Lecture 7: Putting decorrelation scales into practice

Recap

In Lecture 6, we considered the random walk and implications for diffusion, and then we examined autocovariance, autocorrelation, and implications for degrees of freedom in an idealized case. Today, we'll put this into practice with some specific examples.

We defined the autocovariance as $C_{xx}(r)$ and noted that it has units and depends on the separation r between data points. The autocorrelation $\rho(r)$ is normalized to have a maximum value of 1 when $r = 0$, that is when data are correlated with themselves.

In the idealized case that we considered last time, we used an analytic form of the autocovariance and said that the decorrelation time—that is the time interval between statistically independent observations—could be computed by integrating the autocorrelation over its entire range.

$$\text{“decorrelation” scale} = \tau_{eff} = \int_{-\infty}^{\infty} \rho(t)\, dt. \tag{1}$$

This approach works well when data are well behaved, but in reality, the autocovariance can be poorly defined for large values of r.

Spread of a tracer concentration

First recall that the autocovariance is the convolution of a data set with itself. One definition of the autocovariance is:

$$C_{xx}(r) = \frac{\sum_{i=1}^{N-r}\sum_{j=1+r}^{N} x_i' x_j'}{N-r}. \tag{2}$$

This provides an “unbiased” estimate of the autocovarince—unbiased because we divide each covariance by the number of data pairs that are available at separation r.

The unbiased estimator is ill-behaved for large lags, because we have few data pairs to average. Thus more typically we use a “biased” estimator:

$$C_{xx}(r) = \frac{\sum_{i=1}^{N-r}\sum_{j=1+r}^{N} x_i' x_j'}{N}. \tag{3}$$

This tapers to zero for large lags.

The autocorrelation resembles the autocovariance, except that it is normalized by the data variance:

$$\rho_{xx}(r) = \frac{\sum_{i=1}^{N-r}\sum_{j=1+r}^{N} x_i' x_j'}{N\sigma^2}, \tag{4}$$

where σ is the standard deviation of x.

If we want to figure out how many degrees of freedom we have in a data set of length N, we need to figure out how redundant adjacent data points are—that is we need a “decorrelation” scale. There are three common ways to estimate this:

1. The first zero crossing—that is the first point when the autocorrelation decreases from a positive value to a negative value. This is conceptually easy, but fraught when the data are noisy. Unfortunately, it's easy to show that data with very different autocorrelation structure, and correspondingly very different decorrelation scales, can nonetheless have the same zero crossing.

2. The full-width at half-maximum of the autocorrelation. This is potentially a slightly less noisy-prone approach to obtaining a quick estimate of a decorrelation scale. For a well-behaved decorrelation, such as a triangle autocovariance, this will produce clean results.

3. As we discussed last time, the integral of the autocovariance ($\int_{-\infty}^{\infty} \rho(r)\,dr$) is a robust measure of decorrelation, though it also is subject to interpretation depending on noise and sampling limitations. We'll pursue this approach in more detail.

If we measure N data points, then the effective degrees of freedom should be N divided by the decorrelation scale:

$$N_E = \frac{N}{\sum_{n=-N}^{N} \rho(|n|)}. \tag{5}$$

In writing this, I'm assumed that $\rho(n)$ is well behaved at $\pm N$. If I used an unbiased estimator, then I'd want to suppress the uncertain values at high n

$$N_E = \frac{N}{\sum_{n=-N}^{N} \left(1 - \frac{|n|}{N}\right) \rho(|n|)}. \tag{6}$$

But in Matlab, the "xcov" default is the "biased" estimator, and we can ignore this.

So consider the case when data are random white noise, and adjacent points are uncorrelated. Then $\rho(n) = \delta(n)$ which is 1 only when $n = 0$, and

$$N_E = \frac{N}{\sum_{n=-N}^{N} \delta(|n|)} = N. \tag{7}$$

In other words, in this case, all data are independent.

More realistically, geophysical data tend to be correlated over time and space. Usually if we sum from $-N$ to $+N$, our results will be contaminated by noise, so we actually want to sum only over the center portion of the autocovarince.

$$N_E = \frac{N}{\sum_{n=-l}^{l} \rho(|n|)}, \tag{8}$$

where $l < N$. The trick is to decide on l: how many values should we sum? If we lack any other information, we can just test all possible l and look at the range of N_E values that we find. Usually we want to be as conservative as possible, so we assume the smallest possible N_E—that is the largest possible decorrelation scale.

If we write this continuously, we can express the values as integrals instea of sums. In other words the observed mean of x is:

$$\{x\} = \frac{1}{T} \int_0^T x(t)\,dt. \tag{9}$$

And in continuous form,

$$N_E = \frac{T}{\int_{-T}^{T} \rho(t)\,dt}. \tag{10}$$

The expected value of the variance is:

$$\begin{aligned} E_2 &= \frac{1}{T^2}\int_0^T\int_0^T \langle x'(t_1)x'(t+2)\rangle\, dt_1\, dt_2 && (11)\\ &= \frac{\sigma^2}{T}\int_{-T}^{T}\rho(|t|)\, dt && (12)\\ &= \frac{\sigma}{N_E}. && (13)\end{aligned}$$

This means that:

$$T_{eff} = \frac{T}{N_E} = \int -T^T \rho(t)\, dt. \tag{14}$$

In class we looked at a range of examples to understand how different studies have analyzed decorrelation, and how decorrelation can be sensitive to the large-scale variations in the data record.

Introduction to data and models

Now that we've built a statistical framework, let's put it to work to interpret our data. When we looked at correlation, we considered data y and we wanted to represent with an approximation $\hat{y} = \alpha x$, and we sought an α that would minimize the difference betwen y and $\hat{y}$. Now we want to expand to a more complicated case.

To do this, let's lay out a little notation. First, we need data. Let's write the data as an N-vector of real numbers

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{15}$$

There are necessarily a finite number of data.

In addition we need a model. We will consider discrete inverse theory, in which case there are a finite number of model parameters, an M-vector

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} \tag{16}$$

The relationship between the data and model parameters may, in general, be stated as a series of L equations

$$\mathbf{f}(\mathbf{d},\mathbf{m}) = \begin{bmatrix} f_1(\mathbf{d},\mathbf{m}) \\ f_2(\mathbf{d},\mathbf{m}) \\ \vdots \\ f_L(\mathbf{d},\mathbf{m}) \end{bmatrix} = \mathbf{0}. \tag{17}$$

The equality in (17) implies that both the data and model are perfect. We will relax this unlikely assumption as we find solutions. It is nearly always possible to write the relationship as:

$$\mathbf{d} = \mathbf{g}(\mathbf{m}) \tag{18}$$

where g is a vector function. If **g** is linear then we can expres this as a matrix equation.

$$\mathbf{d} = \mathbf{G}\mathbf{m} \tag{19}$$

where $\mathbf{G}$ is an $N \times M$ matrix.