ctools - Bug #1319

Gaussian spectral function

09/05/2014 02:33 AM - Katagiri Hideaki

Status:	Closed		Start date:	09/05/2014
Priority:	Normal		Due date:	
Assigned To:	Knödlseder Jürgen		% Done:	100%
Category:			Estimated time:	0.00 hour
Target version:				

Description

When I try to use Gaussian spectral function in ctools, I have a trouble.
The model is:

```
<source name="DMline" type="PointSource">
  <spectrum type="Gaussian">
 <parameter name="Normalization" scale="1" value="1.0"  min="1.e-100" max="1.e+100" free="1"/>
 <parameter name="Mean"         scale="1e6"   value="1.0" min="0.01" max="100.0"  free="0"/>
 <parameter name="Sigma"         scale="1"   value="1.0" min="0.01" max="100.0"  free="0"/>
  </spectrum>
    <spatialModel type="SkyDirFunction">
  <parameter name="RA"    scale="1.0" value="266.415000" min="-360" max="360" free="0"/>
  <parameter name="DEC"   scale="1.0" value="-29.006111" min="-90" max="90"  free="0"/>
  </spatialModel>
  </source>
```

And when I execute it,

[katagiri@heag01 14082804_dmline]$ python -i ./test.py

```
            like.execute()
            like.get_loglike()
```

-1658663.8397156834

```
            like.obs().models()['DMline']['Normalization'].value()
```

1.0

the "Normalization" would not change. And I forced it to change like:

```
            like.obs().models()['DMline']['Normalization'].value(1e-100)
            like.get_parameters()
            like.execute()
```

like.get_loglike()

```
like.get_loglike()
```

-1658663.8397156834

the likelihood value did not change. The same problem occurred when using BrokenPowerLaw function.
But this works by using other functions like PowerLaw.

I attached test_redmine.py script and the model file.

Regards,
Hide

History

#1 - 09/05/2014 03:41 PM - Knödlseder Jürgen

Could you also add the ctlike output file so that I can see what's going on? Thanks.

#2 - 09/08/2014 04:56 AM - Katagiri Hideaki

- File ctlike_redmine.log added

I added ctlike.log to this page.

#3 - 09/08/2014 08:45 AM - Knödlseder Jürgen

... I should have seen this earlier. All parameters in the model XML file have free="0", hence none of the parameters gets fitted. That's why you have
a curvature matrix with zero elements (see ctlike log file). I should add a test for this in ctlike so that a warning is given. I added an issue for this:
#1320.

Please set the parameters you want to fit to free="1" and retry.

#4 - 09/09/2014 02:36 AM - Katagiri Hideaki

- File deleted (model_onlydiffuse_test.xml)

#5 - 09/09/2014 02:36 AM - Katagiri Hideaki

- File deleted (ctlike_redmine.log)

#6 - 09/09/2014 02:42 AM - Katagiri Hideaki

- File model_onlydiffuse_test.xml added

- File ctlike_redmine.log added

#7 - 09/09/2014 02:47 AM - Katagiri Hideaki

Knödlseder Jürgen wrote:

> ... I should have seen this earlier. All parameters in the model XML file have free="0", hence none of the parameters gets fitted. That's why you
> have a curvature matrix with zero elements (see ctlike log file). I should add a test for this in ctlike so that a warning is given. I added an issue for
> this: #1320.
>
> Please set the parameters you want to fit to free="1" and retry.

As you can see in the test_redmine.py shell, the parameter "Normalization" is changed from "fix" to "free".
So probably the above result is treated as free=1.
In order to make sure of the result, I replaced free=1 in the model_onlydiffuse_test.xml and got ctlike_redmine.log.
I attached and replaced them the above. It seems that "Normalization" parameter is forced to fix.

Regards,
Hide

#8 - 09/09/2014 09:26 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 50

Okay, for some reason the gradient computation with respect to the normalization gives zero, which then fixes the parameter.

I recognized that the width of the line is 1 MeV, which is basically a Dirac for CTA. ctools/GammaLib evaluates the model at the bin centre, hence I guess that the line position is too far away from any bin centre of your event cube, and thus the response and gradient evaluation returns zero.

Note that energy dispersion is not yet fully implemented in GammaLib, hence I would recommend to set sigma to something like 10% of the line energy (or whatever the energy resolution is). I would expect that this should fix the problem.

You may also try unbinned instead of binned analysis as then you are not tied to a specific energy binning and evaluation at the bin centres. For a fine spectral analysis, unbinned is probably the preferred option (although once energy dispersion is implemented, there shouldn't be any problem, provided that the energy binning is fine enough).

#9 - 09/10/2014 02:26 AM - Katagiri Hideaki

- File ctlike_redmine2.log added

Hi Jurgen,

When I try the Gaussian width as 10% of the line energy, it works finally. I attached ctlike.log again.

I just additionally note that the analysis of 500hr simulation data seems to consume large memory size (>10GB?) and it seems to take a longer time than Fermi Science tools for some reason.

Regards,
Hide

Knödlseder Jürgen wrote:

> Okay, for some reason the gradient computation with respect to the normalization gives zero, which then fixes the parameter.
>
> I recognized that the width of the line is 1 MeV, which is basically a Dirac for CTA. ctools/GammaLib evaluates the model at the bin centre, hence I guess that the line position is too far away from any bin centre of your event cube, and thus the response and gradient evaluation returns zero.
>
> Note that energy dispersion is not yet fully implemented in GammaLib, hence I would recommend to set sigma to something like 10% of the line energy (or whatever the energy resolution is). I would expect that this should fix the problem.
>
> You may also try unbinned instead of binned analysis as then you are not tied to a specific energy binning and evaluation at the bin centres. For a fine spectral analysis, unbinned is probably the preferred option (although once energy dispersion is implemented, there shouldn't be any problem, provided that the energy binning is fine enough).

#10 - 09/10/2014 02:54 AM - Knödlseder Jürgen

- % Done changed from 50 to 60

Nice that it works now.

Do you know which step requires the large memory size? I see you simulate many events, and the memory handling of these events is actually not very effective (it will load all the data into memory and then bin the event cube). Practically this should not be a problem later as runs are relatively short (typically 30 min), hence such big junks should not be loaded.

Concerning the execution time, I recognize that the fit did not go very well. You consumed all 100 iterations and still the fit is not good. The observed number of events differs from the modeled number. The initial value of the line seems to be pretty off from a realistic value (see the high likelihood step delta in the first iteration), and then the normalization becomes negative and hits the lower boundary.

Maybe the problem is also related to the fact that you seem to fix all parameters, except for the Gaussian line. This is certainly not a good idea as the line has then to adjust for all statistical noise in the data. I would at least leave the background model normalization as a free parameter, and likelihood also that of the Galactic diffuse model (although this will be probably pretty degenerate with the background model).

#11 - 10/30/2014 12:14 PM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 60 to 100

Seems to work now, hence I close the issue.

Files

test_redmine.py	1.23 KB	09/05/2014	Katagiri Hideaki
model_onlydiffuse_test.xml	2.06 KB	09/09/2014	Katagiri Hideaki
ctlike_redmine.log	8.2 KB	09/09/2014	Katagiri Hideaki
ctlike_redmine2.log	30.1 KB	09/10/2014	Katagiri Hideaki