compute_cstat_gof

November 30, 2020

How to compute cstat goodness of fit in CIAO/Sherpa

Vinay Kashyap (CXC/CfA)

Aneta Siemiginowska (CXC/CfA), Long Xi (CfA), Jelle Kaastra (SRON)

This is a walkthrough of a python script written by Long Xi, translated from an IDL script written by Vinay Kashyap, based on piecewise approximation of the nominal cstat prescribed by Kaastra (2017, A&A, 605, A51).

It computes the *expected* cstat and the *expected* variance in the cstat for the best-fit model, and compares it to the cstat computed for the observed data and the best-fit model. If the observed cstat is significantly higher than the estimated cstat, the model is likely a bad fit. If it is significantly smaller, it is an indication that the data are being *overfit*.

It requires the sherpa.astro.ui and scipy packages. The example uses an ACIS-S dataset of Quasar Q1127-145, ObsID 866 that has been previously used to demo pyBLoCXS.

This notebook, as well as the cstat_gof.py script, and all the associated data files used here, are available on the IACHEC website.

Initialize packages

[1]:
```
from sherpa.astro.ui import *
from glob import glob
import numpy as np
from scipy.special import factorial
```

Load in the cstat goodness-of-fit calculator script

This is available within the associated tar file as cstat/cstat_gof.py

[2]:
```
import cstat_gof
```

Load in example dataset

Set the energy range to be extremely large, because we want to demonstrate what happens with a *bad fit* first.

[3]:
```
load_data("obs866/acis.pi")
ignore()
notice(0.1,10)
```

```
read ARF file obs866/acis.arf
read RMF file obs866/acis.rmf
read background file obs866/acis_bg.pi
```

Define a model

Again, because we want to demonstrate a bad fit, let us pick a model that is clearly inappropriate for a Quasar spectrum. We will assume an optically thin thermal emission model with no absorption.

[4]:
```
set_model(xsapec.kT1)
```

Set the statistic to cstat

[5]:
```
set_stat('cstat')
```

Fit the model to the data

[6]:
```
fit()
```

```
WARNING: data set 1 has associated backgrounds, but they have not been
subtracted, nor have background models been set
Dataset               = 1
Method                = levmar
Statistic             = cstat
Initial fit statistic = 5.72125e+07
Final fit statistic   = 3273.77 at function evaluation 141
Data points           = 679
Degrees of freedom    = 677
Probability [Q-value] = 0
Reduced statistic     = 4.83571
Change in statistic   = 5.72092e+07
   kT1.kT         64           +/- 11.2441
   kT1.norm       0.00274832   +/- 0.000143993
WARNING: parameter value kT1.kT is at its maximum boundary 64.0
```

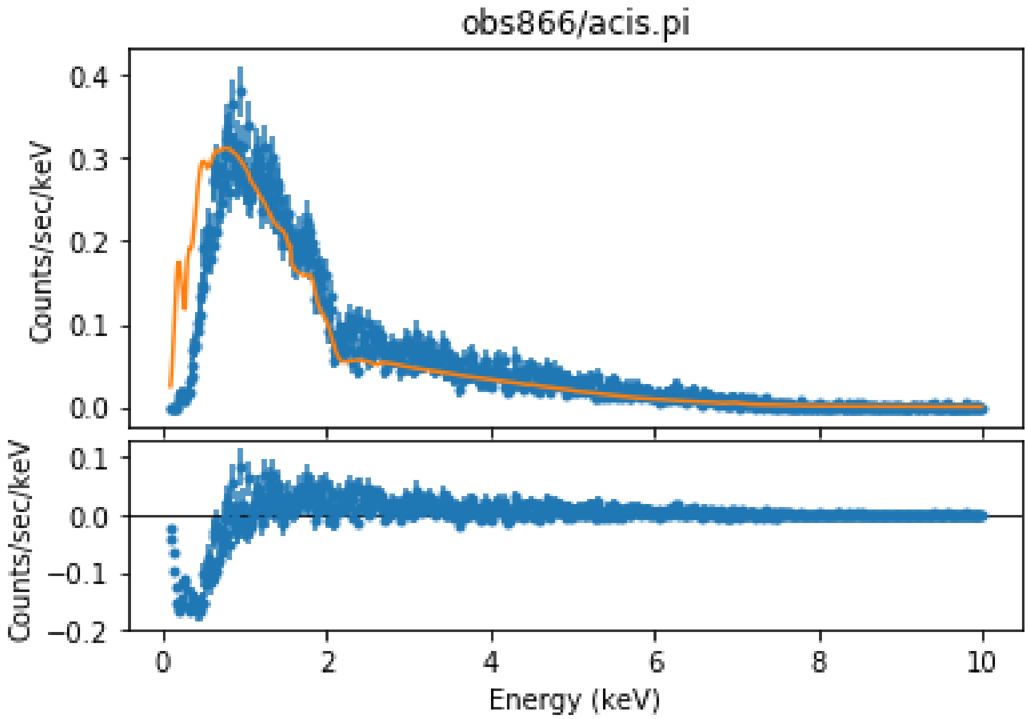
Show the fit

Notice that the fit is obviously bad, with large residuals at low energies that contributes to a large cstat value. If this were a chisq fit, we would say reduced chisq = 4.8

[7]:
```
plot_fit_resid()
calc_stat_info()
obscstat = calc_stat()
print("observed cstat = ",obscstat)
```

```
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: data set 1 has associated backgrounds, but they have not been
subtracted, nor have background models been set
Dataset               = 1
Statistic             = cstat
Fit statistic value   = 3273.77
Data points           = 679
Degrees of freedom    = 677
Probability [Q-value] = 0
Reduced statistic     = 4.83571
```

```
WARNING: data set 1 has associated backgrounds, but they have not been
subtracted, nor have background models been set
observed cstat =  3273.7730508449263
```



Compute expected cstat for best-fit model

Extract the energy array and the model values from the plot and call the **cstat_gof** script.

[8]:
```
x = get_data_plot().x
y = get_data_plot().y
[Ce,Cv] = cstat_gof.gof_cstat(x,y)
print("expected cstat = ",Ce," +- ",np.sqrt(Cv))
```

```
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
expected cstat =  713.980182016839  +-  37.46020112454112
```

Ce is the cstat value expected if the data were generated based on the model values (Eq 4 of Kaastra 2017).

Cv is the expected variance (Eq 6 of ibid).

They are computed using the approximations given in Sec 3, ibid).

Evaluate goodness of fit

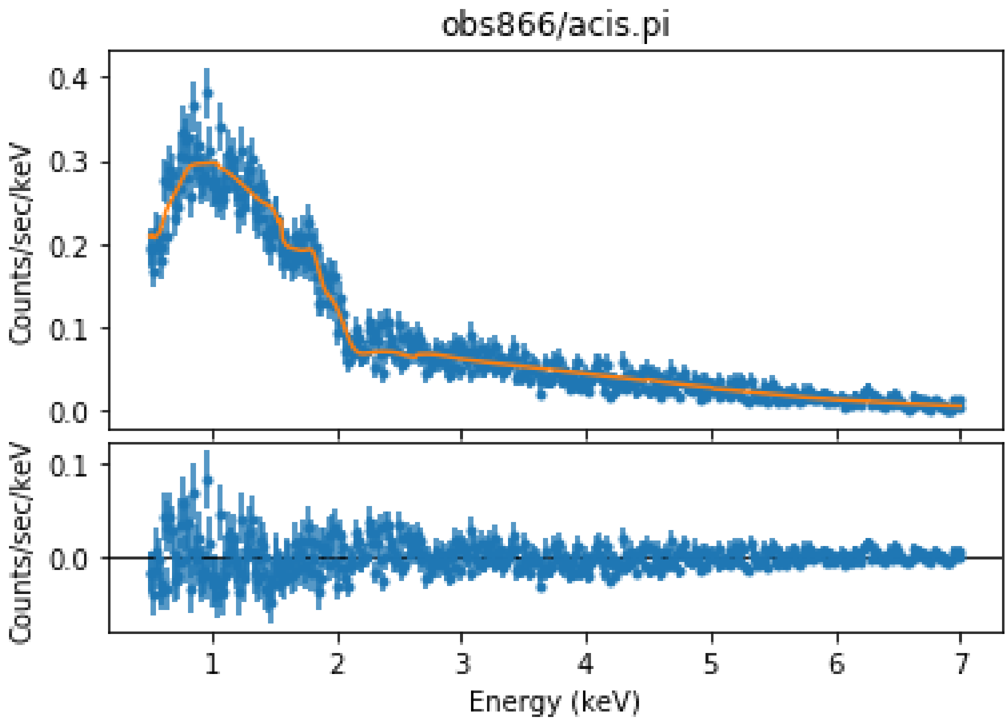
[9]:
```
print("how good is the fit?")
qual = (obscstat-Ce)/np.sqrt(Cv)
print("Observed cstat is off from expected by ",qual," sigma")
```

```
how good is the fit?
Observed cstat is off from expected by  68.33366591700178  sigma
```

Whew, that is a *very* bad fit.
Now let us see how a good fit behaves
We will use a shorter energy range and a more realistic model (an absorbed power-law, natch)

[10]:
```
ignore()
notice(0.5,7.0)
set_model(xstbabs.abs1*xspowerlaw.pl1)
fit()
plot_fit_resid()
obscstat = calc_stat()
print("observed cstat = ",obscstat)
```

```
WARNING: data set 1 has associated backgrounds, but they have not been
subtracted, nor have background models been set
Dataset               = 1
Method                = levmar
Statistic             = cstat
Initial fit statistic = 3.14697e+07
Final fit statistic   = 476.76 at function evaluation 32
Data points           = 446
Degrees of freedom    = 443
Probability [Q-value] = 0.129483
Reduced statistic     = 1.07621
Change in statistic   = 3.14692e+07
   abs1.nH        0.0707008    +/- 0.00771314
   pl1.PhoIndex   1.17661      +/- 0.0239493
   pl1.norm       0.000534254  +/- 1.35057e-05
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: data set 1 has associated backgrounds, but they have not been
subtracted, nor have background models been set
observed cstat =  476.75951254247997
```



That seems like a good fit, cstat is 477 with 443 degrees of freedom.
How deviant is the fit?

[11]:
```
x = get_data_plot().x
y = get_data_plot().y
[Ce,Cv] = cstat_gof.gof_cstat(x,y)
print("expected cstat = ",Ce," +- ",np.sqrt(Cv))
print("how good is the fit?")
qual = (obscstat-Ce)/np.sqrt(Cv)
print("Observed cstat is off from expected by ",qual," sigma")
```

```
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
WARNING: The displayed errorbars have been supplied with the data or calculated
using chi2xspecvar; the errors are not used in fits with cstat
expected cstat =  481.93243579262844  +-  30.819214681479007
how good is the fit?
Observed cstat is off from expected by  -0.16784734145926083  sigma
```

Observed cstat is off by about -0.2 sigma from the expected cstat. *That is definitely a good fit!*