XC

The procedure XC computes the digital cross-correlation of two (equal-length) one-dimensional arrays, and determines the maximum of the cross-correlation function by fitting a low-order polynomial or a Gaussian to the central peak. The error is estimated from the covariance matrix of the standard errors in the fitting parameters. The code works in pixel or in Fourier space. Upon request, a plot is shown to illustrate the least-squares fittings.

Syntax

XC,y1,y2,s1,s2,delta,e_delta[,nrange=variable][,npoints=variable] [,order=variable]

[,/plot][,/gauss][,/fourier]

Return Value

XC returns the location of the cross-correlation maximum and an estimate of its uncertainty. Both in units of pixels.

Arguments

y1 - (float array) 1st array

y2 - (float array) 2nd array

s1 - (float array) uncertainties for 1st array

s2 - (float array) uncertainties for 2nd array

Keywords

- nrange - (integer) half range of the shifts in the cross-correlation integral
- npoints- (integer) number of pixels around the highest-valued pixel to enter the fit. (default: 7)
- order - (integer) order of the polynomial: 2 or 3 (default: 2 = 2nd order) A gaussian can also specified setting order<0 (or by using the keyword 'gauss'.

- plot - produces a plot illustrating the model fitting
- gauss - use a Gaussian instead of a polynomial to model the peak of the cross-correlation function
- fourier - compute the cross-correlation in Fourier space (nrange is n_elements(y1)/2-1 regardles of input nrange)
- extras - extra plotting keywords can be used and will be passed along to plot

Discussion

The cross-correlation of two arrays (o spectra) **T** and **S** of equal and even number of elements N is defined as a new array **C** with

$$C_i = \sum_{k=1}^{N} T_k S_{k+i-\frac{N}{2}}, \tag{1}$$

where i runs from 1 to N. If the spectrum **T** is identical to **S**, but shifted by an an integer number of pixels p, the maximum value in the array **C** will correspond to its element $i = p + \frac{N}{2}$. Cross-correlation can be similarly used to measure shifts that correspond to non-integer numbers. In this case, finding the location of the maximum value of the cross-correlation function can be performed with a vast choice of algorithms.

Another decision that needs to be made is related to the pixels that are used in computing C_i. If the arrays **T** and **S** have equal size, the number of elements available for the calculation in Eq. 1 will be reduced as the index departs from $i = \frac{N}{2}$; the indices will exceed the array boundaries. It is convenient then to keep the number of pixels constant and equal to the maximum, N, for all values of i, by filling the missing values of the second array (**S**) with those in the other extreme of the array. Thus, we will use the entire array **T** as originally arranged, but

$$\begin{array}{rcll} S_i & = & S_{N-i}, & \text{for } i < 1, \quad \text{and} \\ S_i & = & S_{i-N}, & \text{for } i > N. \end{array} \tag{2}$$

This is equivalent to assume that the input array **S** is a discrete version of a periodic function which repeats itself every N pixels. With this choice, Eq. 1 is analogous to the discrete correlation of two periodic functions (Brigham 1974).

The discrete cross-correlation of two periodic functions T and S, a third periodic function C, transforms into a product in Fourier space

$$\mathcal{F}[C] = \mathcal{F}(T)^* \mathcal{F}(S), \tag{3}$$

where the superscript $*$ indicates the complex conjugate. Naturally, the same expression holds for our arrays **C**, **T**, and **S**. While the calculation of C_i for all values

of i using Eq. 1 requires N^2 multiplications and additions, working in Fourier space may speed up the calculation dramatically, as fast Fourier transforms can be calculated with a number of operations proportional to $N \log_2 N$.

This discussion of the cross-correlation function is based in pixel space (x), but everything applies directly to velocity space (v) by sampling the spectra with uniform steps in $\delta\lambda/\lambda$, or $\log\lambda$. Our case of interest is when the spectra correspond to different observations of the same object or very similar objects. Intrinsic differences between 'template' and 'object' are therefore negligible and random errors are expected to dominate. The errors in the cross-correlation function (see Eq. 1), assuming errors in the input spectra are uncorrelated, can be written (e.g., Murdoch & Hearnshaw 1991, Statler 1995)

$$\sigma_i^2 \equiv \sigma^2(C_i) = \sum_k T_k^2 \sigma^2(S_{i+k}) + S_{i+k}^2 \sigma^2(T_k). \tag{4}$$

Considering correlations will add two more terms to the right-hand side of Eq. 4 involving the covariances among the fluxes' uncertainties: a term

$$2 \sum_k \sum_{j=k+1} T_k T_j \sigma(S_{i+k}, S_{i+j}) \quad + \quad S_{i+k} S_{i+j} \sigma(T_k, T_j), \tag{5}$$

that accounts for correlations across frequencies in any given spectrum, and a term

$$2 \sum_k T_k S_{i+k} \sigma(T_k, S_{i+k}), \tag{6}$$

that describes correlations across spectra. In realistic situations, the finite width of the point-spread function will make non-zero the correlations between neighboring frequencies; the correlation matrix will be block diagonal. Correlations at any given frequency across spectra are also likely (e.g. a bad CCD column), but note that the correlation matrix is then strictly diagonal, and therefore only C_0 will be affected. XC neglects correlations across frequencies or spectra, using Eq. 4.

We use a simple model to determine the maximum of the cross-correlation function. The parameters of the model are determined by χ^2-square fitting to N_f data points near the maximum (see, e.g., Press et al. 1986), and the covariance matrix of the standard errors of the parameters **U**=[U_{ij}] is estimated by inverting the curvature matrix

$$[\mathrm{U}_{ij}] = [\alpha_{ij}]^{-1}, \tag{7}$$

which is approximated

$$\alpha_{ij} \simeq \sum_{k=1}^{Nf} \frac{1}{\sigma_k^2} \frac{\partial P(x_k)}{\partial a_i} \frac{\partial P(x_k)}{\partial a_j}. \tag{8}$$

Gaussian

If we model the cross-correlation peak with a Gaussian profile

$$P(x) = a_1 + a_2 \exp(-z^2/2) \tag{9}$$

where

$$z = \frac{x - a_3}{a_4}, \tag{10}$$

the derivatives are

$$\begin{aligned} \frac{\partial P(x_k)}{\partial a_1} &= 1 \\ \frac{\partial P(x_k)}{\partial a_i} &= \left(\frac{a_2}{a_4}\right)^{\delta_{3i}+\delta_{4i}} z_k^{i-2} \exp(-z_k^2/2) \qquad i = 2,3,4 \end{aligned} \tag{11}$$

where δ_{ij} is the Kronecker delta. These can be used in Eq. 8 to compute the curvature matrix, and finally (Eq. 7) the covariance matrix [U_{ij}]. The center of the Gaussian is at $x_{\max} = a_3$, and the variance is $\sigma^2(x_{\max}) = \mathrm{U}_{33}$.

Second order polynomial

If we model the cross-correlation peak with a polynomial

$$P(x) = \sum_{k=1}^{m+1} a_k x^{k-1} \tag{12}$$

of order $m = 2$, a maximum will exist when $a_3 < 0$ and its location is where the first derivative $P'(x)$ cancels out

$$x_{\max} = -\frac{a_2}{2a_3}, \tag{13}$$

The curvature matrix has in this case a simple expression

$$\alpha_{ij} \simeq \sum_k \frac{x_k^{i+j-2}}{\sigma_k^2}. \tag{14}$$

and the uncertainty in the maximum location can be calculated by linear error propagation

$$\sigma^2(x_{\max}) = \sum_{k=1}^{m+1} \left(\frac{\partial x_{\max}}{\partial a_k}\right)^2 \mathrm{U}_{kk} + 2 \sum_{k=1}^{m} \sum_{j=k+1}^{m+1} \left(\frac{\partial x_{\max}}{\partial a_k}\right) \left(\frac{\partial x_{\max}}{\partial a_j}\right) \mathrm{U}_{kj} \tag{15}$$

$$= \frac{1}{4a_3^2} \left(\mathrm{U}_{22} + \frac{a_2^2}{a_3^2}\mathrm{U}_{33}\right) - \frac{a_2}{2a_3^3}\mathrm{U}_{23}. \tag{16}$$

Third order polynomial

This corresponds to Eq. 12 for $m = 3$. The two roots of $P'(x)$ are

$$x = \frac{-a_3 \pm \beta}{3a_4} \tag{17}$$

where $\beta = \sqrt{a_3^2 - 3a_2a_4}$. To avoid large roundoff errors when $a_2a_4 \ll a_3^2$, we can define

$$\gamma = 1 + \frac{\beta}{|a_3|}, \tag{18}$$

and rewrite the solutions (Press et al. 1998)

$$x_1 = -\frac{a_3\gamma}{3a_4}, \qquad x_2 = -\frac{a_2}{a_3\gamma}. \tag{19}$$

Once we have identified which one we are interested in, the uncertainty can again be derived by substituting in Eq. 15 the non-zero terms

$$\begin{aligned}
\frac{\partial x_1}{\partial a_2} &= \frac{a_3}{2\beta|a_3|} & ;\frac{\partial x_2}{\partial a_2} &= -\frac{1}{a_3\gamma}\left(1 + \frac{3a_2a_4}{2\beta|a_3|\gamma}\right) \\
\frac{\partial x_1}{\partial a_3} &= -\frac{a_3^2\gamma}{3a_4\beta|a_3|} & ;\frac{\partial x_2}{\partial a_3} &= \frac{a_2}{\beta|a_3|\gamma} \\
\frac{\partial x_1}{\partial a_4} &= \frac{a_2a_3}{2a_4\beta|a_3|} + \frac{a_3\gamma}{3a_4^2} & ;\frac{\partial x_2}{\partial a_4} &= -\frac{3a_2^2}{2a_3\beta|a_3|\gamma^2}.
\end{aligned} \tag{20}$$

References

Brigham, E. O. 1974, The Fast Fourier Transform and Applications. Englewood Cliffs, NJ: Prentice Hall

Murdoch, K., & Hearnshaw, J. B. 1991, Astrophysics and Space Science, 186, 137

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. 1986, Numerical Recipes, Cambridge: Cambridge University Press

Statler, T. 1995, AJ, 109, 1371

Example

Cross-correlation of two gaussians of the same width ($\sigma = 7$ pixels), but shifted by 20 pixels, and measured with a signal-to-noise of 100. The default is to use a 2nd-order polynomial to fit 7 data points around the maximum of the cross-correlation peak:

```
IDL>  x=findgen(100)
IDL>  y=exp(-((x-50.)/sqrt(2.)/7.)^2)
IDL>  y2=exp(-((x-70.)/sqrt(2.)/7.)^2)
IDL>  xc,y,y2,y/100.,y2/100.,delta,edelta,/pl
IDL>  print,delta,edelta
      -20.000000           0.061816146
```

Version History

C. Allende Prieto, McDonald Observatory, initial version coded, November 2004

Improved error calculation added in October 2006

Gaussian model added in December 2006

See Also

Allende Prieto, C. 2007, AJ, in press (astro-ph/)