Variational methods - 4D-Var $\rightsquigarrow$ Linear Models

$$x_1 = Mx_0 \quad , \quad x_0 \sim N(\mu_0, P_0)$$

$$y_1 = Hx_1 + \eta \quad , \quad \eta \sim N(0, R)$$

<u>So far</u> EnKF / KF: $p(x_1 | y_1) \propto p(x_1)\, p(y_1 | x_1)$

Note: $x_1 = Mx_0 \rightarrow$ is "just a fctn of x_0".

$$p(x_0 | y_1) \propto p(x_0)\, \underbrace{p(y_1 | x_0)}_{N(HMx_0,\, R)}$$

→ Find initial condition to match data!

→ $p(x_0 | y_1) \propto \exp(-\mathcal{F}(x_0))$

$$\mathcal{F}(x_0) = \frac{1}{2}(x_0 - \mu_0) P_0^{-1} (x_0 - \mu_0) + \frac{1}{2}(y - HMx_0) R^{-1} (y - HMx_0)$$

Variational method: optimize / minimize $\mathcal{F}(x_0)$

$$\nabla \mathcal{F} = P_0^{-1}(x_0 - \mu_0) + M^T H^T R^{-1} (HMx_0 - y) = 0$$

$$(P_0 + M^T H^T R^{-1} M^T H)\, x_0 = P_0^{-1} \mu_0 + M^T H^T R^{-1} y$$

$$\left\| \begin{array}{l} x_0^* = \mu_0 + K(y - HM\mu_0) \\ K = P_0 M^T H^T (HMP_0 M^T H^T + R)^{-1} \end{array} \right.$$

<u>just like KF.</u>

$$\nabla^2 J = (P_0 + M^T H^T R^{-1} M H)$$

→ Posterior covariance $(P_0 + M^T H^T R^{-1} M H)^{-1} = (I - KHM) P_0$

→ Also just like before.

→ Variational method: optimize $\log(p(x_0 | y_1))$

→ base estimates on $p(x_0|y_1) \propto p(x_0) p(y_1|x_0)$

→ EnKF : compute mean and cov. of $p(x_k | y_{1:k})$

Questions:

(1) Do EnKF and Var. give the same state estimate of x_1?

→ Yes, check with calculation.

(2) How to cycle a var method?

(i) $p(x_0|y_1) \propto p(x_0) \, p(y_1|x_0)$

$$\Downarrow$$

$$x_0^a = \mu_0 + K(y - HM\mu_0)$$

$$P_0^a = (I - KHM) P_0$$

(ii) propagate to time (1)

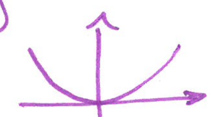
$$\underbrace{\begin{aligned} x_1^a &= M x_0^a \\ P_1^a &= M P_0^a M^T \end{aligned}}_{\text{(this is what KF would give you)}}$$

use this → as prior for next assimilation.

Rapid Review of optimization

What is minimizer and minimum of

$$f(x) = x^2.$$



$$f'(x) = 2x = 0 \quad \rightarrow \quad x^* = 0 \quad \text{(minimizer)}$$

$$f(x^*) = (x^*)^2 = 0 \quad \text{(minimum)}$$

What if f is not quadratic?

Taylor expand f at x_u.

$$f(x) = \underbrace{f(x_u) + f'(x_u)(x - x_u) + \tfrac{1}{2} f''(x_u)(x - x_u)^2}_{f_0} + \dots$$

↳ optimize f_0

$$f_0(x) = f(x_u) + f'(x_u)(x - x_u) + \tfrac{1}{2} f''(x_u)(x - x_u)^2$$

$$f_0'(x) = f'(x_u) + f''(x_u)(x - x_u) = 0$$

$$f''(x_u)(x - x_u) = -f'(x_u)$$

$$x = x_u - \left(f''(x_u)\right)^{-1}\left(f'(x_u)\right)$$

↳ set $x_u = 0$ and repeat.

↳ at each iteration you optimize a quadratic fcn

This is Newton's method.

This is very fast if fcn is nearly quadratic

This can converge well, but can also blow up.

→ Newton requires first and 2nd derivatives.

Multivariate version:

$$F_0(x) = F(x_k) + \nabla F|_k (x-x_k) + \frac{1}{2}(x-x_k) H_k (x-x_k)$$

H_k is Hessian of F at x_k: $[H_k]_{ij} = \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_{x=x_k}$

$$x^* = x_k - H_k^{-1} \nabla F_k$$ / better: $\underline{H_k(x-x_k) = -\nabla F_k}$

(no inverse).

Suppose you have more structure in F.

$F = \frac{1}{2} r(x)^2$ (multivariate: $F = \frac{1}{2} \sum r_i^2(x)$)
(see later)

$$F'(x) = r(x) r'$$
$$F''(x) = (r')^2 + r r''$$

Newton's method:

$$x_{k+1} = x_k + \left((r')^2 + r r''\right)^{-1} (r r')$$

Gauss-Newton:

$$x_{k+1} = x_k + (r')^2 (r r')$$

- "treat F at each step as if r was linear!)
- GN is "easier" than N because I only need first derivatives.
- GN only works for some "special" F.

Multivariate version:

$$F(x) = \frac{1}{2} \sum_{i=1}^{n} r_i(x)^2 = \frac{1}{2} r(x)^T r(x)$$

$$\nabla F(x) = J^T r \qquad [J]_{ij} = \frac{\partial r_i}{\partial x_j}, \text{ Jacobian}$$

approximate Hessian: $H \approx J^T J$

$$\hookrightarrow \quad x_{k+1} = x_k + (J^T J)^{-1} J^T r$$

$$\underbrace{\left((J^T J)(x_{k+1} - x_k) = J^T r \right)}$$

at each step you solve a LS problem!

Gauss-Newton and Kalman gains.

$$F(x) = \frac{1}{2}(x-\mu)^T B^{-1}(x-\mu) + \frac{1}{2}(HMx-y)^T R^{-1}(HMx-y)$$

$$= \frac{1}{2} r^T r \quad \leftarrow \text{prepare for GN}$$

$$r = \begin{pmatrix} B^{-1/2}(x-\mu) \\ R^{-1/2}(HMx-y) \end{pmatrix} \qquad J = \begin{pmatrix} B^{-1/2} \\ R^{-1/2}HM \end{pmatrix}$$

$$J^T J = \left[B^{-1/2} \quad M^T H^T R^{-1/2} \right] \begin{pmatrix} B^{-1/2} \\ R^{-1/2}HM \end{pmatrix}$$

$$J^T J = B^{-1} + M^T H^T R^{-1} HM \qquad \text{looks familiar}$$

$$(J^T J)^{-1} = (I - KHM)B$$

$$K = BM^T H^T (HMBM^T H^T + R)^{-1}$$

(see before)

$$J^T r = \left[B^{-1/2} \; M^T H^T R^{-1/2} \right] \begin{pmatrix} B^{-1/2}(x-\mu) \\ R^{-1/2}(HMx-y) \end{pmatrix}$$

$$J^T r = B^{-1}(x-\mu) + M^T H^T R^{-1}(HMx-y)$$

$$(J^T J)^{-1} J^T r$$

$$= (I - KHM)B\left(B^{-1}(x-\mu) + M^T H^T R^{-1}(HMx - y) \right)$$

At $x = 0$

$$(J^T J)^{-1} J^T r = (I - KHM)B(-B^{-1}\mu - M^T H^T R^{-1} y)$$

$$= -\left[\mu + K(y - HM\mu)\right]$$

see sofw

first iteration of GN, starting with $x = 0$

$$x_{k+1} = x_k - (J^T J)^{-1}(J^T r)$$

$$x = \mu + K(y - HM\mu)$$

And the iteration ends here.

→ Kalman filter formulas show up during optimization of $-\log p(x_0 | y_1) - \log p(x_1 | y_1)$ using GN!