Subject: Re: STANDARD DEVIATON
Posted by Struan Gray on Tue, 01 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Oops. I need more (or less) coffee:

> function imageSD, image
>
> localmean = smooth(float(image), 3, /edge_truncate)
> localsd = sqrt((float(image)-temporary(localmean))^^2)
> localsd = smooth(temporary(localsd), 3, /edge_truncate)
>
> return, localsd
>
> end

Naturally, you should take the local average *before*
the square root.

function imageSD, image

localmean = smooth(float(image), 3, /edge_truncate)
localsd = (float(image)-temporary(localmean))^^2
localsd = smooth(temporary(localsd), 3, /edge_truncate)
localsd = sqrt(temporary(localsd))

return, localsd

end

Struan

Subject: Re: STANDARD DEVIATON
Posted by Benno Puetz on Tue, 01 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster wrote:

> For some 3x3 pixels x_i (and a mean of x_mean), the standard deviation
> is
> s = 1/9 * sqrt(sum((x_i-x_mean)^2)). img=img-smooth(img,3) does the
> inner subtraction of the x_means,

unfortunately, that's not quite true (except for special cases) since the x_i
get only their respective mean subtracted, not the center pixel mean. I came

up with the same solution initially but it would not stand the test ...

haven't had a better idea since :(

--
Benno Puetz
Kernspintomographie
Max-Planck-Institut f. Psychiatrie Tel.: +49-89-30622-413
Kraepelinstr. 10 Fax : +49-89-30622-520
80804 Muenchen, Germany

Subject: Re: STANDARD DEVIATON
Posted by Struan Gray on Tue, 01 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Ben Marriage, ben@met.ed.ac.uk writes:

> I want to calculate the standard deviation of a 3x3
> pixel area for each element of a satellite image
>
> Does anyone know of a simple and elegant (read "quick")
> method of doing this without using loops?

You can use a box smooth to do the local averaging and renormalisation, with variable widths if needed. If you use the fact that the s.d. is the square root of the local variance, you'll need to smooth twice: once when calulating a local mean, and again when creating the s.d. array.

```
function imageSD, image

  localmean = smooth(float(image), 3, /edge_truncate)
  localsd = sqrt((float(image)-temporary(localmean))^^2)
  localsd = smooth(temporary(localsd), 3, /edge_truncate)

  return, localsd

end
```

You can set different edge-effects and averaging widths by passing different parameters to the box smooth function. The box smooth normalises by the square of the smoothing width, so if you want to treat the image as a estimate of a population you'll have to renormalise - but this is pretty unlikely with image data.

Struan

Subject: Re: STANDARD DEVIATON
Posted by Alex Schuster on Tue, 01 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Ben Marriage wrote:

> I want to calculate the standard deviation of a 3x3 pixel area for each
> element of a satellite image (5000x2000 pixels). At the moment I use for
> loops to scan through the image and then use the built-in IDL standard
> deviation routines. I've tried to speed this up with the use of
> convolution and whatnot but never seemed to solve the problem. Does
> anyone know of a simple and elegant (read "quick") method of doing this
> without using loops?

Without loops yes, elegant, well yes of course, but quick I don't know?
Our fastest SUN takes 8.5 seconds for a dist(5000, 2000). How fast was
your routine using convolution, and how fast should it be?

For some 3x3 pixels x_i (and a mean of x_mean), the standard deviation
is
s = 1/9 * sqrt(sum((x_i-x_mean)^2)). img=img-smooth(img,3) does the
inner subtraction of the x_means, img=img^2 squares each element,
convol(img, k) sums them all up Then qrt(img)/9 gives the result.

```
function foo, img, width
return, sqrt( convol( ( img - smooth( img, width ) )^2, $
                  fltarr( width, width ) + 1.0 ) ) / width^2
end
```

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de