# Data Winnowing via Importance

Bill Menke, April 21, 2020

Consider a linear inverse problem $\mathbf{Gm} = \mathbf{d}$ for data $\mathbf{d}$ (of length $N$) and model parameters $\mathbf{m}$ (of length $M$). The estimated model parameters are $\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}}$, where $\mathbf{G}^{-g}$ is a generalized inverse, and their covariance is $\mathbf{C}$. The problem that I am considering is how to select the subset $\hat{\mathbf{d}}$, of length $\widehat{N} < N$, that leads to model parameters $\widehat{\mathbf{m}}^{\text{est}}$ having a covariance $\hat{\mathbf{C}}$ that is as close as possible to $\mathbf{C}$.

The predicted data is $\mathbf{d}^{\text{obs}} = \mathbf{Gm}^{\text{est}}$. Inserting $\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}}$ into this equation yields $\mathbf{d}^{\text{pre}} = \mathbf{GG}^{-g}\mathbf{d}^{\text{obs}} \equiv \mathbf{Nd}^{\text{obs}}$. When data resolution matrix $\mathbf{N} \equiv \mathbf{GG}^{-g}$ is unequal to the identity matrix, each predicted data is a non-trivial linear combination of the observed data. The importance vector $\mathbf{n} \equiv \text{diag}(\mathbf{N})$ quantifies the extent to which a datum contributes to its own prediction.

My idea is to solve the problem with all $N$ data, compute the importance $\mathbf{n}$, and then remove the datum with the least importance, leading to a dataset of length $N - 1$. The process is repeated until the length $\widehat{N}$ is reached.

The method seems to work on test cases of a strait line fit and a quadratic fit. Some theoretical development is going to be needed to understand the general case.

Example 1. Straight line $d_i = m_1 + x_i m_2$, where $x$ is an auxiliary variable, $N = 101$, each datum has a different prior variance $\sigma_{d_i}^2$, and $\hat{N} = 6$. Note in Figure 1 that the procedure selects low-error data from the ends of the interval.

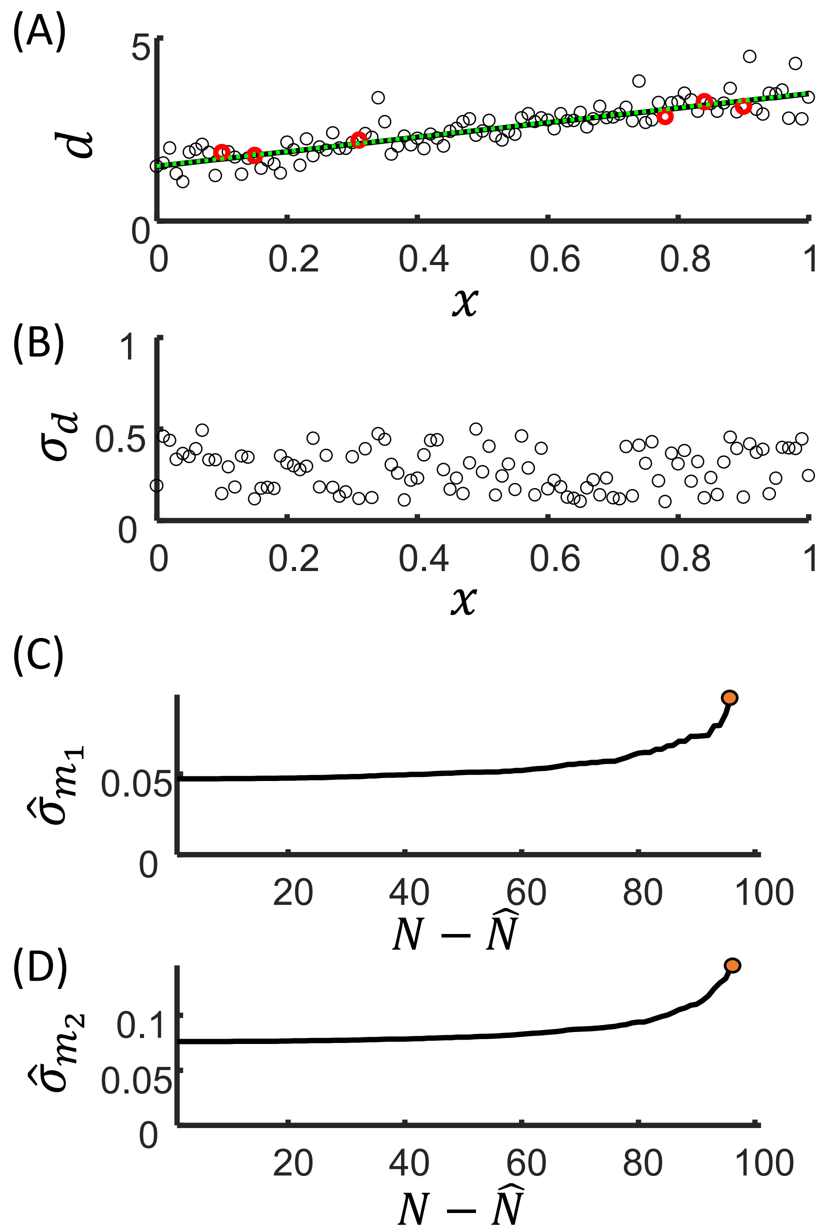


Fig. 1. (A) The observed data $\mathbf{d}^{\text{obs}}$ (black circles), the predicted data $\mathbf{d}^{\text{pre}}$ (black line) and the predicted data (green line) based on the winnowed data $\hat{\mathbf{d}}^{\text{obs}}$ (red circles). (B) Square root of the prior variance of $\mathbf{d}^{\text{obs}}$. (C) and (D) Square root of the posterior variances of $m_1$ and $m_2$, respectively, as a function of $\hat{N}$.

Example 1. Same as the previous example, but for the quadratic curve $d_i = m_1 + x_i m_2 + x_i^2 m_3$. The solution is by weighted least-squares. Note in Figure 1 that the procedure selects low-error data from the ends of the interval and at its center.

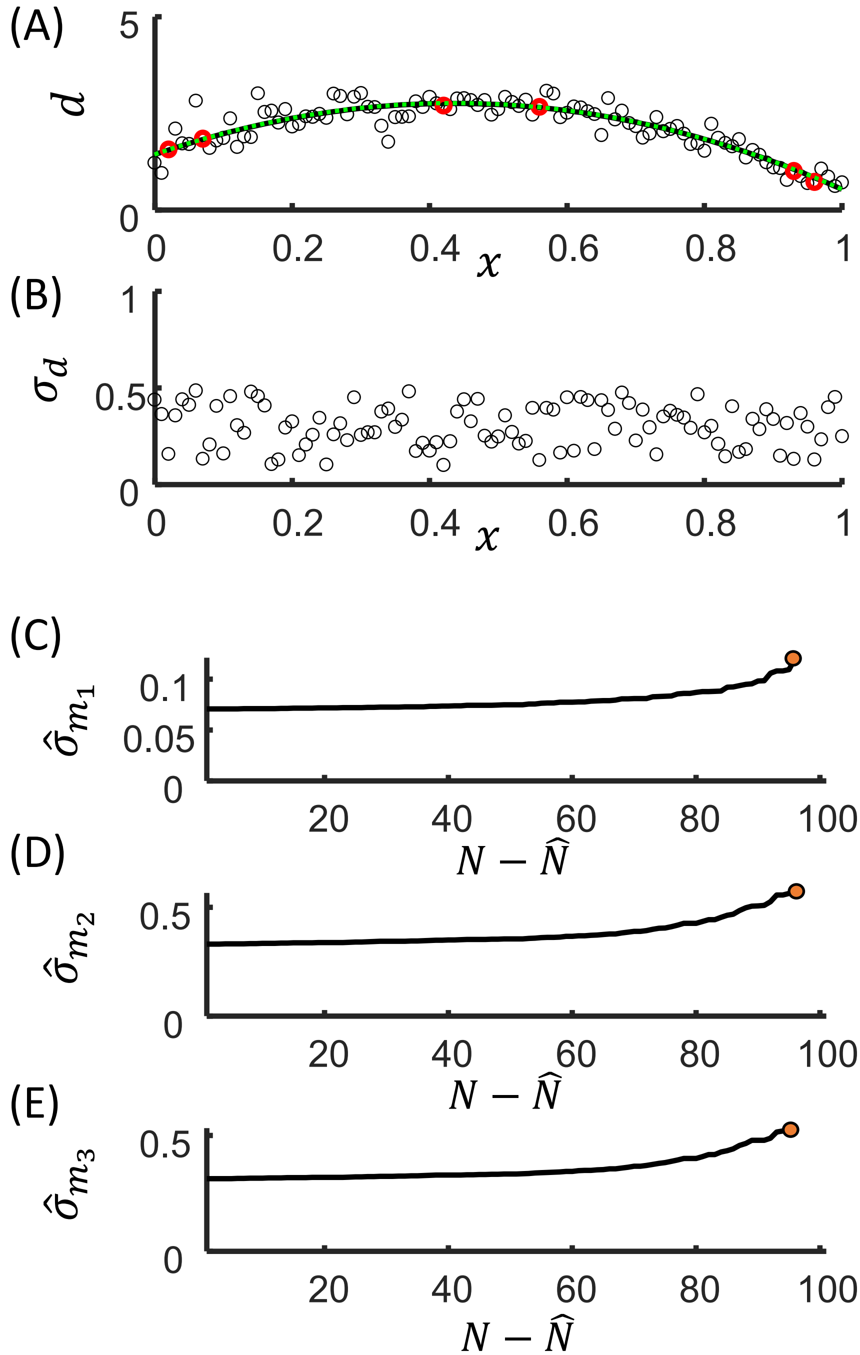


Fig. 1. (A) The observed data $\mathbf{d}^{\text{obs}}$ (black circles), the predicted data $\mathbf{d}^{\text{pre}}$ (black line) and the predicted data (green line) based on the winnowed data $\hat{\mathbf{d}}^{\text{obs}}$ (red circles). (B) Square root of the prior variance of $\mathbf{d}^{\text{obs}}$. (C), (D) and (E) Square root of the posterior variances of $m_1$, $m_2$ and $m_3$, respectively, as a function of $\widehat{N}$.

# Core Part of Algorithm

```
% weighted least squares inversion
W = diag(sigmad.^(-2) );
GMG = (G'*W*G)\(G'*W);
mest = GMG*dobs;
dpre = G*mest;
Cm = GMG * diag(sigmad.^(2)) * GMG';
NN = G * GMG;  % data resolution matrx
n = diag(NN);  % data imporantance

% set up for "whittling away" iteration
Nshort = N;
xshort = x;
dshort = dobs;
sigmadshort = sigmad;
mshort = mest;
NNshort = NN;
nshort = n;
Cmshort = Cm;
Cmall = zeros(N,M,M);
Cmall(1,:,:)=Cm;
fprintf('%d m %.2f %.2f cov slope %.4f\n', Nshort, mshort(1),
mshort(2), Cmshort(2,2) );

% "whittling away" iteration
for iter = [2:N-Nfinal+1]
[nsort, irow] = sort(nshort, 'descend' );
Nshort = Nshort-1;
irow = irow(1:Nshort);
dshort = dshort(irow);
xshort = xshort(irow);
sigmadshort = sigmadshort(irow);
Gshort = [ones(Nshort,1),xshort];
Wshort = diag(sigmadshort.^(-2) );
GMGshort = (Gshort'*Wshort*Gshort)\(Gshort'*Wshort);
mshort = GMGshort*dshort;
dpreshort = Gshort*mshort;
Cmshort = GMGshort * diag(sigmadshort.^(2)) * GMGshort';
Cmall(iter,:,:)=Cmshort;
NNshort = Gshort * GMGshort;  % data resolution matrx
nshort = diag(NNshort);  % data imporantance
fprintf('%d m %.2f %.2f cov slope %.4f\n', Nshort, mshort(1),
mshort(2), Cmshort(2,2) );
end
```