Lecture 16: Linear estimation theory (Foundations of objective mapping)

Recap

We've been looking at strategies to use our knowledge of the relationship between two data fields in order to infer the parameters that describe this relationship. For example, we used estimates of sea level rise to fit a functional form to the data, and we looked at the correlation between different variables. Now we're going to flip this relationship around to ask how to use prior knowledge about the covariance of two different quantities in order to infer a quantity that we didn't measure. The quintessential example of this comes from the challenge of mapping irregularly spaced Argo data (our measured quantity) onto a regular grid (unmeasured values).

To get started, we'll return to the definitions that we wrote down when we first considered correlation and covariance.

Linear estimation theory

Sometimes we may expect on theoretical grounds that there is a linear relationship between observable variables. In this case, we may want to find the best linear model. Let

$$\hat{y} = \alpha x \tag{1}$$

be the prediction of y where the variables x and y have zero mean

$$\langle x \rangle = \langle y \rangle = 0 \tag{2}$$

Our goal is to choose a value for α that is optimum in some sense. Previously we did this by computing the covariance between x and y, but now we're going to assume that we don't know y. Nonetheless, we'll proceed in the same way that we did when we derived expressions for correlation or least squares fits. Define the error as the difference between the model and the observed y:

$$\epsilon = \hat{y} - y \tag{3}$$

A reasonable optimization criterion is to minimize the **mean square error** (MSE)

$$\langle \epsilon^2 \rangle = \langle (\hat{y} - y)^2 \rangle = \langle (\alpha x - y)^2 \rangle = \alpha^2 \langle x^2 \rangle - 2\alpha \langle xy \rangle + \langle y^2 \rangle. \tag{4}$$

To minimize this with respect to α, we differentiate by α and set the result to zero.

$$\frac{\partial \langle \epsilon^2 \rangle}{\partial \alpha} = 2\alpha \langle x^2 \rangle - 2\langle xy \rangle. \tag{5}$$

Thus α, sometimes called the gain, is

$$\alpha = \frac{\langle xy \rangle}{\langle x^2 \rangle}. \tag{6}$$

This should be familiar: we derived this previously when we looked at regression coefficients.

Recall the definition of the correlation

$$\rho = \frac{\langle xy \rangle}{\sqrt{\langle x^2 \rangle \langle y^2 \rangle}}. \tag{7}$$

With this value of α, the MSE is

$$\langle \epsilon^2 \rangle = \langle y^2 \rangle \left[1 - \rho^2\right] = \langle y^2 \rangle - \langle \hat{y}^2 \rangle. \tag{8}$$

So the MSE is simply the difference between the variance in y and the variance explained by the model $\hat{y}$. As we noted before, the skill is the fraction of variance explained by the model

$$\frac{\langle \hat{y}^2 \rangle}{\langle y^2 \rangle} = \frac{\langle xy \rangle^2}{\langle x^2 \rangle \langle y^2 \rangle} = \rho^2. \tag{9}$$

Note that the error ϵ is uncorrelated with x:

$$\langle \epsilon x \rangle = 0 \tag{10}$$

The statistical model expressed above has a smaller MSE than any other dynamical or statistical model could have given the same data x and variable to predict y. The statistics needed for the prediction (6) are the covariance $\langle xy \rangle$ betwen the data and the variable that we want to predict, and the variance of the data $\langle x^2 \rangle$. To evaluate the skill (9) of the prediction, we also need the variance of the predicted variable $\langle y^2 \rangle$. As long as these statistics are available, the prediction is statistically optimum. A linear statistical model can also be used to improve the prediction of a dynamical model. Using the output of the dynamical model as the data x, and statistics of the covariance of the model output with predicted variables, a linear statistical model is guaranteed to improve the prediction. This sort of approach is central to weather prediction.

Nonzero mean

So far, we have considered only variables with zero mean (2). If x and y have nonzero means, what would be the best model? Consider the model

$$\hat{y} = \alpha x + \beta, \tag{11}$$

where the constant β is added to take into account nonzero means. The mean square error is

$$\langle \epsilon^2 \rangle = \langle (\alpha x + \beta - y)^2 \rangle. \tag{12}$$

Differentiate (12) with respect to α and β, and set the results to zero

$$\begin{aligned} \frac{\partial \langle \epsilon^2 \rangle}{\partial \alpha} &= 2\langle x(\alpha x + \beta - y) \rangle = 2\alpha \langle x^2 \rangle + 2\beta \langle x \rangle - 2\langle xy \rangle && (13) \\ \frac{\partial \langle \epsilon^2 \rangle}{\partial \beta} &= 2\langle \alpha x + \beta - y \rangle = 2\alpha \langle x \rangle + 2\beta - 2\langle y \rangle && (14) \end{aligned}$$

Equations (13-14) are solved for the unknowns α and β:

$$\begin{aligned} \alpha &= \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} && (15) \\ \beta &= \langle y \rangle - \alpha \langle x \rangle && (16) \end{aligned}$$

Writing the variables in terms of their means and fluctuations

$$\begin{aligned} x &= \langle x \rangle + x' && (17) \\ y &= \langle y \rangle + y' && (18) \\ & && (19) \end{aligned}$$

we find the gain to be

$$\alpha = \frac{\langle x'y' \rangle}{\langle x'^2 \rangle}. \tag{20}$$

Then the optimum model is, using (16) in (11)

$$\hat{y} = \alpha(x - \langle x \rangle) + \langle y \rangle \tag{21}$$

or equivalently

$$\hat{y}' = \alpha x'. \tag{22}$$

So the optimum model is just as before for variables with zero mean. Thus, the best approach is always to take the means out of data before proceeding with linear estimation.

Multiple variables

Consider the prediction of a scalar y from a vector of data $\mathbf{x}$. A linear model would be

$$\hat{y} = \mathbf{a}^T \mathbf{x} \tag{23}$$

where $\mathbf{a}$ is the gain vector. The MSE is

$$\langle \epsilon^2 \rangle = \langle (\hat{y} - y)^2 \rangle = \langle (\mathbf{a}^T \mathbf{x} - y)^2 \rangle. \tag{24}$$

Minimizing

$$\begin{aligned} \frac{\partial \langle \epsilon^2 \rangle}{\partial \mathbf{a}} &= 2\langle \mathbf{x}\mathbf{x}^2 \rangle - 2\langle \mathbf{x} y \rangle = 0 && (25) \\ \mathbf{a} &= \langle \mathbf{x}\mathbf{x}^T \rangle^{-1} \langle \mathbf{x} y \rangle. && (26) \end{aligned}$$

The MSE is then

$$\langle (\hat{y} - y)^2 \rangle = \langle y^2 \rangle - \langle y \mathbf{x}^T \langle \mathbf{x}\mathbf{x}^T \rangle^{-1} \langle \mathbf{x} y \rangle = \langle y^2 \rangle - \langle \hat{y}^2 \rangle, \tag{27}$$

and the skill is

$$\frac{\langle \hat{y}^2 \rangle}{\langle y^2 \rangle} = \frac{\langle y \mathbf{x}^T \langle \mathbf{x}\mathbf{x}^T \rangle^{-1} \langle \mathbf{x} y \rangle}{\langle y^2 \rangle}. \tag{28}$$

The solution above obviously requires the data covariance matrix $\langle \mathbf{x}\mathbf{x}^T \rangle$ to be invertible. This is understood as each datum providing at least some new information, even if the data are not uncorrelated with each other. A completely redundant datum would result in a rank deficient, therefore singular, matrix. As real data almost always have some noise, the noise shows up as an augmentation of the data–data covariance matrix along the diagonal. In essence, we add $\sigma^2 \mathbf{I}$ to the theoretical covariance matrix. The large non-zero diagonal of the covariance matrix means that it is almost always invertible.

The equations derived here describe the core method used to map irregularly spaced data—what oceanographers call “objective mapping” and statisticians call “kriging”. The core principle is that we use a priori knowledge of the data–data and data–model covariances to make inferences about unknown quantities.

Testing with interpolation

Linear interpolation is a good place to start testing our objective mapping procedure. Suppose we measure at two data points: $t = +1$ and $t = -1$. Then our mapped values will depend on the covariances beween our observations ($x(+1)$ and $x(-1)$) and between the observations and the mapped quantites.

For testing purposes, we'll assume a Gaussian covariance of the form

$$\langle x(t_1)x(t_2)\rangle = A\exp\left[\frac{-(t_1-t_2)^2}{T^2}\right]. \tag{29}$$

You'll notice that in this case, the covariance only depends on the spatial separation $t_1 - t_2$. We can write this as a matrix.

Since we want to predict y, we'll also need the covariance between y and x:

$$\langle yx\rangle = A\begin{bmatrix}\exp\left[\frac{-(t-t_1)^2}{T^2}\right]\\ \exp\left[\frac{-(t-t_2)^2}{T^2}\right]\end{bmatrix}. \tag{30}$$

In class we took a quick look at the demo "intgauss.m", which shows the result of choosing different values of the time scale T. As T increases, the interpolation approaches a straight line, and the skill improves. At short time scales, both the estimate and the skill fade rapidly to zero with distance from the data.

Mapped values relax to zero far from data, so unless our background state is really zero, we want to make sure that we're removing a mean from the data and that we focus on mapping anomalies.

```
%intgauss.m (Dan Rudnick)

%Linear estimate interpolation using gaussian correlation.
%
% PARAMETER INPUT
d=input('Data (2-vector for values at t=[-1 1])? ');
scale=input('e-folding scales? ');
noise=input('Noise? ');

% INITIALIZE ARRAYS
d=d(:);
t=(-5:0.1:5)';
skill=zeros(length(t),length(scale));
x=zeros(size(skill));
skillt=zeros(size(skill));
xt=zeros(size(skill));

% DEFINE COVARIANCE
for n=1:length(scale)
  %  define data-data covariance matrix (2x2)
   cov=[1+noise exp(-(2/scale(n))^2); exp(-(2/scale(n))^2) 1+noise];
  %  define data-model covariance matrix (2xN)--2 values for each location 
   ct=[exp(-((t+1)/scale(n)).^2) exp(-((t-1)/scale(n)).^2)];

% --baseline solution
   skill(:,n)=diag(ct/cov*ct');
   x(:,n)=ct/cov*d;
```

```
% --covariance matrix and solution for time derivative
   ctt=-2/(scale(n).^2)*[t+1 t-1].*ct;
   skillt(:,n)=diag(ctt/cov*ctt')/(2/(scale(n).^2));
   xt(:,n)=ctt/cov*d;
end

% PLOT RESULTS
% --Figure 1:  fitted data and skill
figure;
subplot(2,1,1)
plot(t,x),xlabel('t'),ylabel('x');
subplot(2,1,2)
plot(t,skill),xlabel('t'),ylabel('skill')

% --Figure 2:  fitting time derivative instead of raw data
figure;
subplot(2,1,1)
plot(t,xt),xlabel('t'),ylabel('dxdt');
subplot(2,1,2)
plot(t,skillt),xlabel('t'),ylabel('skill-dxdt')
```