URL: http://cxc.harvard.edu/ciao3.4/uncertainty.html
Last modified: December 2006

AHELP for CIAO 3.4

uncertainty

Context: sherpa

Jump to: Description Examples CHANGES IN CIAO 3.2 Bugs See Also

Synopsis

Estimates confidence intervals for selected thawed parameters.

Syntax

```
sherpa> UNCERTAINTY [<dataset range> | ALLSETS] [ <arg_1> , ... ]

where <dataset range> = #, or more generally #:#,#:#,..., such that #
specifies a dataset number, and #:# represents an inclusive range of
datasets; one may specify multiple inclusive ranges by separating them
with commas. The default is to estimate limits using data from all
appropriate datasets.
```

Description

The command−line arguments <arg_n> may be:

UNCERTAINTY Command Arguments

Argument	Description
<sherpa_modelname>.{<paramname> | <#>}	A specified model component parameter (e.g., GAUSS.pos).
<modelname>.{<paramname> | <#>}	A specified model component parameter (e.g., g.pos).

The user may configure UNCERTAINTY via the Sherpa state object structure unc. The current values of the fields of this structure may be displayed using the command print(sherpa.unc), or using the more verbose Sherpa/S−Lang module function list_unc().

The structure fields are:

unc Structure Fields

Field	Description
sigma	Specifies the number of sigma (i.e., the change in statistic).
eps	The tolerance for sigma, influencing the numerical accuracy of the errorbars. (Decreasing eps increases

	errorbar accuracy.)
remin	If in the course of computing the interval, a statistic value is found that is less than the previous best−fit by more than remin, a new fit will be started; after minimization, UNCERTAINTY will run to completion.

Field values may be set using directly, e.g.,

```
sherpa> sherpa.unc.sigma = 2.6
```

NOTE: strict checking of value inputs is not done, i.e., the user can errantly change arrays to scalars, etc. To restore the default settings of the structure at any time, use the Sherpa/S−Lang module function restore_unc().

Confidence interval bounds are determined for each parameter in turn. A given parameter's value is varied while the values of all the other nominally thawed parameters are held fixed to their best−fit values. This is a simplistic method of estimating confidence interval bounds that gives truly accurate results only in special cases, as explained below.

Because UNCERTAINTY estimates confidence intervals for each parameter independently, the relationship between UNCERTAINTY.sigma and the change in statistic value delta_S can be particularly simple: sigma = the square root of delta_S for statistics sampled from the chi−square distribution and for the Cash statistic, and is approximately equal to the square root of (2 * delta_S) for fits based on the general log−likelihood (log(L)).

Confidence Intervals for the uncertainty command

Confidence	sigma	delta_chi−square	delta_log(L)
68.3%	1.0	1.00	0.50
90.0%	1.6	2.71	1.36
95.5%	2.0	4.00	2.00
99.0%	2.6	6.63	3.32
99.7%	3.0	9.00	4.50

Caveats

An estimated confidence interval is accurate if and only if:

- no correlations exist between the varied parameter and any of the other nominally thawed parameters,
- the chi−square or log(L) surface in parameter space is approximately shaped like a multi−dimensional paraboloid, and
- the best−fit point is sufficiently far from parameter space boundaries.

One may determine if these conditions hold, for example, by plotting the fit statistic as a function of each parameter's values (the curve should approximate a parabola) and by examining contour plots of the fit statistics made by varying the values of two parameters at a time (the contours should be elliptical, with principal axes aligned along the parameter axes, and parameter space boundaries should be no closer than approximately 3−sigma from the best−fit point).

If the second and third conditions hold, but the first does not, then the confidence intervals may be determined by using either COVARIANCE and PROJECTION.

If none of the conditions hold, then the output from UNCERTAINTY is meaningless except that it would give an

idea of the scale of the confidence intervals. To accurately determine the confidence intervals, one would have to reparameterize the model, or use Monte Carlo simulations or Bayesian methods.

Example 1

List the current and default values of the unc structure, and restore the default values:

```
sherpa> sherpa.unc.sigma = 5
sherpa> list_unc()
Parameter   Current         Default                         Description
-------------------------------------------------------------------------
sigma             5               1                    Number of sigma
eps            0.01            0.01                Tolerance for sigma
remin          0.01            0.01   Thresh stat change to reminimize
sherpa> restore_unc()
sherpa> list_unc()
Parameter   Current         Default                         Description
-------------------------------------------------------------------------
sigma             1               1                    Number of sigma
eps            0.01            0.01                Tolerance for sigma
remin          0.01            0.01   Thresh stat change to reminimize
```

Example 2

Search parameter space to find a range of parameter values within a confidence interval of 90.0% for all thawed parameters:

```
sherpa> READ DATA example1.dat
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> SOURCE = POLYNOM1D[my]
sherpa> THAW my.c1 my.c2
sherpa> METHOD SIMPLEX
sherpa> FIT
...
sherpa> sherpa.unc.sigma = 1.6
sherpa> UNCERTAINTY

WARNING: found better fit -- reminimizing!
 smplx: v1.3
 smplx:  initial statistic value =     3.33768E-01
 smplx:     converged to minimum =     3.30123E-01 at iteration =     12
 smplx:    final statistic value =     3.30123E-01
            my.c0  -0.861519
            my.c1  0.672815
            my.c2  0.919114


Computed for uncertainty.sigma = 1.6
        ------------------------------------------------------
        Parameter Name      Best-Fit Lower Bound     Upper Bound
        ------------------------------------------------------
            my.c0          -0.861519  -2.32896        +2.45363
            my.c1           0.672815  -0.757506       +0.730418
            my.c2           0.919114  -0.133193       +0.131418
```

Example 3

Search parameter space to find a range of parameter values within a confidence interval of 68.3% for the parameters my.c0 and my.c1:

```
sherpa> sherpa.unc.sigma = 1.0
sherpa> UNCERTAINTY my.c0 my.c1

Computed for uncertainty.sigma = 1
        ------------------------------------------------------
        Parameter Name      Best−Fit Lower Bound     Upper Bound
        ------------------------------------------------------
            my.c0          −0.861519  −1.4335         +1.56432
            my.c1           0.672815  −0.479026       +0.450944
```

CHANGES IN CIAO 3.2

Prior to CIAO 3.2 the UNCERTAINTY command could not be used until the dataset had been fit. This was done to ensure that the parameter values were at their best−fit location, but caused problems when fitting multiple datasets or loading previously−saved analysis sessions. This restriction has now been removed. Please note that the results of UNCERTAINTY will not be valid unless the parameters are at their best−fit values.

Bugs

See the Sherpa bug pages online for an up−to−date listing of known bugs.

See Also

sherpa

berrors, bsyserrors, compute_errors, compute_statistic, covariance, errors, ftest, get_paramest, get_paramestint, get_paramestlim, get_paramestreg, goodness, interval−projection, interval−uncertainty, list_paramest, mlr, projection, region−projection, region−uncertainty, restore_paramest, run_paramest, run_paramestint, run_paramestlim, run_paramestreg, set_errors, set_syserrors, staterrors, syserrors

---

URL: http://cxc.harvard.edu/ciao3.4/uncertainty.html
Last modified: December 2006