sensitivity

January 12, 2017

Abstract

Creates maps of point-source-detection sensitivity in counts s^{-1}, for a given set of exposure maps and energy bands.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING
EPIC PN	IMAGING

2 Use

The task is not XMM-specific.

3 Description

This task is called, together with **econvolverprep**, by the perl task **esensitivity**. The descriptions for all three of these tasks should be consulted together.

The overall purpose of **sensitivity** is to create maps of the minimum count rate of a point source which would just be detectable by **eboxdetect**. The first part of the algorithm calculates the minimum counts within the detection box which would just be detectable; in order to convert this counts-within-a-box to total count rate of the whole source[1], **sensitivity** needs to know the ratio between the two across the width of the instrument field of view. These ratio maps are calculated for each energy band by **econvolverprep** followed by **asmooth** and are supplied to **sensitivity** via the parameter `psfexpmapsets`.

3.1 Theory

The task **eboxdetect** takes as input N images of the same part of the sky in N separate energy bands, and (in 'map' mode) N images of the estimated background counts per pixel. The task applies a sliding

[1] The detection box usually excludes a significant portion of the Point Spread Function or PSF of the source.

box to these stacked images.[2] At each pixel, then, the task has available N values C_i of the actual number of events found within the box and N values $\langle B_i \rangle$ of the expected number of background events within the same box. Considering for a moment just the values from a single band in isolation, if there were no sources in the field, then the probability $p(C)$ of finding C counts in this box would be given by the familiar Poisson expression

$$p(C) = \frac{\langle B \rangle^C \; e^{-\langle B \rangle}}{C!}.$$

The probability $P_{\rm null}(C)$ that the observed count C is due only to background is therefore the sum of $p(x)$ for all values of x from C to infinity. This can be shown to be equal to

$$P_{\rm null}(C) = 1 - Q(C, \langle B \rangle)$$

where Q is the incomplete gamma function, defined by

$$Q(C, \langle B \rangle) = \frac{\int_{\langle B \rangle}^{\infty} \; dt \, e^{-t} t^{C-1}}{\int_0^{\infty} \; dt \, e^{-t} t^{C-1}}.$$

eboxdetect calculates the value of the `LIKE` column for the ith energy band from the likelihood $L_i = -ln(P_{\rm null})$. The `LIKE` value $L_{\rm summ}$ for the summary row[3] is calculated by **eboxdetect** as follows. First, the values of `LIKE` for the individual bands are summed. This sum can be shown itself to follow a Poisson-like probability distribution. Leaving out the details, the probability $P_{\rm null}(\bar{C})$ that the vector $\bar{C}$ of counts in the N bands is due solely to background can be shown to be approximately given by

$$P_{\rm null}(\bar{C}) = Q\Big[N, N - \sum_{i=1}^{N} Q\big(C_i, \langle B_i \rangle\big)\Big].$$

The summary-row value of `LIKE` is thus calculated from

$$L_{\rm summ} = -ln\Big\{Q\Big[N, N - \sum_{i=1}^{N} Q\big(C_i, \langle B_i \rangle\big)\Big]\Big\}. \tag{1}$$

If there is no source at that location then clearly $C_i = B_i$. Suppose however that there is actually a source which contributes, on average, $\langle S_i \rangle$ counts to band i within the box in question. Equation 1 becomes in this case

$$L_{\rm summ} = -ln\Big\{Q\Big[N, N - \sum_{i=1}^{N} Q\big(B_i + S_i, \langle B_i \rangle\big)\Big]\Big\}.$$

The question which **sensitivity** should answer is, how small can the S_i become while retaining $L_{\rm summ} \geq L_{\rm cutoff}$? The first step in answering this is to reduce the degrees of freedom in the problem. For this

[2] Note that, although the name **eboxdetect** starts with an 'e', the detection stage is not EPIC- nor even XMM-specific. The only EPIC-specific part of the **sensitivity** algorithm is the conversion from counts-in-box to full count rate, because this requires knowledge of the PSF. This is taken care of by **econvolverprep**.

[3] This row represents more or less the average of the column values from the individual bands, although columns such as `SCTS` are simply summed to give the summary-row value.

purpose it is convenient to express each $\langle S_i \rangle$ as a product of two factors: a purely spectral factor $\mu_i = \langle S_i / \sum S_i \rangle$, which will be held fixed, and a pure flux factor $\langle S \rangle = \langle \sum S_i \rangle$, which we vary. Let us find that $\langle S_{\rm min} \rangle$ which is the solution to

$$L_{\rm cutoff} = -ln\Big\{Q\Big[N, N - \sum_{i=1}^{N} Q\big(B_i + \mu_i \langle S_{\rm min} \rangle, \langle B_i \rangle\big)\Big]\Big\}. \quad (2)$$

This value of $\langle S \rangle$ represents the expectation value of counts within the detection box of a source of spectrum μ_i which will, on average, result in a source detection against the given background.

This is exactly what **sensitivity** does: solve equation 2. All that remains after that is to calculate $\langle S_{i,\rm min} \rangle = \mu_i \langle S_{\rm min} \rangle$ for all i, then convert these counts values to count rates by multiplication by the box exposure at that pixel, supplied via the parameter `psfexpmapsets`.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

weights	yes	real		0 ≤`weights`

This list of real values contains the information about the spectrum of the source. They correspond to the μ_i values in equation 2. They need not be normalized however: the task will do this itself. At least one of the values in the list must be greater than zero. Note that these values represent the detected spectrum of the source; that is, μ_i is the total number of detected counts within energy band i during a given exposure time, on the optic axis, assuming no chip gaps or bad pixels under the PSF.

boxbkgmapsets	yes	dataset		

A list of input images, one per energy band. These images are made by **esensitivity** calling **asmooth**. They are formed by convolving the background maps with the square detection box.

psfexpmapsets	yes	dataset		

A list of input images, one per energy band. These images are made by **esensitivity** calling **econvolverprep** followed by **asmooth**. They are formed by truncating the normalized PSF within the detection box and convolving the exposure maps with the result. Thus these images are in units of seconds and in some sense represent the exposure 'seen' within the sliding source-detection box.

expmapsets	yes	dataset		

A list of exposure maps. These were utilized in a previous version of the task but appear to serve little function in the present version. In future I'll probably remove them.

sensmapsets	yes	dataset		

The output sensitivity maps, one per energy band. These are in units of counts s^{-1}.

likemin	yes	real		0 <`likemin`

The same cutoff value of detection likelihood L_{cutoff} which was supplied to the parameter of the same name of **eboxdetect**.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

badMinDetPoissonCountsStatus *(error)*
The subroutine minDetPoissonCounts returned with a bad status.

tooFewWeights *(error)*
The user didn't supply any values to the parameter `weights`.

wrongNumBoxBackgroundSets *(error)*
The number of elements in `boxbkgmapsets` didn't equal the number in `weights`.

wrongNumBoxExposureSets *(error)*
The number of elements in `psfexpmapsets` didn't equal the number in `weights`.

wrongNumExpImageSets *(error)*
The number of elements in `expmapsets` didn't equal the number in `weights`.

wrongNumSensMapSets *(error)*
The number of elements in `sensmapsets` didn't equal the number in `weights`.

imageDimensionsUnequal *(error)*
All the images supplied must have the same dimensions. The task found that this was not the case.

nonZeroStatusSensCalc *(warning)*
The subroutine minDetPoissonCounts returned with a non-zero (ie warning) status. This situation is not very informative. I'll improve it in the future.
corrective action: No action at this stage

negativeBackground *(warning)*
The subroutine minDetPoissonCounts found negative values in some of the supplied background images.
corrective action: The same pixels of the sensitivity maps are set to 0

negativeSrcCounts *(warning)*
The subroutine minDetPoissonCounts found negative values in some of the supplied 'source' images.
corrective action: The same pixels of the sensitivity maps are set to 0

6 Input Files

1. A list of N images, supplied to the `boxbkgmapsets` parameter, which are made by **esensitivity** calling **asmooth**. They are formed by convolving the background maps used by **eboxdetect** with the square detection box.
2. A list of N images, supplied to the `psfexpmapsets` parameter, which are made by **esensitivity** calling **econvolverprep** followed by **asmooth**. They are formed by truncating the normalized PSF within the detection box and convolving the exposure maps with the result. Thus these images are in units of seconds and in some sense represent the exposure 'seen' within the sliding source-detection box. Their dimensions and coverage of the sky must be idenitical to the `boxbkgmapsets` images.
3. A list of N exposure maps, supplied to the `expmapsets` parameter. These appear to serve little function in the present version of the task so I'll remove them in the future.

7 Output Files

1. A series of N images, identical in size and sky coverage with the input images. The ith image in the sequence records at each pixel the count rate in band i of a source, of detected spectrum given by `weights`, which would on average give rise to a summary likelihood value equal to `likemin` under the supplied conditions of background and exposure. However pixels at which the box exposure is zero are also set to zero. These images contain all the attributes of the input `expmapsets`.

8 Algorithm

`I'll do this another time.`

9 Comments

-

References