BkgModel

November 18, 2019

1 How to model cosmic-ray background

One of the most important challenges in IACT astronomy is dealing with residual background due to cosmic-ray showers misclassified as candidate photons. In this tutorial you will learn how to deal with background for a real data set, for which there is no background model provided with the data/IRF. You will see how we can construct a background model from the data, and to compare results from different analysis methods to assess the impact of different strategies to deal with background on the results.

As usual start by importing the gammalib, ctools, and cscripts Python modules.

```
In [ ]: import gammalib
        import ctools
        import cscripts
```

You may want to use the matplotlib package for plotting.

```
In [ ]: %matplotlib inline
        import matplotlib.pyplot as plt
```

1.1 Dataset and selection

For this exercise we will use a public dataset from the H.E.S.S. experiment providing observations of the Crab nebula.

H.E.S.S. data must be already available on your computer. Let's check this is the case

```
In [ ]: obs = gammalib.GObservations('$HESSDATA/obs/obs_crab.xml')
        print(obs)

        obsinfo = cscripts.csobsinfo(obs)
        obsinfo['offset'] = True
        obsinfo['ra'] = 83.63 # coordinates of source
        obsinfo['dec'] = 22.01
        obsinfo.run()

        print('pointing zenith angles (deg): ' +
              ', '.join(str(x) for x in (obsinfo.zeniths())))
        print('offset angles w.r.t. the Crab nebula (deg): ' +
              ', '.join(str(x) for x in (obsinfo.offsets())))
```

We have 4 observations, for a total of 30219 candidate photons. The four observations were conducted with zenith angles of the pointing directions between 45 deg and 50 deg, and with offsets from the source of interest, the Crab nebula, between 0.5 and 1.5 deg approximately.

The first thing we want to do is selecting the events. In particular there are two important parameters to select on. * Energy threshold: for a real dataset this varies depending on atmospheric conditions, night-sky background ... The H.E.S.S. collaboration provides a recommended energy threshold as part of the IRF, that we can select by setting the (hidden) parameter usethres to DEFAULT. In this case we want to skip the manual energy selection by setting emin (or emax) to INDEF. * Offset from the center of the camera: in practise modelling the instrument response far away from the camera center is difficult, so we want to keep only events within a certain offset, that for H.E.S.S. can be be set to 2 deg.

```
In [ ]: select  = ctools.ctselect(obs)
        select['usethres'] = 'DEFAULT'
        select['emin']     = 'INDEF' # no manual energy selection
        select['rad']      = 2.
        select['tmin']     = 'INDEF' # no temporal selection
        select.run()
```

Let's check the observations after event selection.

```
In [ ]: for run in select.obs():
            print(run)
```

The four observations last about 28 minutes each, and the safe energy thresholds vary between 660 GeV and 870 GeV.

1.2 Generating a background model from the data

We can use the csbkgmodel tool to generate a background model from the data. This tool makes the hypothesis that, over the entire field of view, background dominates over gamma-ray emission. Later on the background model will be refined by fitting it to the data along with a model for the gamma-ray signal.

```
In [ ]: bkgmodel = cscripts.csbkgmodel(select.obs())
        bkgmodel['instrument'] = 'HESS'
        bkgmodel['spatial']    = 'LOOKUP'
        bkgmodel['slufile']    = '$CTOOLS/share/models/hess_bkg_lookup.fits'
        bkgmodel['gradient']   = True
        bkgmodel['spectral']   = 'NODES'
        bkgmodel['ebinalg']    = 'LOG'
        bkgmodel['emin']       = 0.7
        bkgmodel['emax']       = 30.0
        bkgmodel['enumbins']   = 8
        bkgmodel['runwise']    = True
        bkgmodel.run()
```

Some important remarks: * the spatial shape of the background is modelled using a lookup table that provides the background rate as a function of offset from the center of the camera and measured energy, generated from the empty-field observations made available by the H.E.S.S. collaboration; * we multiply the azimuthally symmetric lookup model by a bi-linear gradient in camera coordinates (2 free parameters); * the average spectrum is modelled using a piece-wise broken power law with 8 energy nodes between 700 GeV and 30 TeV (8 free parameters). The free parameters are fitted to the data for each observation (runwise = True), to get a first reasonable estimate of their values.

Let's look at the resulting background model.

```
In [ ]: print(bkgmodel.models())
```

Are the background model parameters varying from one observation to the other?

1.3 Finding candidate sources

We can search for candidate sources in a skymap. for the moment let's not apply any background subtraction.

```
In [ ]: skymap  = ctools.ctskymap(select.obs())
        skymap['bkgsubtract'] = 'NONE'
```

Visualize the skymap.

The Crab nebula is very bright, thus it should be visible by eye in the skymap. We can use the cssrcdetect tool to determine the source poisition from the data.

```
In [ ]: srcdetect = cscripts.cssrcdetect(skymap.skymap().copy()) # copy the map to avoid smoot
        srcdetect['srcmodel']   = 'POINT'
        srcdetect['bkgmodel']   = 'NONE' # we will determine the background model in a later s
        srcdetect['threshold']  = 5
        srcdetect['corr_rad']   =  0.1
        srcdetect.run()
```

Let's look at the resulting model. It contains a point source Src001 at the position of the Crab nebula.

```
In [ ]: print(srcdetect.models())
```

1.4 Unbinned analysis

Now we can combine the models we have derived for the background and the source and set it as model for the selected observations.

```
In [ ]: models = gammalib.GModels()
        for model in bkgmodel.models():
            models.append(model)
        for model in srcdetect.models():
            models.append(model)
        select.obs().models(models)
```

Now you can perform an unbinned fit of the model to the observations.

To make sure the source and background model satisfactorly reproduce the observations compute spatial and spectral residuals.

1.5 On/Off analysis

Perform a classical On/Off analysis without assuming any background model. Tip: since you have multiple observations you should decide whether you want to stack them or to keep them separated for a joint fit. When not using a background model it may be useful to stack so that you have larger statistics to determine the background rates from the Off regions. Remember to check the spectral residuals after doing the fit.

The classical On/Off analysis without background model may suffer from biases in the low-counting regime, when there are energy bins in the Off spectra with zero counts. One way to get around this is to perform an On/Off background analysis using a background model, that is, to use the cstat statistic.

For this we can use the background model previsouly dtermined, and we just need to pass the appropriate parameter values to csphagen. Note than in this case we do not want to stack the observations since our background model is defined per observation.

```
In [ ]: phagen = cscripts.csphagen(select.obs())
        phagen['srcname']       = 'Src001'
        phagen['ebinalg']       = 'LOG'
        phagen['emin']          = 0.66
        phagen['emax']          = 30.0
        phagen['enumbins']      = 20
        phagen['coordsys']      = 'CEL'
        phagen['ra']            = 83.63
        phagen['dec']           = 22.01
        phagen['rad']           = 0.2
        phagen['bkgmethod']     = 'REFLECTED'
        phagen['use_model_bkg'] = True
        phagen['maxoffset']     = 2.0
        phagen['stack']         = False # treat observations separately in joint likelihood fi
        phagen.run()
```

Now perform the likelihood fit of this On/Off variant and check again the spectral residuals.

How do the Crab parameters compare between the three different analyses (unbinned, On/Off with/without background model)? How do the spectral residuals in the source region compare? How do the spectral residuals in the Off region compare between the two cases with/without background model? Can you guess why?

1.6 For further exercise

- Explore other spatial models for csbkgmodel: Gaussian or energy-dependent Gaussian in lieu of the lookup table, with or without gradient. What is the impact of this choice on the source parameters? And on the fit residuals?
- Repeat the source detection procedure by using a background-subtracted skymap, either by using the csbkgmodel model or the ring background estimation. Does the position of the source change much? Would this be still the case for a fainter source?
- Compute the residual maps in a few different energy ranges. Compute the spectral residuals for a few different regions scattered across the field of view. What can you say about the quality of the background model? Does the conclusion change if you use, e.g., a Gaussian in lieu of the lookup table?