Subject: Re: algorithm question. Can I get rid of the for loop?
Posted by bobgstockwell on Fri, 22 Mar 2013 17:29:31 GMT
View Forum Message <> Reply to Message

On Thursday, March 21, 2013 3:32:15 PM UTC-6, Søren Frimann wrote:
> Hi All.
>
>
>
> I have an implementation of a hampel filter (see e.g.
http://exploringdatablog.blogspot.dk/2012/01/moving-window-f ilters-and-pracma.html) in IDL.
>
>
>
> My implementation looks like this:
>
>
>
> ##
>
> FUNCTION hampel, x, y, dx, THRESHOLD=threshold
>
>
>
> Compile_Opt idl2
>
>
>
> IF N_Elements(threshold) EQ 0 THEN $
>
>   threshold = 3
>
>
>
> ;initialize arrays
>
> s0 = FltArr(N_Elements(y))
>
> y0 = FltArr(N_Elements(y))
>
> yy = y
>
>
>
> FOR i=0,N_Elements(y)-1 DO BEGIN
>
>   index = Where((x GE x[i] - dx) AND (x LE x[i] + dx))
>

```
>    y0[i] = Median(y[index]) ; Median filtering
>
>    s0[i] = 1.4826*Median(Abs(y[index] - y0[i])) ;estimating uncertainty
>
>  ENDFOR
>
>
>
>  ol = Where(Abs(y - y0) GE threshold*s0)  ;Index of outliers
>
>  yy[ol] = y0[ol]
>
>
>
>  result = Create_Struct('y',yy, $
>
>                         'sigma',s0)
>
>
>
>  RETURN, result
>
>
>
>  END
>
>  ##################################################
>
>
>
```

> the filter runs a moving window of width 2*dx measured in the same units as x.
>
> x is generally not uniformly spaced (so there's not a constant number of points inside the window as it moves).
>
> x and y can be quite long vectors so the filter takes a long time to run.
>
> Can anyone see any method for speeding the code up?
>
> Any help would be much appreciated!
>
> Cheers,
>
> Søren

Using histogram and reverse indices would be much faster than looping and whereing. (i.e. get all of your "index" arrays in one call, rather than n_elements(y) calls of where).

but, a major point, you must check to see if your where() finds any matches, before using the index and calculating the median.

cheers,
bob