Generalized Least Squares Fitting of a Exponential Function with Prior Constraints
W. Menke, Jan 18, 2014

This work arose from a question a correspondent asked me.

We are given the formula:

$$d_i = \rho \exp(m + r\sin(h_i)) + n_i$$

Where $\rho,\ r$ are known scalars, d_i, h_i are data and auxiliary variable, n_i is Normally-distributed noise, and m is an unknown parameter. We are also given a prior estimate, $\langle m\rangle$ which has prior variance $[\operatorname{cov} m]_A = \sigma_m^2$.

We will solve this problem using linearized Generalized Least Squares, as in Equation 9.21c of Menke's Geophysical Data Analsysis (3rd Edition) book. Le $m^{(p)}$ be the current estimate for $m^{(p)}$, where $m^{(0)}$ is an initial guess. Then define:

$$g_i(m^{(p)}) = \rho \exp\left(m^{(p)} + r\sin(h_i)\right)$$

which has derivative

$$G_i^{(p)} = \left.\frac{\partial g_i}{\partial m}\right|_{m^{(p)}} = \rho \exp\left(m^{(p)} + r\sin(h_i)\right)$$

The solution is obtained by iterating the formula

$$\begin{bmatrix} [\operatorname{cov}\mathbf{d}]^{-1/2}\mathbf{G}^{(p)} \\ [\operatorname{cov}\mathbf{m}]_A^{-1/2}\mathbf{I} \end{bmatrix} \mathbf{m}^{(p+1)} = \begin{bmatrix} [\operatorname{cov}\mathbf{d}]^{-1/2}\{\mathbf{d} - \mathbf{g}\left(\mathbf{m}^{(p)}\right) + \mathbf{G}^{(p)}\mathbf{m}^{(p)}\} \\ [\operatorname{cov}\mathbf{m}]_A^{-1/2}\langle\mathbf{m}\rangle \end{bmatrix}$$

to yield $m^{est} = m^{(p^{final})}$.

Example 1 (top figure; weak prior, since variance is large)

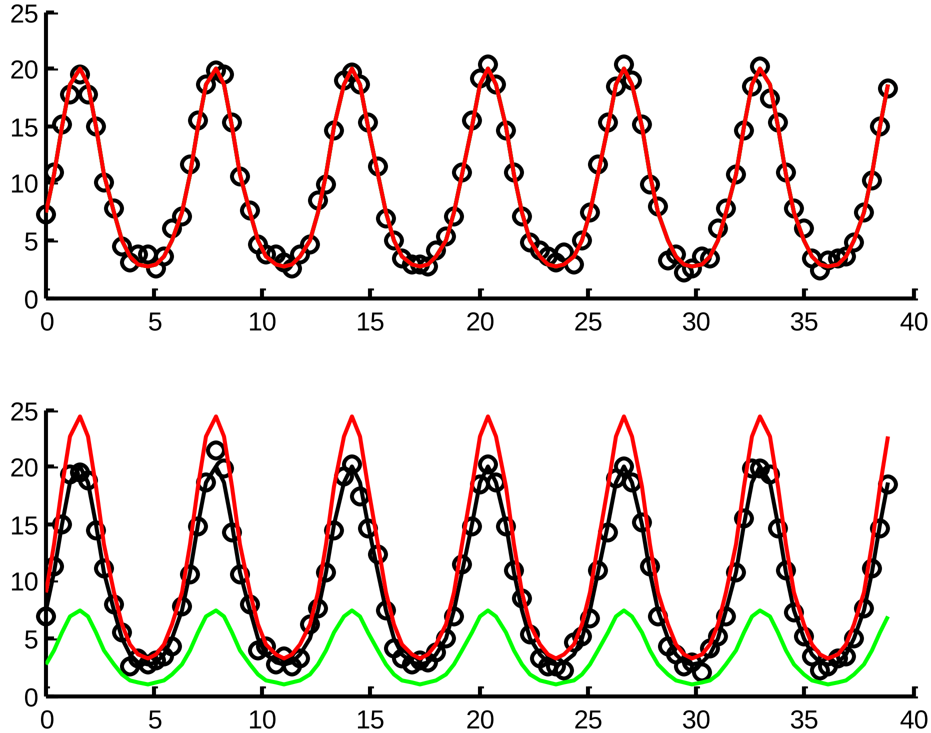
$m^{true} = 2,\ \langle m\rangle = 2.2, \sigma_m = 0.5, m^{(0)} = 3, m^{est} = 1.995$

Note $m^{est} \approx m^{true}$.

Example 2 (bottom figure; strong prior, since variance is small)

$m^{true} = 2, \langle m\rangle = 2.2, \sigma_m = 0.5, m^{(0)} = 1, m^{est} = 2.197$

Note $m^{est} \approx \langle m\rangle$.



Fugure. Plots of d_i, h_i. (Top) Example 1, with $\mathbf{d}^{true}$ (black curve), $\mathbf{d}^{obs}$ (black circles), $\mathbf{g}(m^0)$ (green curve), $\mathbf{g}(m^{est}) = \mathbf{d}^{pre}$ (red curve). (Bottom) Example 2.

MatLab script is attached.

```
clear all

% simplified equation
% d(i) = N(i) = rho * exp( a + r*sin(h(i)))
% that is, with N, h paired vectors and rho, r scalars

% this version uses generalized  least squares with prior info on parameter a

% make a vector of h's
N=100;
Dh = pi/8;
h = Dh*[0:N-1]';

% define scalars as unity
r = 1.0;
rho = 1.0;

% case 1
a = 2.0;
% prior information
a1_prior = 2.2; % prior value of parameret a
s1_prior = 0.5; % prior standard deviation of parameter a

d1_true = rho*exp( a + r * sin(h)); % true d
s = 0.5; % noise level for d
d1_obs = d1_true + random('Normal',0,s,N,1); % observed d is noisy
% top plot
% black curve: true data
% black circles: observed (noisy) data
% green curve: predicted data based on wildly wrong initial guess of a
% red curve: predicted data based on least-squared estimate of a
figure(1);
clf;
subplot( 2, 1, 1 );
set(gca, 'LineWidth', 2 );
axis( [0, 40, 0, 25] );
hold on;
plot( h, d1_true, 'k-', 'LineWidth', 2 );
plot( h, d1_obs, 'ko', 'LineWidth', 2 );

% interative, linearised least squared for a single parameter a
a1_guess = 3;
a1_guess0 = 3;
d1_guess = rho*exp( a1_guess + r * sin(h));
plot( h, d1_guess, 'g-', 'LineWidth', 2 );
for itt=[1:5]
G1 = rho*exp( a1_guess + r * sin(h)); % matrix of devivatives dd/da
Dd1 = d1_obs - rho*exp( a1_guess + r * sin(h)); % deviation of predicted data
from true data
F = [G1; 1/s1_prior];
f = [Dd1+G1*a1_guess; a1_prior/s1_prior];
a1_guess = (F'*F)\(F'*f); % generalized least squares estimate  parameter a
(Menke,eqn 9.21c)
end
a1_est = a1_guess;
```

```
d1_pre = rho*exp( a1_est + r * sin(h));
plot( h, d1_pre, 'r-', 'LineWidth', 2 );
fprintf('case 1: true a %f, prior %f, sigma %s, guess a %f, est a %f\n', a,
a1_prior, s1_prior, a1_guess0, a1_est);

% case 2
a = 2.0;
a2_prior = 2.2; % prior value of parameret a
s2_prior = 0.001; % prior standard deviation of parameter a
d2_true = rho*exp( a + r * sin(h));
d2_obs = d2_true + random('Normal',0,s,N,1);
subplot( 2, 1, 2 );
set(gca, 'LineWidth', 2 );
axis( [0, 40, 0, 25] );
hold on;
plot( h, d2_true, 'k-', 'LineWidth', 2 );
plot( h, d2_obs, 'ko', 'LineWidth', 2 );
a2_guess = 1;
a2_guess0 = 1;
d2_guess = rho*exp( a2_guess + r * sin(h));
plot( h, d2_guess, 'g-', 'LineWidth', 2 );
for itt=[1:5]
G2 = rho*exp( a2_guess + r * sin(h)); % matrix of devivatives dd/da
Dd2 = d2_obs - rho*exp( a2_guess + r * sin(h)); % deviation of predicted data
from true data
F = [G2; 1/s2_prior];
f = [Dd2+G2*a2_guess; a2_prior/s2_prior];
a2_guess = (F'*F)\(F'*f); % generalized least squares estimate  parameter a
(Menke,eqn 9.21c)
end
a2_est = a2_guess;
d2_pre = rho*exp( a2_est + r * sin(h));
plot( h, d2_pre, 'r-', 'LineWidth', 2 );
fprintf('case 2: true a %f, prior %f, sigma %f, guess a %f, est a %f\n', a,
a2_prior, s2_prior, a2_guess0, a2_est);
```