evarimgen

November 4, 2014

Abstract

This task makes a time-variability image.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING
EPIC PN	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This task takes an event list and makes an image in which the value in each pixel is proportional to the variability with respect to time of the flux of events which fall within that pixel. The task cycles through all the events in the list which fall within the spatial extent of the image; for each of these events, a value δ of the Kolmogorov-Smirnov statistic is calculated. The value of δ for the i,jth pixel depends on both the cumulative number of events n within the whole image and the cumulative number $n_{i,j}$ within the pixel. The formula for δ is

$$\delta_{i,j} = \left| \frac{n}{N} - \frac{n_{i,j}}{N_{i,j}} \right|,$$

where N is the total number of events that fall within the image bounds and $N_{i,j}$ is the total number of events that fall within the i,jth pixel. (The value of N is easily found by counting all the events that fall within the image bounds; whereas $N_{i,j}$ can be read from the total flux image, provided that this is binned up to the same bounds and pixel size.) For each pixel, the raw variability image is formed from the maximum value of δ encountered for each pixel during the above procedure. The values $V_{i,j}$ in the final variability image are obtained by using the weighting

$$V_{i,j} = \sqrt{\eta} + 0.12 + 0.11/\sqrt{\eta},$$

where

$$\eta = \frac{nN}{n+N}.$$

These last two formulae can be found within [1]

Pixels which have too few events for the KS statistic to be reliably calculated (see parameter `cutoff`) are set to the value of 0.827574; at this value of the Kolmogorov-Smirnov (KS) statistic, it can be shown that the probability that this value or higher results from a non-variable series is 50%. Note the implication that pixels having zero variability are *not* zero-valued in the output image.

Note that the KS statistic does not require any information about the time of occurrence of the events. It is therefore immune to gaps in the sequence (such as might be caused by good time interval (GTI) filtering) and not very sensitive to variability of the background. However there is some undesirable effect due to background variability. The KS statistic of a series of events measures the magnitude of the difference between the variability of that series and a reference series. In the present version of the task, the reference series is provided by the events occurring within the entire image bounds. If this is dominated by background flares, steady point sources will differ in variability from the reference through being *less* variable, but will nevertheless display high KS values. It is therefore recommended that the event list be filtered to remove events from periods in which background bursts occurred, before submitting the event list to **evarimgen**. A future version of **evarimgen** may contain a facility for performing this filtering within the task itself.

The statistic is, on the other hand, sensitive to the ordering of the events in time. Since event lists produced by the sas usually contain at least some out-of-order events, a facility has been provided within the task for sorting the events into time order. This is requested via the parameter `sortevlist` (which is 'yes' by default). If `sortevlist`=no, **evarimgen** checks the ordering of the events and generates a warning if any out-of-order events are found. The sorting is internal to the task and in no circumstances is the event list modified.

The task as presently constructed does not handle well a situation in which a single bright source contributes most of the events. (However, this must be much brighter than simple intuition would suggest, since the sheer number of background pixels means that they nearly always provide the majority of events.) In this case, no matter how variable the source, **evarimgen** is likely to show it as only weakly variable, because it is largely being compared with itself. Future developments in the task may solve this problem.

A knowledge of $N_{i,j}$, the total flux of events within each pixel, is necessary in order to calculate the Kolmogorov-Smirnov statistic. If a total-flux image has previously been calculated from the event list, this may be submitted to the task via the parameters `withimageset` and `imageset`. However at the moment it is better practice to calculate this within the task, so it is recommended that the default value of `withimageset`=no be used. In this case `imageset` is the name of the newly created output flux image, and therefore it may not be safe to leave this parameter at its default value. If `withimageset`=no and `imageset` happens to point to another file of the same name, the previous file may be overwritten.

The first priority for future development of **evarimgen** will be to allow the task to read the variability image size and pixel numbers from the input flux image, where this is supplied. The variability image would then automatically be compatible with an exposure map produced using the same image size settings, which is presently not the case. Lack of an exposure map or other mask image of the same dimensions prevents optimum application of the smoothing task **asmooth** to the output of **evarimgen**.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

evlistset	yes	dataset	‘foo.bar’	

The name of the event list.

xcolumn	no	string	X	

Name of the column to supply image x coordinate.

ycolumn	no	string	Y	

Name of the column to supply image y coordinate.

xrangetype	no	string	maxlegal	‘maxlegal’/‘maxactual’/‘user’

This controls the cutoff points for x values. If `xrangetype` is set to ‘maxlegal’, the values of `TLMINn` and `TLMAXn` are used, where `n` is the number of the column whose name is given by the `xcolumn` parameter. A value of `xrangetype`=‘maxactual’ dictates that `TDMINn` and `TDMAXn` (the true minimum and maximum values of the column entries) should be used instead. If `xrangetype`=‘user’, the task looks for user-supplied minimum and maximum values in the parameters `xmin` and `xmax`. Events which have x values outside the range are discarded.

yrangetype	no	string	maxlegal	‘maxlegal’/‘maxactual’/‘user’

Similar to `xrangetype`, but for the y coordinate.

imagebinning	no	string	numberofpixels	‘numberofpixels’/‘pixelsize’

Controls the way in which the number of x and y pixels are calculated. If `imagebinning`=‘numberofpixels’, the numbers of x and y pixels are entered directly via the parameters `nxpixels` and `nypixels`. On the other hand if `imagebinning`=‘pixelsize’, the user must enter the x and y dimensions of the pixel via the parameters `xpixelsize` and `ypixelsize`.

xmin	no	real	-3000.0	

When `xrangetype` is set to ‘user’, this parameter is expected.

xmax	no	real	3000.0	

When `xrangetype` is set to ‘user’, this parameter is expected.

ymin	no	real	-3000.0	

When `yrangetype` is set to ‘user’, this parameter is expected.

ymax	no	real	3000.0	

When `yrangetype` is set to ‘user’, this parameter is expected.

nxpixels	no	integer	600	$1 <= nxpixels <= 1000$

When `imagebinning` is set to ‘numberofpixels’, this parameter is expected.

nypixels	no	integer	600	$1 <= nypixels <= 1000$

When `imagebinning` is set to ‘numberofpixels’, this parameter is expected.

xpixelsize	no	real	1.0	$xpixelsize > 0$

When `imagebinning` is set to 'pixelsize', this parameter is expected.

ypixelsize	no	real	1.0	*ypixelsize* > 0

When `imagebinning` is set to 'pixelsize', this parameter is expected.

cutoff	no	integer	4	$cutoff >= 0$

Pixels with fewer events than the value of `cutoff` are set to zero in the variability image.

withimageset	no	boolean	no	

This is set if the total flux image should be read (using the name given in parameter `imageset`) rather than created. Note that, if `withimageset`=no, the total flux image is created within the task and saved to the name given in parameter `imageset`.

imageset	no	string	'outimage.ds'	

Name of the total flux image. If `withimageset`=yes, the task attempts to read the total flux image from a file of this name. If `withimageset`=no, the task generates the total flux image internally and saves it to a file of this name.

varimageset	no	string	'outvarimage.ds'	

The variability image is saved to a file of this name.

sortevlist	no	boolean	yes	

The algorithm works best if the events are sorted in time. This parameter tells the task to do this sort. The task generates a warning if `sortevlist`=no and the events are out of time order.

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.

xPixelTooSmall *(error)*
The X range, when divided by the specified pixel size, gives too many pixels (there should be fewer than 1000). Either the chosen pixel size is too small or the X range is too large.

yPixelTooSmall *(error)*
The Y range, when divided by the specified pixel size, gives too many pixels (there should be fewer than 1000). Either the chosen pixel size is too small or the Y range is too large.

badImageSize *(error)*
A flux image has been read, which has dimensions in either pixels or world coordinates that is different to those specified via the parameter interface. This error will be downgraded to a warning when **evarimgen** becomes able to use the flux image dimensions to set those of the variability image.

scaledImage *(error)*
The input flux image has non-integer values. It cannot therefore be used to calculate the Kolmogorov-Smirnov statistic.

unorderedEvlist *(warning)*
The entries in the TIME column of the event list are not uniformly in increasing order. The task should be run again with the parameter `sortevlist` set.
corrective action: No action.

NoWCS *(warning)*
The event list does not contain a complete set of WCS keywords for one of the columns.
corrective action: Keyword values are constructed from specified image dimensions.

6 Input Files

1. (Mandatory) A FITS event list which contains at least two columns, one of which is named `TIME`. Usually this will be expected to contain at least three columns, the `TIME` column plus two others giving image x and y coordinates.
2. (Optional) A FITS total-flux image made previously from the event list. The data type may be any of those output by **evselect**, eg int8, int16, int32, real32 or real64. The bounds and size in pixels of this image must be the same as specified for the variability image via the relevant **evarimgen** parameter settings.

7 Output Files

1. A total-flux image, if one has not been supplied to the task. This has data type real32.
2. The variability image. This has data type real32.

8 Algorithm

```
read parameters;

if (sortevlist) {
  sort the event list;
} else {
  check that the events occur in time order, warn if not;
}

calculate the dimensions of the variability image;

if (withimage) {
  load fluxImage;
} else {
  make fluxImage;
  write fluxImage as a FITS file;
}

# Make the KSS image:
ksImage(1:max_i, 1:max_j) = 0.0;
counts( 1:max_i, 1:max_j) = 0;
for (event from 1 to n_events) {
  if (the event is within the image bounds) {
    calculate pixel (i,j) that this event falls into;
    counts(i, j) = counts(i, j) + 1;
    d = abs((event - 1) / n_events - counts(i, j) / fluxImage(i, j));
    if (ksImage(i, j) < d) {
      ksImage(i, j) = d;
    }
  }
}

# Weight the KSS image:
foreach (i) {
  foreach (j) {
    if (counts(i, j) > cutoff) {
      sqrtNe = sqrt(n_events * counts(i, j) / (n_events + counts(i, j)));
      ksImage(i, j) = ksImage(i, j) * (sqrtNe + 0.12 + 0.11 / sqrtNe);
    } else {
      ksImage(i, j) = 0.827574; # value at which prob of variability is 50%
    }
  }
}

write ksImage as a FITS file;
```

9 Comments

-

References

[1] W.T. Vetterling W.H. Press, S.A. Teukolsky and B.P. Flannery. *Numerical Recipes in Fortran.* Cambridge University Press, New York, 1992.