Reconstructing Resolution from Covariance, with Application to Ensembles of Solutions

Bill Menke, October 21, 2018

This derivation follows up on the ideas of Menke (2018, section 5.3). We start by examining a linear least squares problem with data $\mathbf{d} = \mathbf{Gm}$ (with covariance $\mathbf{C}_d$) and prior information $\mathbf{h} = \mathbf{Hm}$ (with covariance $\mathbf{C}_h$). Generalized Least Squares gives the estimated model parameters $\mathbf{m}^{\text{est}}$ and posterior covariance $\mathbf{C}_m$ as:

$$\mathbf{m}^{\text{est}} = \mathbf{A}^{-1}[\mathbf{G}^{\text{T}}\mathbf{C}_{\text{d}}^{-1}\mathbf{d}^{obs} + \mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{h}]$$

$$\mathbf{C}_m = \mathbf{A}^{-1} \quad \text{with} \quad \mathbf{A} \equiv [\mathbf{G}^{\text{T}}\mathbf{C}_{\text{d}}^{-1}\mathbf{G} + \mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}] \tag{1}$$

As Menke (2018) points out, the problem is perfectly resolved as long as $\mathbf{A}^{-1}$exists; that is $\mathbf{R} = \mathbf{I}$. Nevertheless, a non-trivial resolution matrix can be constructed for *deviations* $\Delta\mathbf{m} = \mathbf{m} - \mathbf{m}^{\text{H}}$ of the model parameters away from the prior solution $\mathbf{m}^{\text{H}}$. By prior solution, we mean the solution implied by the prior information, acting alone; that is: $\mathbf{m}^{\text{H}} = [\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}]^{-1}\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{h}$, which has covariance $\mathbf{C}_m^{\text{H}} = [\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}]^{-1}$. (Menke (2018) recommends adding very weak smallness prior information to the problem in cases where the prior information is not complete, so that $[\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}]^{-1}$ always exists). The resolution matrix $\mathbf{R}^{\text{G}}$ for $\Delta\mathbf{m}$ is:

$$\mathbf{R}^{\text{G}} \equiv \mathbf{A}^{-1}\mathbf{G}^{\text{T}}\mathbf{C}_{\text{d}}^{-1}\mathbf{G} = \mathbf{A}^{-1}\left(\mathbf{A} - \mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}\right) = \mathbf{I} - \mathbf{A}^{-1}\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}$$

$$\mathbf{R}^{\text{G}} = \mathbf{I} - \mathbf{C}_m\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H} = \mathbf{I} - \mathbf{C}_m[\mathbf{C}_{\boldsymbol{m}}^{\text{H}}]^{-1} \tag{2}$$

Note that the resolution is exactly zero when $\mathbf{C}_m = \mathbf{C}_m^{\text{H}}$. This is the case where the data contributes no information, so that the posterior covariance of the model parameters is just their prior covariance.

The resolution $\mathbf{R}^{\text{G}} = \mathbf{I} - \mathbf{C}_m\mathbf{H}^{\text{T}}\mathbf{C}_{\text{h}}^{-1}\mathbf{H}$ can be constructed from the matrices $\mathbf{C}_m$, $\mathbf{H}$ and $\mathbf{C}_h$. I propose that this formula be applied to any problem for which: (1) an estimate of the posterior covariance matrix $\mathbf{C}_m$ is available, and (2) the prior information is linear (with known $\mathbf{H}$ and $\mathbf{C}_h$). The second criterion is not especially restrictive, since many problems can be adequately regularized with linear information, such a small solution size, small first or second derivative, etc. The approximation is accurate as long as the underlying problem has an error surface that is approximately quadratic near its minimum.

Now suppose one has a large number, say L, of solutions $\mathbf{m}^{(i)}$ that sample a posterior probability distribution $p\left(\mathbf{m}|\mathbf{d}^{\text{obs}}\right)$ (constructed, say, using the Metropolis-Hastings algorithm). The mean $\bar{\mathbf{m}}$ and covariance $\mathbf{C}_m$ can be estimated as the sample mean and covariance:

$$\bar{\mathbf{m}}^{est} = \frac{1}{L}\sum_i \mathbf{m}^{(i)} \quad \text{and} \quad [\mathbf{C}_m^{est}]_{jk} = \frac{1}{L}\sum_i \left(m_j^{(i)} - \bar{m}_j\right)\left(m_k^{(i)} - \bar{m}_k\right) \tag{3}$$

And the resolution as:

$$\mathbf{R}^{\mathrm{G}} \approx \mathbf{I} - \mathbf{C}_m^{est}\mathbf{H}^{\mathrm{T}}\mathbf{C}_{\mathrm{h}}^{-1}\mathbf{H} \tag{4}$$

In cases where $\mathbf{H}$ and $\mathbf{C}_h$ have not been explicitly stated, the Metropolis-Hastings algorithm may be used to sample the prior distribution, acting alone, and Equation 3 used to estimate $\mathbf{C}_m^{\mathrm{H}}$. The resolution is then given by:

$$\mathbf{R}^{\mathrm{G}} \approx \mathbf{I} - \mathbf{C}_m^{est}[\mathbf{C}_m^{\mathrm{H}est}]^{-1} \tag{4}$$

This procedure may yield good results even when the prior constraints are nonlinear, as long as the error surface for the prior information is approximately quadratic near its minimum.

We now examine a linear test scenario in which $M = 11$ model parameters are evenly spaced in an auxiliary parameter x. The data kernel $\mathbf{G}$ has $N = 11$ rows, each of which exponentially decline with column number:

$$d_i = \sum_j G_{ij}^0 m_j \quad \text{with} \quad G_{ij}^0 \propto \exp\{-c_i x_j\}$$

Here c_i are decay rates that increase with row number i. The true model parameters are $m_i = 1$ and the observed data $\mathbf{d}^{obs}$ are computed from the true data by adding Normally-distributed, uncorrelated noise with zero mean and variance σ_d^2. The prior information is taken to be smallness of the first model parameter, and smallness of differences between adjacent model parameters:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and has variance $\sigma_h^2 \gg \sigma_d^2$. The problem is solved both using Generalized Least Squares and by applying the Metropolis-Hastings algorithm to the distribution:

$$p(\mathbf{m}|\mathbf{d}^{obs}) \propto \exp\{-½E^2 - ½L^2\} \quad \text{with}$$

$$E^2 = (\mathbf{d}^{obs} - \mathbf{d}^{pre})^{\mathrm{T}}\mathbf{C}_d^{-1}(\mathbf{d}^{obs} - \mathbf{d}^{pre}) \quad \text{and} \quad L^2 = (\mathbf{h} - \mathbf{h}^{pre})^{\mathrm{T}}\mathbf{C}_h^{-1}(\mathbf{h} - \mathbf{h}^{pre})$$

$$\mathbf{d}^{pre} = \mathbf{Gm} \quad \text{and} \quad \mathbf{h}^{pre} = \mathbf{Hm} \quad \text{and} \quad \mathbf{C}_d = \sigma_d^2\mathbf{I} \quad \text{and} \quad \mathbf{C}_h = \sigma_h^2\mathbf{I}$$

The solution, variance and resolution inferred from 10^6 realizations drawn from this distribution compare well with a reference solution calculated using Generalized Least Squares (Figure 1).

We then examine a second, weakly nonlinear test scenario, created from the first by adding a quadratic term to the data equation:

$$d_i = \sum_j G_{ij}^0 m_j + q_0 \sum_j Q_{ij} m_j^2$$

The parameter q_0 quantifies the strength of the nonlinear interaction. The elements of $\mathbf{Q}$ are randomly chosen with $|Q_{ij}| < \max_{p,q}(|G_{pq}|)$. We set $q_0 = 0.01$, so that the problem is only weakly nonlinear, with a solution that differs by about 20% from the $q_0 = 0$ linear solution. A reference solution is computed using Linearized Generalized Least Squares, utilizing the gradient:

$$G_{ij} \approx \frac{\partial d_i}{\partial m_j} = G_{ij}^0 + 2 q_0 Q_{ij} m_j$$

and with a linearized estimate of $\mathbf{C}_m$ and $\mathbf{R}^{\mathbf{G}}$ calculated using the G_{ij} of the last iteration. The solution, variance and resolution inferred from 10^6 realizations drawn from the distribution compare well with those calculated from the reference solution.

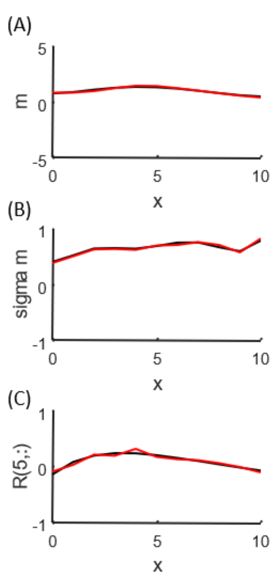

Figure 1. Results for the linear scenario. (A) The solution $\mathbf{m}$. (B) The standard deviation σ_m of the estimated model parameters. (C) The middle row (row 5) of $\mathbf{R}^{\mathrm{G}}$. Results for the Metropolis-Hastings method (red) compare well with Generalized Least Squares (black).

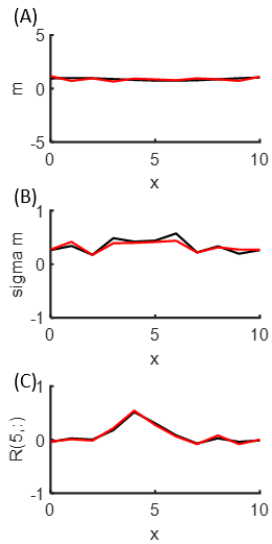

Figure 2. Results for the weakly nonlinear scenario. (A) The solution **m**. (B) The standard deviation σ_m of the estimated model parameters. (C) The middle row (row 5) of $\mathbf{R}^{G}$. Results for the Metropolis-Hastings method (red) compare well with Generalized Least Squares (black).

References

Menke, W., 2018. Geophysical Data Analysis: Discrete Inverse Theory, 4th Edition, Elsevier, 322pp.

MATLAB CODE

```
clear all;

LINEAR=0;

% data kernel
N=11;
x = [0:N-1]';
M=N;
G = zeros(N,M);
for i=[1:10]
    c = 0.03*i;
    v = exp(-c*x);
    G(i,:)=v'/sum(v);
end

% nonliner term
Q = random('Normal',0,1,N,M);
if( LINEAR )
    q0=0;
else
    q0 = 0.01;
end

% data
```

```
mtrue = 0.9*ones(M,1);
dtrue = G*mtrue;
for i=[1:M]
    dtrue(i) = dtrue(i) + q0*mtrue'*(squeeze(Q(:,i)).*mtrue);
end

sigmad = 1e-2;
dobs = dtrue + random('Normal',0,sigmad,N,1);
Cd = (sigmad^2)*eye(N,N);
Cdi = (sigmad^(-2))*eye(N,N);
Cdi2 = (sigmad^(-1))*eye(N,N);

% prior information
K=M;
H = toeplitz( [1; -1; zeros(K-2,1)], [1, zeros(1,M-1)] );
h = H*mtrue;
sigmah = 1;
Ch = (sigmah^2)*eye(K,K);
Chi = (sigmah^(-2))*eye(K,K);
Chi2 = (sigmah^(-1))*eye(K,K);

% Generalized least squared on the data kernel only
F = [Cdi2*G; Chi2*H];
f = [Cdi2*dobs; Chi2*h];
Ai = inv(F'*F);
mest = Ai*(F'*f);
Cm = Ai;
R = Ai*G'*Cdi*G;

% solution of the nonlinear problem by Newton's Method
mg = mest;
fi = [Cdi2*dobs; Chi2*h];
for itt=[1:50]
    dg = G*mg;
    Gi = G;
    for i=[1:M]
        dg(i) =   dg(i)   + q0*mg'*(squeeze(Q(:,i)).*mg);
        Gi(i,:) = Gi(i,:) + (2*q0*squeeze(Q(:,i)).*mg)';
    end
    Df = fi - [Cdi2*dg; Chi2*H*mg];
    Fi = [Cdi2*Gi; Chi2*H];
    Dm = (F'*F)\(F'*Df);
    mg = mg + Dm;
end

% recompute data and gradient
dg = G*mg;
Gi = G;
for i=[1:M]
    dg(i) = dg(i) + q0*mg'*(squeeze(Q(:,i)).*mg);
    Gi(i,:) = Gi(i,:) + (2*q0*squeeze(Q(:,i)).*mg)';
end

% solution, covariance, resolutin
mNM = mg;
Fi = [Cdi2*Gi; Chi2*H];
CmNM = inv(Fi'*Fi);
RNM = eye(M,M) - CmNM*H'*Chi*H;

% [mtrue, mest, mg]
% [dobs, dg, dobs-dg]
mquality=(mtrue-mg)'*(mtrue-mg)/(mtrue'*mtrue);
dquality=(dobs-dg)'*(dobs-dg)/(dobs'*dobs);
```

```
fprintf('Difference of nonlinear soln from linear soln %f\n', (mg-mest)'*(mg-
mest)/(mest'*mest) );
fprintf('Relative error of nonlinear soln %f\n', mquality);
fprintf('Relative error of nonlinear fit to data %f\n', dquality);

if( dquality > 0.05 )
    % stop if failed to fit the data adequately
    xxx
end

%statistics
Nadopts=0;
counts=0;
msum = zeros(M,1);
mprod = zeros(M,M);

% starting guess
m = random('Normal',mNM,0.1,M,1);
dpre = G*m;
for i=[1:M]
    dpre(i) = dpre(i) + q0*m'*(squeeze(Q(:,i)).*m);
end
ed = Cdi2*(dobs-dpre);
eh = Chi2*(h-H*m);
E = ed'*ed + eh'*eh;
logp = -0.5*E;

figure(1);
clf;

subplot(3,1,1);
hold on;
set(gca,'LineWidth',2);
set(gca,'FontSize',14);
xlabel('x');
ylabel('m');
axis([x(1), x(end), -5, 5 ]);
plot(x,mNM,'k-','LineWidth',2);

subplot(3,1,2);
hold on;
set(gca,'LineWidth',2);
set(gca,'FontSize',14);
xlabel('x');
ylabel('sigma m');
axis([x(1), x(end), -sigmah, sigmah ]);
plot(x,sqrt(diag(CmNM)),'k-','LineWidth',2);

subplot(3,1,3);
hold on;
set(gca,'LineWidth',2);
set(gca,'FontSize',14);
xlabel('x');
ylabel('R(5,:)');
axis([x(1), x(end), -1, 1 ]);
plot(x,RNM(5,:)','k-','LineWidth',2);

% iterations
Nr = 1000000;
%Nr = 100000;
Ntrain = floor(Nr/10);
for i=[1:Nr]
```

```
    % successor state
    ms = m + random('Normal',0,0.1,M,1);
    ds = G*ms;
    for ii=[1:M]
        ds(ii) = ds(ii) + q0*ms'*(squeeze(Q(:,ii)).*ms);
    end
    eds = Cdi2*(dobs-ds);
    ehs = Chi2*(h-H*ms);
    Es = eds'*eds + ehs'*ehs;
    logps = -0.5*Es;
    test = exp(logps - logp);

    % MH test for adopting the successor
    adopt = 0;
    if( test>=1 )
        adopt=1;
    else
        test2 = random('uniform',0,1,1,1);
        if( test>test2 )
            adopt=1;
        end
    end
    if( adopt )
        m = ms;
        ed = eds;
        eh = ehs;
        E = Es;
        logp = logps;
        Nadopts = Nadopts+1;
    end

    % statistics
    if( i>Ntrain )
        counts = counts+1;
        msum = msum + m;
        mprod = mprod + m*m';
    end

end

mmean = msum / counts;

CmMH = zeros(M,M);
for ii = [1:M]
for jj = [1:M]
     CmMH(ii,jj) = (mprod(ii,jj)-msum(ii)*mmean(jj)-
mmean(ii)*msum(jj)+counts*mmean(ii)*mmean(jj))/(counts-1);
end
end

RMH = eye(M,M) - CmMH*H'*Chi*H;

subplot(3,1,1);
plot(x,mmean,'r-','LineWidth',2);
subplot(3,1,2);
plot(x,sqrt(diag(CmMH)),'r-','LineWidth',2);
subplot(3,1,3);
plot(x,RMH(5,:)','r-','LineWidth',2);
```