GammaLib - Feature #2694

Implement more complex background models for H.E.S.S. data analysis

10/04/2018 11:13 AM - Knödlseder Jürgen

Status:	Closed	**Start date:**	10/04/2018
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.6.0		

Description

The analysis of the H.E.S.S. DR1 data indicates that more complex background models are needed with respect to those that are currently available. For example, a background model is needed that allows for spatial gradients over the field of view.

As a first step, a GCTAModelBackground class should be added that implements a factorized background model, equivalent to the GModelSky class. The spatial component of this model should be of type GCTAModelSpatial, another class that is needed. The GCTAModelRadial class should then derive from GCTAModelSpatial, and may be renamed GCTAModelSpatialRadial for consistency. A GCTAModelSpatialMultiplicative class should be added to allow for a multiplicative combination of models.

Related issues:

Related to GammaLib - Change request # 2695: Allow RoI centres different from...	**Closed**	**10/08/2018**

History

#1 - 10/04/2018 02:12 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- % Done changed from 0 to 20

I added the class GCTAModelBackground to implement the general factorized CTA background model. The class does not full work so far, in particular the npred() and mc() methods need to be tested.

I added the class GCTAModelSpatial that implements the spatial part of the GCTAModelBackground model. To register all possible spatial components, a GCTAModelSpatialRegistry class was also added.

The GCTAModelRadial class now derives from GCTAModelSpatial. Still, the GCTAModelRadial::eval() and GCTAModelRadial::mc() methods need to be implemented so that the derived classes can also be used from the GCTAModelBackground class. All derived classes now also register to GCTAModelSpatialRegistry.

Finally, a unit test was added that test a XML file that is managed by GCTAModelBackground. The model type is CTABackground, the XML file is shown below. The unit test does not yet cover the test of the spatial part, that still needs to be implemented.

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="My model" type="CTABackground" instrument="CTA">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor"   scale="1e-6" value="61.8" min="0.0"  max="1000.0" free="1"/>
      <parameter name="Index"       scale="-1"   value="1.85" min="0.0"  max="+5.0"   free="1"/>
      <parameter name="PivotEnergy" scale="1e6"  value="1.0"  min="0.01" max="1000.0" free="0"/>
    </spectrum>
    <spatialModel type="Gaussian">
      <parameter name="Sigma" scale="1.0" value="3.0" min="0.01" max="10.0" free="1"/>
    </spatialModel>
  </source>
</source_library>
```

#2 - 10/08/2018 12:33 PM - Knödlseder Jürgen

- % Done changed from 20 to 30

I turns out that the H.E.S.S. data do not contain DETX and DETY information. I therefore added some code to GCTAEventList that computes the DETX and DETY information from the event direction using the pointing information. Note that DETX and DETY are defined in sky coordinates.

I also verified that the new GCTAModelBackground class produces the same result as the formal radial acceptance model. Now the code is ready for implementing more complex background models.

Note that the GCTAModelSpatial::npred method assumes that the RoI centre is the pointing direction. The same is by the way true for the GCTAModelAeffBackground and GCTAModelIrfBackground methods. At least some code should be added to these methods that throws an exception if the RoI centre is different.

#3 - 10/08/2018 01:38 PM - Knödlseder Jürgen

- % Done changed from 30 to 40

I added a test to the npred() methods of GCTAModelAeffBackground, GCTAModelIrfBackground and GCTAModelBackground that verifies that the RoI centre is not significantly different from the pointing direction. In case of difference, an exception is thrown.

#4 - 10/08/2018 01:50 PM - Knödlseder Jürgen

- Related to Change request #2695: Allow RoI centres different from pointing direction in unbinned analysis added

#5 - 10/08/2018 05:18 PM - Knödlseder Jürgen

I added Python interfaces for the existing classes and merged the current code into the integration branch.

#6 - 10/08/2018 05:52 PM - Knödlseder Jürgen

Now also merged in devel. Issue is however not yet finished, we want to have new background models at the end.

#7 - 10/18/2018 04:23 PM - Knödlseder Jürgen

- % Done changed from 40 to 50

I added two further classes for spatial model of the background:

- GCTAModelSpatialMultiplicative implements a multiplicative spatial model, allowing to multiply spatial components to form a combined background model
- GCTAModelSpatialGradient implements a spatial gradient over the field of view

The XML format for these models is as follows:

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="My model" type="CTABackground" instrument="CTA">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor"   scale="1e-6" value="61.8" min="0.0"  max="1000.0" free="1"/>
      <parameter name="Index"       scale="-1"   value="1.85" min="0.0"  max="+5.0"   free="1"/>
      <parameter name="PivotEnergy" scale="1e6"  value="1.0"  min="0.01" max="1000.0" free="0"/>
    </spectrum>
    <spatialModel type="Multiplicative">
      <spatialModel type="Gaussian">
        <parameter name="Sigma" scale="1.0" value="3.0" min="0.01" max="10.0" free="1"/>
      </spatialModel>
      <spatialModel type="Gradient">
        <parameter name="Grad_DETX" scale="1.0" value="0.0" min="-10.0" max="10.0" free="1"/>
```

```
      <parameter name="Grad_DETY" scale="1.0" value="0.0" min="-10.0" max="10.0" free="1"/>
    </spatialModel>
   </spatialModel>
  </source>
</source_library>
```

I tested this model on the H.E.S.S. data and it seems to describe quite nicely the background event distributions.

I added a new cscript called csbkgmodel that automatically generates background models of this format. The csbkgmodel allows to generate runwise background models, and it loops over each observation in an observation definition XML file and prefits the background model to the data. This is in particular necessary if the power law should be replaced by a node function, which seems necessary for HESS. For stability reasons, csbkgmodel first fits a power law, and then replaces the power law by a 8 nodes function and refits the new model. This procedure seems to result in quite stable results.

#8 - 12/12/2018 02:27 PM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 50 to 100

For the time being this is sufficient, hence close the issue.