esensitivity

January 12, 2017

Abstract

Creates maps of sensitivity (ie the lowest count rate of source which is detectable by eboxdetect) in counts s^{-1}, for a given source spectrum, exposure and set of energy bands.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING
EPIC PN	IMAGING

2 Use

pipeline processing	possibly
interactive analysis	yes

3 Description

This task operates in conjunction with the tasks **econvolverprep** and **sensitivity**; users should consult the documentation for all three of these tasks together.

esensitivity is, in a sense, the inverse of **eboxdetect**, which, as most users know, searches for point sources in XMM EPIC images. **eboxdetect** finds sources by comparing the number of x-ray events C which fall within a square sliding box with the expected number of background events $\langle B \rangle$ expected to occur within the same box. At locations where the detection likelihood L, which is minus log of the probability that the detected counts is due to a Poissonian statistical fluctuation of the background, exceeds a certain limit, the software reports the occurrence of a point source. At any given pixel of the input image there is, therefore, a fixed relationship between C, $\langle B \rangle$ and L. The essential task of **esensitivity** is to invert this relationship, ie to answer the question: given a cutoff value of detection likelihood $L_{\rm cutoff}$ and the expected background counts $\langle B \rangle$, to what value of detected counts $C_{\rm min}$ do these numbers correspond. $C_{\rm min}$ thus represents the minimum value of counts which will be classified by **eboxdetect** as a source.

The above description assumes that source detection is carried out on a single image. However **eboxdetect** is far more commonly employed to search for point sources on several images in parallel. These

images are assumed to be of the same part of the sky but in different energy bands. In this case the recipe for calculating likelihood is a little more complicated. **esensitivity** can invert this style of source detection too but the reader is referred to the task documentation of **sensitivity** for details of the algorithm. **sensitivity** is the task which actually performs the inversion.

esensitivity is a perl script which calls firstly **asmooth** in order to convolve the background maps with the detection box; then a task called **econvolverprep** which calculates the relationship at each image pixel between total counts beneath the point spread function (PSF) and the counts falling within the detection box; finally **sensitivity** is called. The output is a series of images, 1 per energy band, which map the value of source count rate which would just be large enough to be detected on average as a point source in the corresponding input image.

3.1 Caveats

There are many ways in which sources might be detected and selected; however **esensitivity** is designed to cope with only a single method, namely the sliding-box detection on parallel images as performed by **eboxdetect**. The task cannot determine the details of the detection method for itself, thus it is up to the user to ensure that the parameters used to invoke **esensitivity** match those of the corresponding call to **eboxdetect**. Ideally **eboxdetect** should be called using the following parameter settings:

- –bkgimagesets=list of N background maps β_i
- –usemap=yes
- –detmasksets=list of N detector masks (made from exposure maps by **emask**
- –expimagesets=list of N exposure maps ϵ_i
- –nruns=1. NOTE that this is neither the default value nor that used in the XMM pipeline.
- –likemin=$L_{\rm cutoff}$
- –boxsize=b
- –imagesets=list of N images I_i
- –boxlistset=output file name
- –pimin=list of N lower bounds $E_{i,\rm lo}$ of the input images etc
- –pimax=list of N upper bounds $E_{i,\rm hi}$ of the input images etc

The correct matching call of **esensitivity** uses the corresponding values of the appropriate parameters, viz:

- –bkgmapsets=list of N background maps β_i
- –expmapsets=list of N exposure maps ϵ_i
- –likemin=$L_{\rm cutoff}$
- –boxhalfsize=$(b-1)/2$. Note that b as presently defined in **eboxdetect** is always odd.

Note that **esensitivity** also needs to know the mean energies $E_{i,\mathrm{mean}} = (E_{i,\mathrm{hi}} + E_{i,\mathrm{lo}})/2$. If the energy bands are stored in the data subspaces (DSSs) of the images, then the user should set `psfestyle`='dss' and supply the list of images I_i to `dssimagesets`; if not, the parameter `psfestyle` will need to be set to 'user' (currently the default value) and a list of the mean values explicitly supplied via the parameter `psfenergies`.

It should be emphasised that **esensitivity** cannot yet be made to match the complicated source detection and selection procedure which is currently used to make XMM-product source lists. Application of **esensitivity** to such lists will probably produce results which are apprximately correct but the degree of approximation is not known and for this reason I don't recommend the procedure. In future I hope we will make changes to the way these product source lists are made which will allow matching sensitivity maps to be made.

3.2 Source spectrum

eboxdetect in its present form makes no assumptions about the source spectrum, but I have not thought of a way to calculate any sort of generic sensitivity: it seems to me impossible to disassociate the detection sensitivity from the source spectrum. In other words, though the detection procedure may be generic, I would naturally expect it to function with different efficiency on source populations with differing spectra. Hence the spectrum of the source of interest must be supplied as a list of weights μ_i to the parameter `weights`. The interpretation of the ith weight 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. The weight values don't have to be normalized to 1 (**esensitivity** will do this itself), but it is not permitted to have them all = 0. A good way to get a generic set of weight values is to average normalized sets of `SCTS` values from an ensemble of sources detected in the same set of energy bands. Note that by 'spectrum' is meant the *detected* spectrum of the x-ray source, that is the x-ray spectrum after multiplication by the instrument response function. Note also that this unfortunately introduces an approximation to the **esensitivity** algorithm, because due to the energy dependence of the mirror vignetting function the same source will have a slightly different detected spectrum at the edge of the field of view than at the centre.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
bkgmapsets	yes	dataset		

The list of N background maps B_i used in the **eboxdetect** source detection run.

expmapsets	yes	dataset		

The list of N exposure maps ϵ_i used in the **eboxdetect** source detection run.

weights	yes	real		0 ≤`weights`

The detected spectrum of the source (that is, the x-ray spectrum multiplied by the instrument response function) as represented by N weights, the ith weight μ_i representing 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.

sensmapsets	yes	dataset		

The list of N output sensitivity maps (maps of the lowest detectable point-source count rate).

likemin	yes	real		0 <`likemin`

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

boxhalfsize	yes	integer		0 ≤`boxhalfsize`

This should match the value of –boxsize supplied to **eboxdetect**. Suppose –boxsize=b; the corresponding value of `boxhalfsize` should be $(b-1)/2$. Note that –boxsize as presently defined in **eboxdetect** is always odd.

naxial	no	integer	36	4 ≤`naxial`

This parameter is passed to **econvolverprep** and a description of its effects should be sought therein.

nradial	no	integer	10	1 ≤`nradial`

This parameter is passed to **econvolverprep** and a description of its effects should be sought therein.

psfestyle	no	string	user	user—dss

The task needs to know the mean values of the energy bands of the N input images I_i to **eboxdetect**. There are two ways in which this information can be supplied to **esensitivity**: either directly via the

parameter `psfenergies`, or via data subspace (DSS) records of all event selections within the image datasets themselves as supplied to the parameter `dssimagesets`. NOTE that the inclusion of a DSS in XMM EPIC images is optional, thus these may not be present.

psfenergies	yes	real		1 ≤`psfenergies`

If `psfestyle`='user', the task seeks a list of N mean energy values in the present parameter.

dssimagesets	yes	dataset		

If `psfestyle`='dss', the task seeks a list of N mean energy values in the data subspaces of the N datasets I_i supplied to the present parameter.

astest	no	boolean	no	

Set this to 'yes' if you want to test the perl script **esensitivity** without invoking any child processes (ftools or other sas tasks).

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.

tooFewBkgMaps *(error)*
The list `bkgmapsets` was empty.

wrongNumExpMaps *(error)*
The number of elements in `expmapsets` doesn't equal the number in `bkgmapsets`.

wrongNumWeights *(error)*
The number of elements in `weights` doesn't equal the number in `bkgmapsets`.

wrongNumSensMaps *(error)*
The number of elements in `sensmapsets` doesn't equal the number in `bkgmapsets`.

wrongNumPsfEnergies *(error)*
The number of elements in `psfenergies` doesn't equal the number in `bkgmapsets`.

wrongNumDssImages *(error)*
The number of elements in `dssimagesets` doesn't equal the number in `bkgmapsets`.

badPsfStyle *(error)*
The value of parameter `psfestyle` was not recognized.

sasCcfEnvVarNotSet *(error)*
The environment variable SAS_CCF is not set. It should be set to point to a Calibration Index File (CIF).

noCifFound *(error)*
The environment variable SAS_CCF was not empty, but the Calibration Index File (CIF) it pointed to does not exist.

asmoothFailed *(error)*
esensitivity attempted to invoke **asmooth** but failed.

econvolverprepFailed *(error)*
esensitivity attempted to invoke **econvolverprep** but failed.

sensitivityFailed *(error)*
esensitivity attempted to invoke **sensitivity** but failed.

6 Input Files

1. A list of the N background maps, supplied to the `bkgmapsets` parameter, which were used by **eboxdetect** in its detection run.
2. A list of the N exposure maps, supplied to the `expmapsets` parameter, which were used by **eboxdetect** in its detection run.

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

esensitivity is a perl script and can probably stand in as its own algorithm section.

9 Comments

•

References