^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k){s^k}^T B_k^{-1}}{{s^k}^T B_k^{-1} y} \\ &= J_k + \frac{(s^k - J_k y^k){s^k}^T J_k}{{s^k}^T J_k y}. \end{aligned}$$

Under suitable hypotheses $x^k \to \bar{x}$ superlinearly.

Broyden Updates

Algorithm: Broyden's Method (Inverse Updating)

Initialization: $x^0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n\times n}$

Having (x^k, B_k) compute (x^{k+1}, B_{x+1}) as follows:

$$\begin{aligned} s^k : &= -J_k g(x^k) \\ x^{k+1} : &= x^k + s^k \\ y^k : &= g(x^{k+1}) - g(x^k) \\ J_{k+1} &= J_k + \frac{(s^k - J_k y^k){s^k}^T J_k}{{s^k}^T J_k y}. \end{aligned}$$