Time Series Basics and Operators

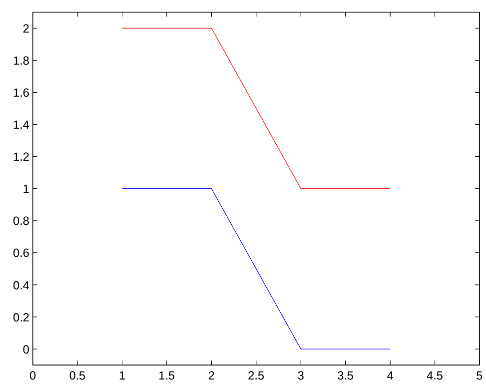
T. H. Merrett*

McGill University

February 20, 2008

[1]

1. Are these two series similar?



$$(2\ 2\ 1\ 1)$$
$$(1\ 1\ 0\ 0)$$

Yes. So, unlike vectors, adding a constant should not change anything.

Let's "translate" by subtracting the average value:

$$(\frac{1}{2}\frac{1}{2} - \frac{1}{2} - \frac{1}{2})$$

We'll just work with series that sum (and average) to 0 from now on.

2. (Digression on what this means in terms of vectors.)

Subtracting the average reduces the dimension of the space by 1, because there is now a linear dependence among the coefficients of a vector $v : v_1 + v_2 + ..v_n = 0$.

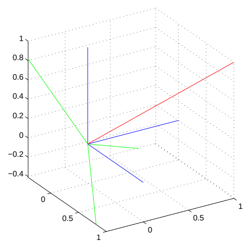
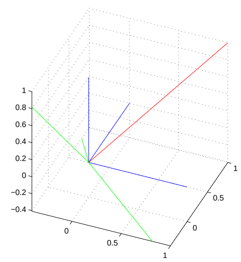
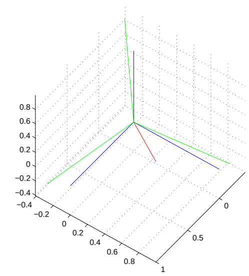
In two dimensions, a vector (c, s) becomes $\frac{c-s}{2}(1, -1)$: everything maps into the line $(1, -1)$.

In three dimensions, everything maps into the subspace orthogonal to (1,1,1). If its basis vectors are $B_1 = \frac{1}{\sqrt{2}}(1, -1, 0)$ and $B_2 = \frac{1}{\sqrt{6}}(1, 1, -2)$, then we have the mappings

$$\begin{array}{llll}
(1,0,0) \Rightarrow (\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3}) \Rightarrow & \frac{1}{\sqrt{2}}B_1 + \frac{1}{\sqrt{6}}B_2 & \Rightarrow (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{6}}) \\
(0,1,0) \Rightarrow (-\frac{1}{3}, \frac{2}{3}, -\frac{1}{3}) \Rightarrow & -\frac{1}{\sqrt{2}}B_1 + \frac{1}{\sqrt{6}}B_2 & \Rightarrow (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{6}}) \\
(0,0,1) \Rightarrow (-\frac{1}{3}, -\frac{1}{3}, \frac{2}{3}) \Rightarrow & -\sqrt{\frac{2}{(}}3B_2 & \Rightarrow (0, -\sqrt{\frac{2}{3}})
\end{array}$$

(What are the angles between these three projections?)

The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.

3. Back to time series with zero averages.

The dot product of $(1,1,-1,-1)$ with itself is 4.

(Actually, let's divide the result by the length of the series: so 1.)

The dot product of $(1,1,-1,-1)$ with $(-1,1,1,-1)$ is 0.

But the second is just the first *shifted*, so if we are looking for *similarity* of time series, we would like a shifted time series to match itself.

So try $(\sim,1,1,-1)\cdot(-1,1,1,-1)=\frac{3}{4}$.

Just in case, let's look at

$$\begin{aligned}(\sim,\sim,1,-1)\cdot(-1,1,1,-1) &= 0\\ (\sim,\sim,\sim,1)\cdot(-1,1,1,-1) &= -\frac{1}{4}\end{aligned}$$

and at

$$\begin{aligned}(1,-1,-1,\sim)\cdot(-1,1,1,-1) &= -\frac{3}{4}\\ (-1,-1,\sim,\sim)\cdot(-1,1,1,-1) &= 0\\ (-1,\sim,\sim,\sim)\cdot(-1,1,1,-1) &= \frac{1}{4}\end{aligned}$$

This is called the *cross-covariance*, CCV, (after dividing by 4, the series length)

lag	-3	-2	-1	0	1	2	3
CCV	$-.25$	0	.75	0	$-.75$	0	.25

4. A first Aldat implementation of the cross-covariance uses

query(*j*	*q*)	*cand*(*k*	*c*)
1	1	1	-1
2	1	2	1
3	-1	3	1
4	-1	4	-1

and is

> **for** *lag* <− − 3 : 3
>   **let** *k* **be fun succ**(*lag*) **of** *j* **order** *j*;
>   **let** *qc* **be red** + **of** $q \times c$;
>   *CCV* <+ [*lag*,*qc*] **in** (*cand* **ijoin** [*k*, *q*] **in** *query*;
> **end**

(Note. This doesn't give the results above: **succ** is cyclic. How must we modify the code? If we use the code as is, what will the result be?)

But this uses a loop, which is not necessary since all the calculations may be done in any order. To exploit Aldat parallelism, we must get away from this loop.

It really is a matrix multiplication.

$$\begin{pmatrix} \sim & \sim & \sim & 1 \\ \sim & \sim & 1 & 1 \\ \sim & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & \sim \\ -1 & -1 & \sim & \sim \\ -1 & \sim & \sim & \sim \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 3 \\ 0 \\ -3 \\ 0 \\ 1 \end{pmatrix}$$

(How does this change if we allow cycles?)

We can build this matrix in Aldat using

lag(*j*)	*query*(*h*	*q*)
−3	1	1
−2	2	1
−1	3	−1
0	4	−1
1		
2		
3		

> **let** *hm* **be red max of** *h*;
> **let** *k* **be** $h - j$;
> *qmat* <−[*j*, *k*, *q*,*hm*] **where** $0 < k$ **and** $k \leq hm$ **in** (*lag* **ijoin** [*h*, *q*,*hm*] **in** *query*);

(How must this change to allow cycles?)

And here's the matrix multiplication.

> **let** *qc* **be** (**equiv** + **of** $q \times c$ **by** *j*)/*hm*;
> [*j*, *qc*] **in** *qmat* **ijoin** *cand*;

Now there is no loop and Aldat can do everything in parallel.

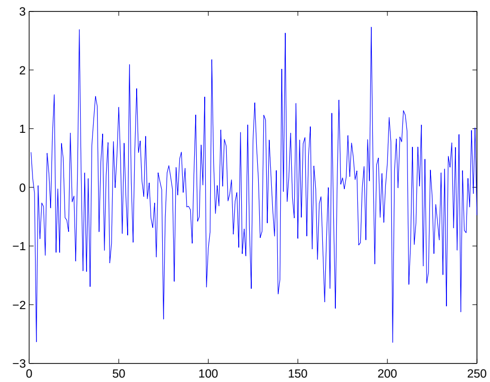
Is there a way to avoid both sequential processing and storing $2n - 1$ copies of the time series?

(Note [Per94] calls this covariance function "biased" because we've divided by *hm* for every row, even though only one of the sums has all *hm* terms. For his "unbiased" version, replace *hm* by *hm* −abs(*j*) in the denominator—why?)

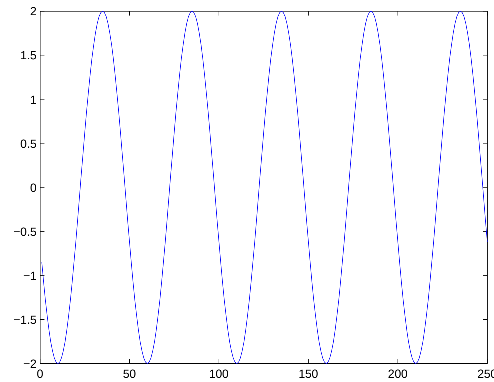
The definition of cross-covariance is given in, for example, [SS06, p.31].

5. Let's use the cross-covariance to pull a signal out of noise.

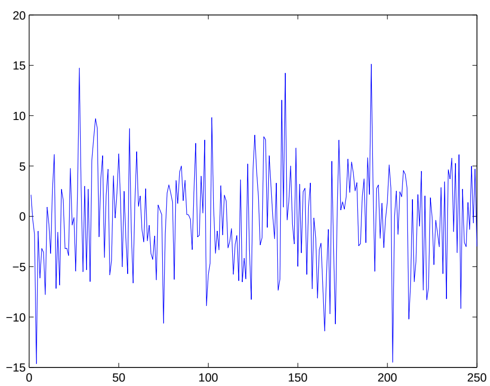
Here is Gaussian white noise.



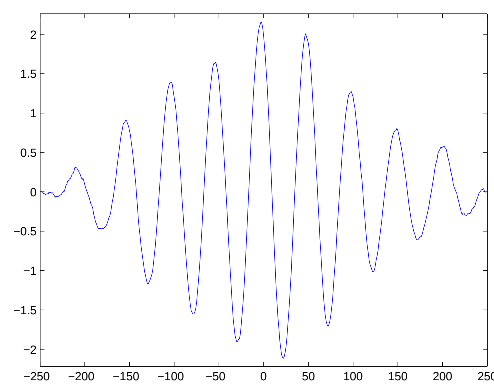
Here is a signal $(2\times\cos(2\times\pi\times x/50+0.6\times\pi))$.



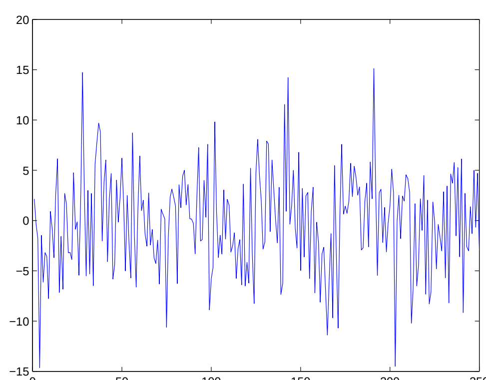
Here is a noisy signal (cos + 5×white).



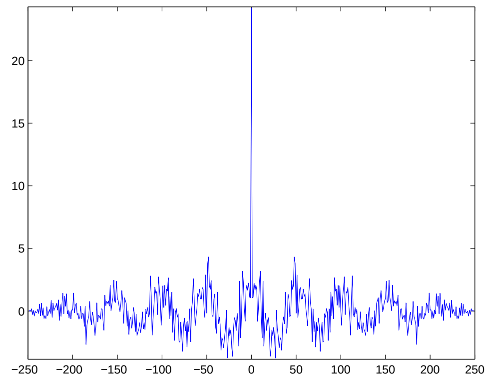
And here is the cross-covariance of the noisy signal and the true signal.



How do we know to try a periodic signal when we are given only the noisy signal?



We take the *auto-covariance* of the noisy signal, i.e., the cross-covariance of it with itself.



Now we can define ([SS06, p.31]) the *cross-correlation*, which is the cross-covariance normalized. If $CCV(S1,S2)$ is the cross-covariance of two signals (time series), *S1* and *S2*, then

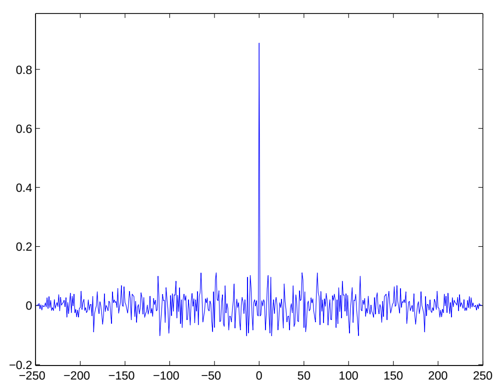
- the auto-covariance of one signal, *S1*, is $CCV(S1,S1)$
- the cross-correlation of *S1* and *S2* is

$$CCF(S1, S2) = CCV(S1, S2)/\sqrt{CCV(S1, S1)[0] \times CCV(S2, S2)[0]}$$

  where the [0] notation means that only the central element, corresponding to $lag = 0$, is used from the series representing the autocorrelations.
- the auto-correlation of *S1* is $CCF(S1,S1)$.

6. (Digression on the statistics of white noise.)

Here is the autocorrelation of the white noise, above.



White noise can be drawn from any statistical distribution. We used a Gaussian (normal) distribution above, but let's work with finite distributions.

a)

values	1	2	3
probs.	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
mean 0	$-\frac{5}{4}$	$-\frac{1}{4}$	$\frac{3}{4}$

Two *random variables*, a_1, a_2 drawn independently from this distribution.

$p_{a_1 a_2} \times 16$

	$a_2 \times 4$	-5	-1	3
a_1	-5	1	1	2
$\times$	-1	1	1	2
4	3	2	2	4

$a_1 \times a_2 \times 16$

	$a_2 \times 4$	-5	-1	3
a_1	-5	25	5	-15
$\times$	-1	5	1	-3
4	3	-15	-3	9

$p_{a_1 a_2} \times a_1 \times a_2 \times 16^2$

	$a_2 \times 4$	-5	-1	3	
a_1	-5	25	5	-30	$\to 0$
$\times$	-1	5	1	-6	$\to 0$
4	3	-30	-6	36	$\to 0$

The covariance is the sum of this last matrix and is zero.

Extreme correlation: $a_2 = a_1$.

$p_{a_1 a_2} \times 6$

	$a_2 \times 4$	-5	-1	3
a_1	-5	1	0	0
$\times$	-1	0	1	0
4	3	0	0	4

$a_1 \times a_2 \times 16$

	$a_2 \times 4$	-5	-1	3
a_1	-5	25	5	-15
$\times$	-1	5	1	-3
4	3	-15	-3	9

$p_{a_1 a_2} \times a_1 \times a_2 \times 6 \times 16$

	$a_2 \times 4$	-5	-1	3	
a_1	-5	25	0	0	$\to 0$
$\times$	-1	0	1	0	$\to 0$
4	3	0	0	36	$\to 0$

The covariance is (25+1+36)/6×16 $= 0.6548$

b)

values	4	5	6	7
probs.	0.1	0.2	0.3	0.4
mean 0	-2	-1	0	1

Two random variables, a_1, b_2 drawn independently from this and the previous distribution.

$p_{a_1 b_2}$

	b_2	-2	-1	0	1
a_1	-5	.1	.1	.05	0
$\times$	-1	0	.1	.05	.1
4	3	0	0	.2	.3

$a_1 \times b_2 \times 16$

	b_2	-2	-1	0	1
a_1	-5	2.5	1.25	0	-1.25
$\times$	-1	.5	.25	0	$-,25$
4	3	-1.5	$-.75$	0	.75

$p_{a_1 b_2} \times a_1 \times b_2 \times 16^2$

	b_2	-2	-1	0	1
a_1	-5	.25	.125	0	0
$\times$	-1	0	.025	0	$-.025$
4	3	0	0	0	.225

The covariance is the sum of this last matrix and is 0.6.

Let's consider a time series of random variables: a_1, b_1, a_2, b_2.

Here is the (auto) *covariance matrix* of this time series (with itself).

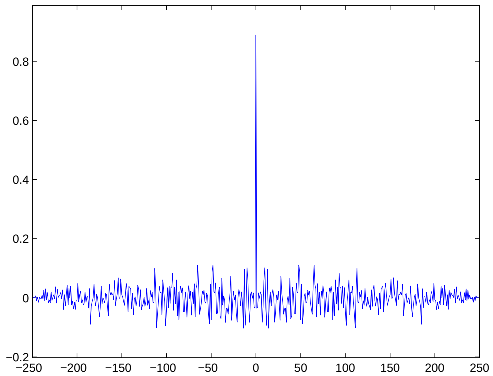
γ_{ab}	t	a_1	b_1	a_2	b_2
s	a_1	.6458	.6	0	.6
	b_1	.6	1	.6	0
	a_2	0	.6	.6458	.6
	b_2	.6	0	.6	1

(You should check that the auto-covariance of b is 1.)

White noise is any time series of random variables, $w_1, w_2, ..,$ whose autocovariance matrix is a multiple of the unit matrix, e.g.

γ_w	t	w_1	w_2	w_3	w_4	...
s	w_1	.6458	0	0	0	...
	w_2	0	.6458	.6	0	...
	w_3	0	0	.6458	0	...
	w_2	0	0	0	.6458	...
:	:	:	:	:		

Here is the autocorrelation of the white noise from Note 5.



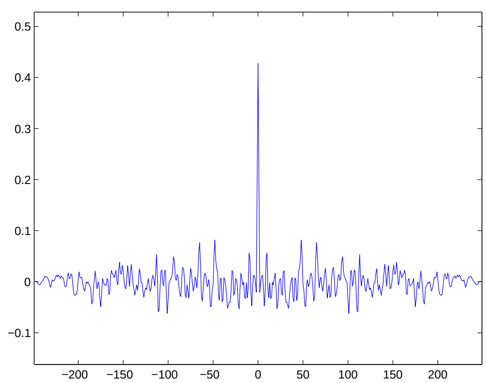
The central spur corresponds to the diagonal elements of the autocovariance matrix. Note that the values are not zero everywhere else, but if we averaged over many instances of this time series all but the central peak would vanish.

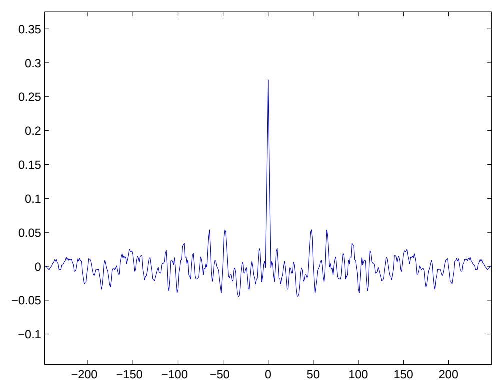
The central spur doubles in width (figure below, left) if we plot a two-element moving average,

$$w2_t = (w_{t-1} + w_t)/2$$

and trebles in width (figure below, right) if we plot a three-element moving average,

$$w3_t = (w_{t-2} + w_{t-1} + w_t)/3$$





These correspond to tridiagonal and 5-diagonal covariance matrices, respectively. For example, [SS06, p.21] uses

$$\gamma_w(s,t) = E(w_s w_t) = \sigma_w^2 \delta_{s,t}$$

to show

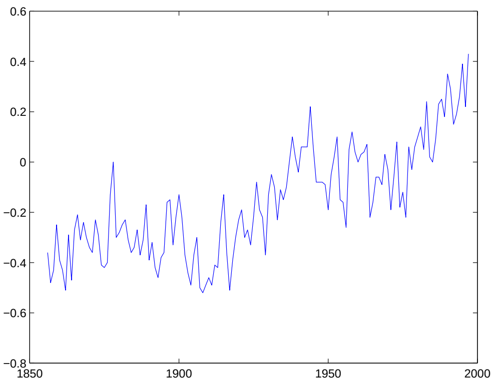
$$\gamma_{w3}(s,t) = \begin{cases} 3/9 & s = t \\ 2/9 & \mid s-t \mid = 1 \\ 1/9 & \mid s-t \mid = 3 \\ 0 & \mid s-t \mid \geq 3 \end{cases}$$

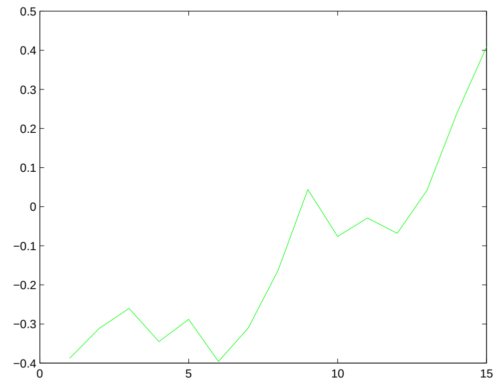
(In order to reserve the terms "auto/cross-covariance/correlation" for these theoretical constructs, [SS06, p.31] use "*sample* auto/cross-covariance/correlation" for what we have been calling "auto/cross-covariance/correlation".)

7. Scaling. In Note 1, we "translated" time series by subtracting their average values from each term. This is called "shifting". We may also want to "scale" a time series, e.g., $5, 5, -5, -5$, before comparing with another, e.g., $1, 1, -1, -1$. We do this by dividing by its standard deviation.

The *Pearson Correlation Coefficient* between two time series *S1* and *S2* is $CCF(\hat{S1}, \hat{S2})[0]$ where $\hat{S}$ is S shifted and scaled. This is inversely related to the Euclidean distance between the shifted and scaled time series, $1 - D^2(\hat{S1}, \hat{S2})/(2n)$, to be precise [SZ04, p.92].

We can also do time scaling. Here are global relative temperature plots, in Celsius degrees, the first by year (from www.stat.pitt.edu/stoffer/tsa2/globtemp.dat, a data file for [SS06]), the second by decade.





They should be comparable, but we must change timescales. Either we *downsample* the annual data

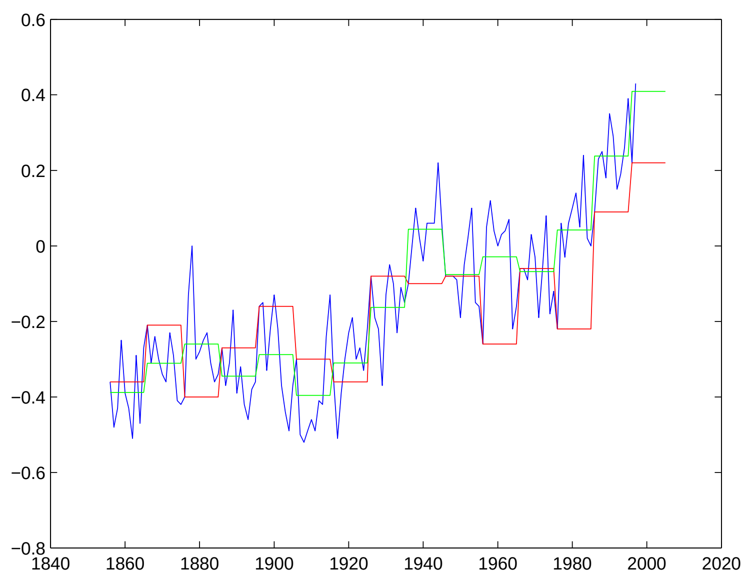
```
for k = 1:ceil(sizeX(1)/w), Y(k) = X(1+(k-1)*w); end
```

where w is the "window" size, or we *upsample* the decade data

```
for k = 1:sizeX(1)*w, Y(k) = X(1+floor((k-1)/w)); end
```

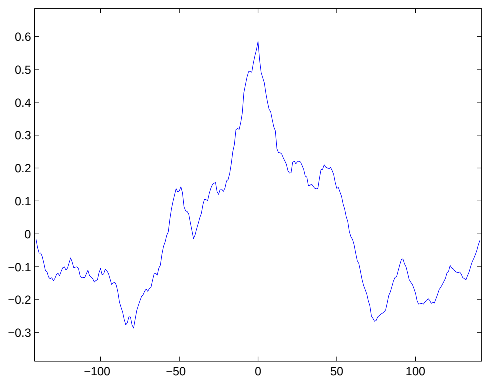
(See [SZ04, p.94].)

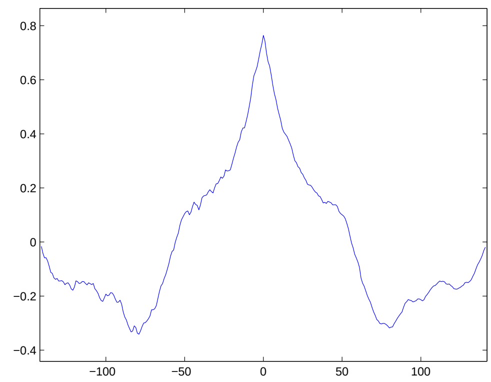
Here is the annual data again, with the downsampled data upsampled again and shown in red.



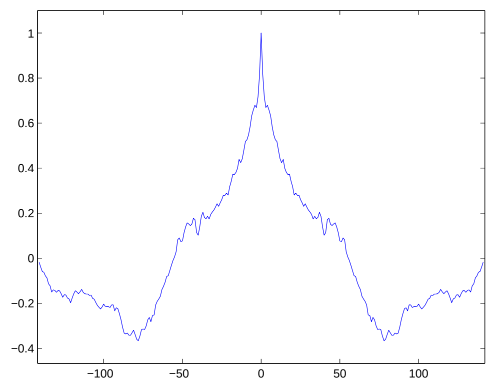
(This plot also shows, in green, the upsampled result of downsampling by average value of each group, not just the first value.)

Here are the cross-correlations of the annual data with the up-sampled decade date, using first values (left plot) and average values (right plot).





For comparison, here is the autocorrelation of the annual data (with itself).



An improvement on timescaling by integer factors is *time warping*: for two series of lengths m and n, respectively, upsample both to the least common multiple, $lcm(m, n)$. Then the distance squared between the series is the sum of the squares of the difference between associated terms. In the special case that $lcm(m, n) = mn$, this is

$$\sum_{l=1}^{mn-1} (x_{\lfloor k/m \rfloor} - y_{\lfloor k/n \rfloor})^2/mn$$

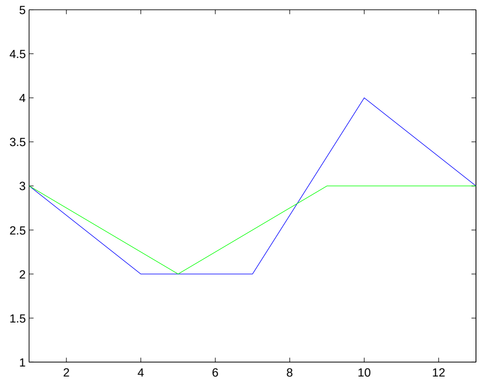
which is the same as the distance squared between the upscaled series/mn.

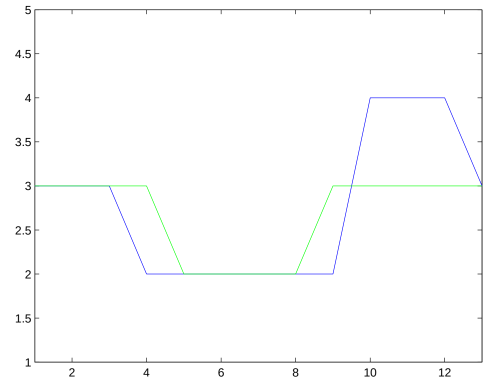
For example, consider the series *Y1* and *Y2*.

X	1	2	3	4	5	6	7	8	9	10	11	12	13
Y1	3			2			2			4			3
Y2	3				2				3				3
Y	3	4	3	2	2	1	2	2	3	4	5	4	3

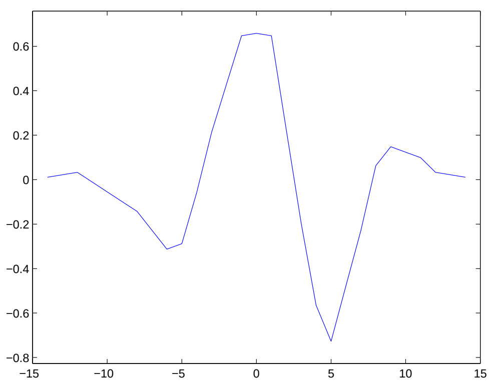
(Y is a series from which *Y1* and *Y2* might have been sampled: they are consistent with each other.)

Here are the data (left figure) and the results of upsampling (right figure).





The cross-correlation exceeds 0.6 at lags $-1, 0$ and 1.



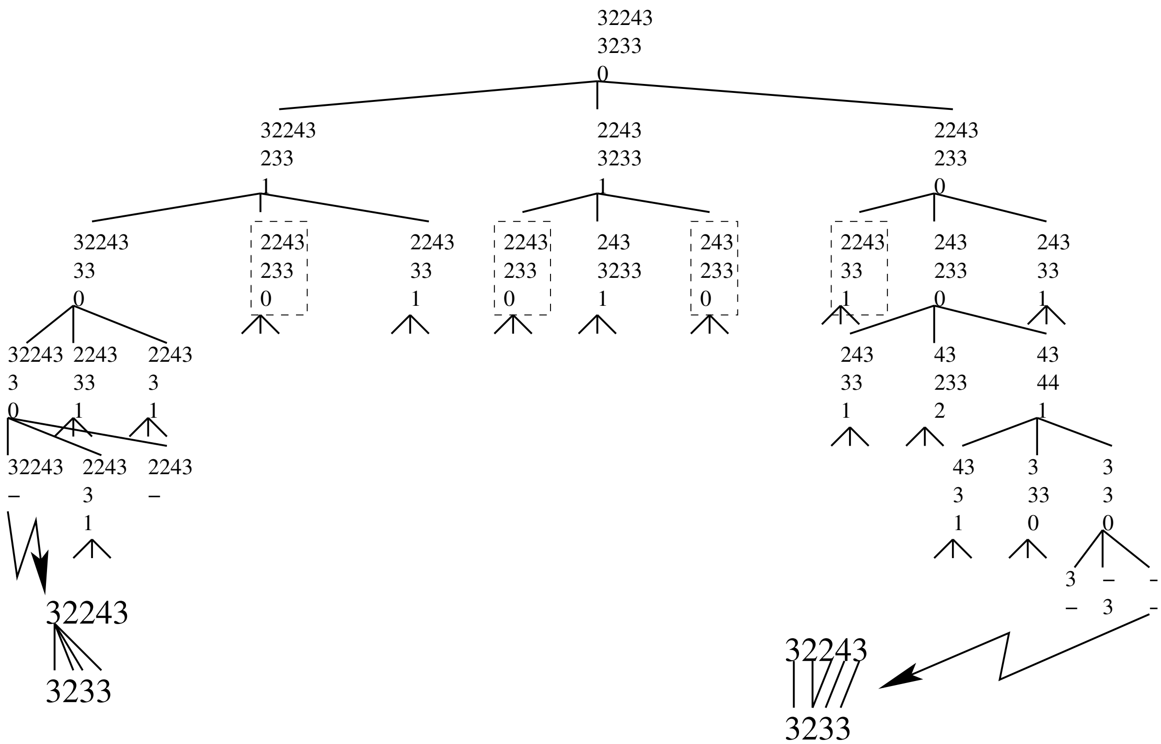
This is uniform time warping. *Dynamic* time warping stretches and squeezes the time axis to minimize the distance between the two series,

The distance squared between series *Y1* and *Y2* is found from the rule

$$D^2_{DTW}(Y1,\, Y2) = D^2(\mathbf{first}\, Y1, \mathbf{first}\, Y2) + \mathbf{min} \begin{cases} D^2_{DTW}(Y1, \mathbf{rest}\, Y2) \\ D^2_{DTW}(\mathbf{rest}\, Y1,\, Y2) \\ D^2_{DTW}(\mathbf{rest}\, Y1, \mathbf{rest}\, Y2) \end{cases}$$

where $D^2(\mathbf{first}\, Y1, \mathbf{first}\, Y2) = (\mathbf{first}\, Y1 - \mathbf{first}\, Y2)^2$.

This produces a tree of comparisons. The minimum distance seems to be 1.



This is done with dynamic programming in $\mathcal{O}(mn)$ time. Since that is too expensive for long series, *local* dynamic time warping does the dynamic programming within small neighbourhoods of each point on the time axis (after globally stretching both series to the same length). Time warping is partially discussed in [SZ04, pp.96–9].

References

[Per94] Matthew Perry. Different discrete autocorrelation functions produce strikingly different results. *Personal Engineering & Instrumentation News*, pages 75–80, Jan. 1994. URL=archive.chipcenter.com/dsp/DSP000524F1.html.

[SS06] Robert H Shumway and David S Stoffer. *Time Series Analysis and Its Applications With R Examples*. Springer Science+Business Media, LLC, New York, 2006.

[SZ04] Dennis Shasha and Yunyue Zhu. *High Performance Discovery in Time Series: Techniques and Case Studies*. Springer Verlag, New York, 2004.