Lecture 6: Diffusivity from random walk, autocovariance, and decorrelation

Recap

In Lecture 5, we examined variance ellipses as a means to characterize a joint pdf. We then considered the a random walk, which we linked to the central limit theorem. For the random walk, we considered the net displacement of a particle from time step 0 to time step N:

$$x_N = x_0 + \Delta t \sum_{n=1}^{N} v_n, \tag{1}$$

with

$$\langle v_n \rangle = 0. \tag{2}$$

Assuming we define $x_O = 0$ (for housekeeping purposes, then the mean position $\langle x_N \rangle$ will converge to 0. But the spread of possible values of x_N, that is the variance will increase linearly with total elapsed time $t_N = N\Delta t$:

$$\langle x_N^2 \rangle == (\Delta t)\langle v^2 t_N. \tag{3}$$

We employed this in the definition of diffusivity, which describes the time rate of change of the variance of position:

$$k = \frac{1}{2}\frac{d\langle x^2 \rangle}{dt} = \frac{1}{2}\langle v^2 \rangle \Delta t. \tag{4}$$

And we looked at numerical experiments that showed this amazing spread.

The second numerical case we looked at asked how the variance of position would change if we normalized it by the number of time steps—that is if we asked about the average displacement per time step. In that case, we saw that everything converged to zero—this is what we'd expect, and we'll explore this point a little further in this lecture.

Spread of a tracer concentration

Now that we've defined diffusivity, we can ask how a patch of tracer might spread out. The diffusivity k, plugged into our equation linking $\langle x^2 \rangle$ and $\langle v^2 \rangle$ tells us that

$$\langle x^2 \rangle = 2kt. \tag{5}$$

Let's consider a tracer of mass M released from the origin. Over time, it is pushed around by multiple random velocity impulses. Since the sum of many random events has a Gaussian distribution that depends on its variance $\langle x^2 \rangle$. Thus the tracer concentration Γ:

$$\Gamma(r,t) = \frac{M}{\sqrt{2\pi kt}} \exp\left(\frac{-r^2}{4kt}\right). \tag{6}$$

If this works, then the tracer distribution should be consistent with the diffusion equation:

$$\frac{\partial \Gamma}{\partial t} = k\frac{\partial^2 \Gamma}{\partial x^2}. \tag{7}$$

Does this work? Plug (6) into (7) to check.

Of course, in real world examples, tracer is also advected by a mean flow, so we should augment our diffusion equation with an advection term.

$$\frac{\partial \Gamma}{\partial t} + \vec{v} \cdot \nabla \Gamma = k\frac{\partial^2 \Gamma}{\partial x^2}. \tag{8}$$

Sampling errors

We started out this class by defining the true mean of our data:

$$\langle x \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{\infty} x_n \tag{9}$$

This definition makes sense, but it defies the reality of our world: we never have infinite data, and the systems we sample are not necessarily stationary (with consistent, invariant statistics) over all time. This means that we need to compute not true statistics, but **sample statistics**. We now wish to determine the accuracy of our sample statistics to aid in interpretation. The difference between the true and sampling statistics is called the **sampling error**.

The **sampling mean** is defined as

$$\{x\} = \frac{1}{N} \sum_{n=1}^{N} x_n. \tag{10}$$

Note the use of curly braces as a notation for the sample mean. The distinguishing factor between (9) and (10) is that the sample mean is computed over a finite number of realizations.

Consider the problem of estimating the true mean with the sample mean. Suppose the sample mean were calculated an infinite number of times, then we could consider the mean of the sample mean, and compare it to the true mean. The **bias** is defined to be the mean difference between the sample mean and the true mean:

$$E_1 = \langle \{x\} - \langle x \rangle \rangle = \langle \frac{1}{N} \sum_{n=1}^{N} x_n \rangle - \langle x \rangle = \frac{1}{N} \sum_{n=1}^{N} \langle x_n \rangle - \langle x \rangle = \frac{1}{N}(N \langle x \rangle) - \langle x \rangle = 0. \tag{11}$$

This tells us that the sample mean is unbiased relative to the true mean: in the limit of many samples, the sample mean converges to the true mean.

Now think about the variance of the the sample mean:

$$E_2 = \langle [\{x\} - \langle x \rangle]^2 \rangle = \langle \left(\frac{1}{N} \sum_{n=1}^{N} x_n - \langle x \rangle \right)^2 \rangle. \tag{12}$$

Moving the true mean inside the sum results in

$$E_2 = \langle \left(\frac{1}{N} \sum_{n=1}^{N} [x_n - \langle x \rangle] \right)^2 \rangle. \tag{13}$$

The quantity in square brackets in (**??**) is the fluctuation, so

$$E_2 = \frac{1}{N^2} \langle \left(\sum_{n=1}^{N} [x'_n] \right)^2 \rangle. \tag{14}$$

The average squared sum of fluctuations is the sum over all elements of the covariance matrix of the fluctuations

$$E_2 = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \langle x'_n x'_m \rangle. \tag{15}$$

Assuming the fluctuations are independent with equal variance then

$$E_2 = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\delta_{nm}\sigma^2 = \frac{\sigma^2}{N}. \tag{16}$$

So the variance decreases as the inverse of N. The square root of E_2 is called the **root-mean-square error** or the **standard error**, and decreases as the inverse of the square root of N:

$$\sqrt{E_2} = \frac{\sigma}{\sqrt{N}}. \tag{17}$$

Recognizing this improvement of error by $N^{-1/2}$ is key to understanding many measures of statistical uncertainty. This result also illustrates a parallel between sampling error and the random walk (which showed that variance of position expanded like N rather than N^2).

The rms error defined by E_2 depends on the true variance. Now let's consider the thornier situation when we are constrained by limited sampling. Our estimate of the variance of anomalies x' is:

$$\hat{\sigma}^2 = \left\{x'^2\right\}. \tag{18}$$

The mean of this estimate is:

$$\langle\hat{\sigma}^2\rangle = \frac{1}{N}\sum_{n=1}^{N}\langle x'^2\rangle = \sigma^2, \tag{19}$$

so this estimate of variance is unbiased. The variance of the estimate of variance is:

$$F_2 \equiv \langle\left(\left\{x'^2\right\} - \sigma^2\right)^2\rangle = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\langle x'^2_n x'^2_m\rangle - \sigma^4. \tag{20}$$

In estmating F_2, we've used (19) which means that the cross terms ($\langle\left\{x'^2\right\}\sigma^2\rangle$) produce $2\sigma^4$. Now assuming that the fluctuations x'_n and x'_m are independent,

$$F_2 = \frac{1}{N^2}\left[N(N-1)\sigma^4 + N\langle x'^4\rangle\right] - \sigma^4 = \frac{1}{N}\left(\langle x'^4\rangle - \sigma^4\right). \tag{21}$$

Thus the error in the variance depends on the fourth moment. This is a problem of closure. To determine the error of any statistic we need to know a higher order statistic.

Now, let's come back to the potential bias in our estimate $\hat{\sigma}^2$ in the real-world scenario when the true mean $\langle x\rangle$ is unknown, and we're stuck working with $\{x\}$. In this case, we know that we want $\hat{\sigma}^2$ to be unbiased relative to the true σ^2, but we might need to do a little adjustment. We'll define:

$$\hat{\sigma}^2 = A\left\{(x - \{x\})^2\right\}, \tag{22}$$

where A is an unknown adjustment parameter that we want to find. Then

$$\begin{aligned}
F_1 &= \langle\hat{\sigma}^2 - \sigma^2\rangle && (23)\\
&= A\langle\left\{(x - \{x\})^2\right\}\rangle - \sigma^2 && (24)\\
&= A\langle\left\{\left[(x - \langle x\rangle) - (\{x\} - \langle x\rangle)\right]^2\right\}\rangle - \sigma^2 && (25)\\
&= A\langle\left\{x'^2\right\} - (\{x\} - \langle x\rangle)^2\rangle - \sigma^2, && (26)
\end{aligned}$$

where we have assumed no correlation between errors in the mean ($\{x\} - \langle x \rangle$) and the individual anomalies ($x - \langle x \rangle$). This leads to

$$F_1 = A\langle \sigma^2 - \frac{\sigma^2}{N} \rangle - \sigma^2 \tag{27}$$

$$= A\sigma^2 \left(1 - \frac{1}{N} \right) - \sigma^2 \tag{28}$$

$$= 0, \tag{29}$$

implying that in order to obtain zero bias,

$$A = \frac{N}{N-1}. \tag{30}$$

In essence, when we compute the mean from the original data, we lose a degree of freedom, so we need to compute the standard deviation by assuming $N - 1$ degrees of freedom as a normalization, rather than N degrees of freedom. Software packages that compute standard deviation (e.g. Matlabs "std" function) take this into account.

Correlated uncertainties and degrees of freedom

We've been considering idealized situations in which random variables x_n are completely independet, so that the covariance of x is of the form:

$$\langle x'_n x'_m \rangle = \sigma^2 \delta_{nm}. \tag{31}$$

This works well if your random variables come from a random number generator, but not so well if they come from geophysical quantities that are sampled more rapidly in time (or space) than the system actually evolves. In reality variables that are closely spaced in time or space tend to be correlated. If the statistics are stationary, then the covariance between measurements might depend only on their separation. For example:

$$\langle x'_n x'_m \rangle = \sigma^2 \rho(|n - m|) \tag{32}$$

We can compute the autovariance of x with itself at multiple lags as a lagged covariance. (In Matlab you can do this with the function "xcov".) There are some things to keep in mind.

1. For a record of finite length, the number of data pairs that we can average to compute the autocovariance varies with the size of the lag. For a record with N values, at zero lag, the autocovariance is an average of N data pairs. At lag n, it is an average of $N - n$ data pairs. This means that our estimate of the autocovariance has larger uncertainty for larger lags.

2. To deal with the uncertainties in pairs, the default in Matlab is to compute a "biased" autocovariance, by scaling the autocovariance by $(N - |n|)/N$ so that the autocovariance tapers to zero at large lag.

3. You can also compute an "unbiased" autocovariance, which will compute the average covariance based on the available number of pairs. This works well near zero lag, when you can average over many pairs, but it has the downside of producing a highly uncertain covariance for large lag.

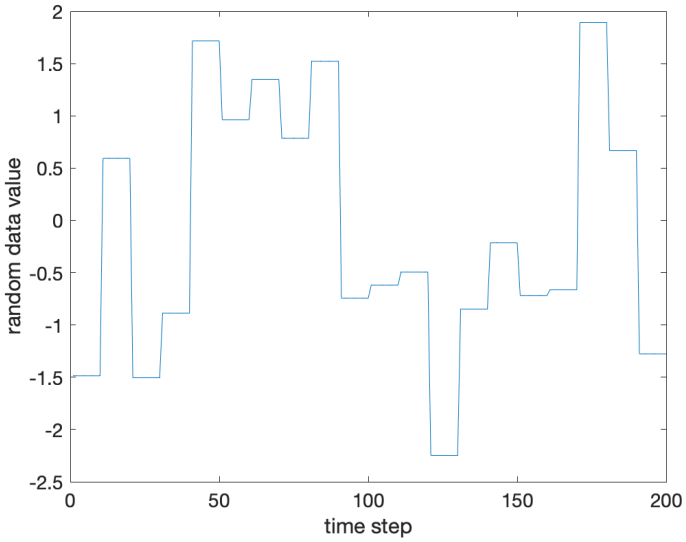
4. The autocovariance has units, corresponding to the units of the data squared. For many calculations, we want the autocorrelation rather than the autocovariance—that is we want an autocorrelation that varies between -1 and +1 and that is unitless. In Matlab, we can compute this using the "coeff" flag, which produces values consistent with correlation coefficients.

Now let's think about an example. Consider the case of random variables that are each repeated 10 times in our data record. In class we produced a data record using the following lines of Matlab code:

```
% first define a set of random numbers
N=1000;
x=randn(N,1);
% now make them repeat 10 times
y=x*ones(1,10);

% plot this out:  is it right?
plot(y(:))
%   no, this case visibly repeats the entire record 10 times.
%   that's just because when we convert from a matrix to a vector,
%   Matlab does this by concatenating the columns of the matrix.
%
%  If we were using python, the data storage would be inverted.

% take the transpose and plot a subset
z=y';
plot(1:200,z(1:200))
xlabel('time step','FontSize',14)
ylabel('random data value','FontSize',14)
h=gca;
set(h,'FontSize',14)
shg
```

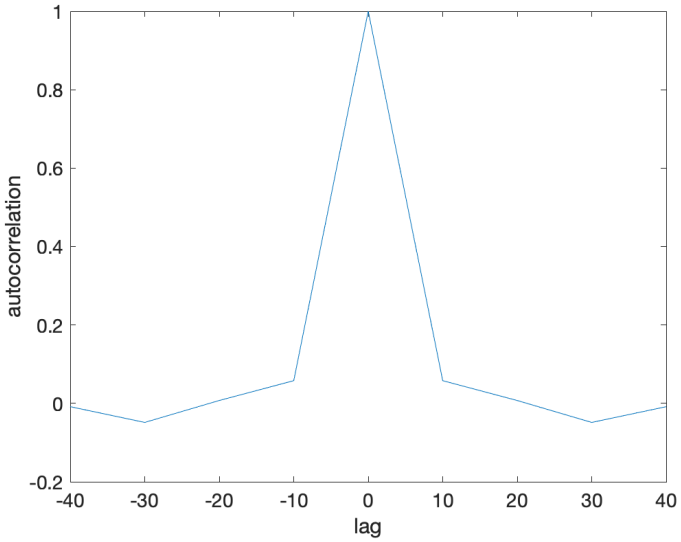


Once we have our data, which clearly have correlated consecutive values, we can compute the autocorrelation:

```
% compute the autocorrelation
Nbig=N*10;
plot(-Nbig+1:Nbig-1,xcov(z(:),'coeff'))
xlabel('lag','FontSize',14)
ylabel('autocorrelation','FontSize',14)
h=gca;
set(h,'FontSize',14)
```

The zero lag version of this is hard to read, but let's zoom in on the center of the distribution:

```
axis([-40 40 -.2 1])
```



This shows that the autocovariance is effectively a triangle rising from 0 at $\tau = -10$ to 1 at $\tau = 0$ and then falling to 0 at $\tau = 10$. This might not be what you'd expected, but it's not entirely surprising: the autocovariance represents a convolution of the data with itself, and the convolution of a boxcar distribution with itself is a triangle.

How many data steps should it take to obtain an independent value? In this case we know that every 10th value is independent, so our "decorrelation scale" is 10 measuremewnts, and the number of degrees of freedom $Neff$ is the total number of values in z ($Nbig$) divided by 10, which is N.

How can we show this more generally? There are a number of heuristic approaches that people use, but a good formalism is to integrate the autocovariance.

$$\text{"decorrelation" scale} = \tau_{eff} = \int_{-\infty}^{\infty} \rho(t)\, dt. \tag{33}$$

In this idealized example, that would yield:

$$\begin{aligned}
\tau_{eff} &= \int_{-\infty}^{\infty} \rho(t)\, dt && (34)\\
&= \int_{-10}^{0} \left(1 + \frac{t}{10}\right) + \int_{0}^{10} \left(1 - \frac{t}{10}\right) && (35)\\
&= \left(t + \frac{t^2}{20}\right)\Big|_{-10}^{0} + \left(t - \frac{t^2}{20}\right)\Big|_{0}^{10} && (36)\\
&= \left(10 - \frac{100}{20}\right) + \left(10 - \frac{100}{20}\right) = 10. && (37)
\end{aligned}$$

More details to come next time....