ctools - Bug #3542

cssens bugs

02/17/2021 05:19 PM - Sadeh Iftach

Status:	Closed	**Start date:**	02/17/2021
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0.0		

Description

Hello,

Here are a couple of issues I've run into with cssens

1. too-low values:

In https://github.com/ctools/ctools/blob/devel/cscripts/cssens.py#L387
I've run into errors of the kind:

```
          File "cssens.py", line 754, in run
            result = self._e_bin(ieng)
          File "cssens.py", line 693, in _e_bin
            result = self._get_sensitivity(emin, emax, self._models)
          File "cssens.py", line 401, in _get_sensitivity
            prefactor.value(crab_prefactor * test_crab_flux)
          File
```

```
            return _opt.GOptimizerPar_value(self, *args)
        ValueError: *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified value factor
 9.99999999999999e-11 is smaller than the minimum boundary 1e-10.
```

I suggest something like:

```
        min_pref = prefactor.min()
        val_now = crab_prefactor * test_crab_flux
        val_now = max(val_now, min_pref * 1.01)
        prefactor.value(val_now)
```

1. division by zero

In https://github.com/ctools/ctools/blob/devel/cscripts/cssens.py#L597
I've encountered division by zero errors.

I suggest something like:

```
    rxy_norm = (mean_xx - mean_x * mean_x) * (mean_yy - mean_y * mean_y)
    if rxy_norm < 1e-10:
       rxy_norm = 1
    else:
       rxy_norm = math.sqrt(rxy_norm)

    rxy = (mean_xy - mean_x * mean_y) / rxy_norm
```

History

#1 - 02/17/2021 06:25 PM - Sadeh Iftach

PS

These extremely low values for the prefactor correspond eventually to completely unrealistic sensitivities:

```
      loge   emin   emax crab_flux   photon_flux   energy_flux   sensitivity  regcoeff  nevents      npred
0 -0.200171  0.501  0.794   0.000454  1.212797e-14  1.204168e-14  2.609276e-14   0.23334     22.0  21.999989
```

where the calculation was done for a 10 second exposure!

This seems related to using a background model based on:

```
<source_library title="source library">
  <source name="CTABackgroundModel" type="CTAIrfBackground" instrument="CTA">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" value="1" error="0" scale="1" min="0.001" max="1000" free="1" />
      <parameter name="Index" value="0" error="0" scale="1" min="-5" max="0" free="1" />
      <parameter name="PivotEnergy" value="1" scale="1000000" min="0.01" max="1000" free="0" />
    </spectrum>
  </source>
  <source name="source_00047513" type="PointSource" tscalc="1">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" value="1" error="0" scale="1e-16" min="1e-10" max="10000000000" free="1" />
      <parameter name="Index" value="-2.5" scale="1" min="-5" max="0" free="0" />
      <parameter name="PivotEnergy" value="1" scale="1000000" min="0.001" max="1000" free="0" />
    </spectrum>
    <spatialModel type="PointSource">
      <parameter name="RA" value="265.97" scale="1" min="-360" max="360" free="0" />
      <parameter name="DEC" value="-29.38" scale="1" min="-90" max="90" free="0" />
    </spatialModel>
  </source>
  <source name="merged_mapcube_models" type="DiffuseSource">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" value="1" scale="1" min="1e-09" max="1000000000" free="1" />
      <parameter name="Index" value="0" scale="1" min="-5" max="0" free="1" />
      <parameter name="PivotEnergy" value="1" scale="1000000" min="0.1" max="10" free="0" />
    </spectrum>
    <spatialModel type="DiffuseMapCube" file="output/sense_0/source_00047513/source_00047513_pl_0_merged_mapcube_models.fits">
      <parameter name="Normalization" value="1" scale="1" min="1e-09" max="1000000000" free="0" />
    </spatialModel>
  </source>
</source_library>
```

and includes a DiffuseMapCube

I've tried using the same model, but not allowing the merged_mapcube_models background parameters to be fit. (Only CTABackgroundModel and the test source, source_00047513, have free parameters.) In this case, I get what looks like the expected answer, of ~10^-10 sensitivity for 10sec:

```
      loge   emin   emax crab_flux   photon_flux   energy_flux  \
0 -0.200171  0.501  0.794   1.812952  4.845686e-11  4.811212e-11
1 -0.000077  0.794  1.259   2.237868  3.028069e-11  4.765867e-11
2  0.199984  1.259  1.995   2.639513  1.803496e-11  4.499607e-11

   sensitivity  regcoeff  nevents      npred
0  1.042527e-10  0.970945     35.0  30.346305
1  1.031549e-10  0.946488     26.0  19.374530
2  9.753276e-11  0.978457     20.0  14.034580
```

So wither something is be going on with the DiffuseMapCube, or the inclusion of multiple background models on it's own results in too much complexity for the fit.

#2 - 05/07/2021 11:03 PM - Knödlseder Jürgen

- Target version set to 2.0.0

Iftach, if you want to propose some code to get merged in ctools please go ahead with coding and testing and making a pull request.

#3 - 05/28/2021 03:48 AM - Knödlseder Jürgen

As far as I can see, the corresponding code has now be integrated. Iftach, can you confirm that this issue is done?

#4 - 05/31/2021 05:11 PM - Sadeh Iftach

- Status changed from New to Resolved

- % Done changed from 0 to 100

#5 - 03/01/2022 05:41 PM - Knödlseder Jürgen

- Status changed from Resolved to Closed