Subject: Re: algorithm question. Can I get rid of the for loop?
Posted by Søren Frimann on Wed, 27 Mar 2013 13:31:25 GMT
View Forum Message <> Reply to Message

Den fredag den 22. marts 2013 14.25.44 UTC+1 skrev Heinz Stege:
>
> Before entering the loop get the indices of the lower and upper limits
>
> for all x values by use of the VALUE_LOCATE function. Then you can use
>
> this pre-calculated indices instead of the index array from the WHERE
>
> function.

That was a very nice hint indeed! Below an implementation where this has been done (as well as a few general updates). It's much faster than my older solution although, it retains the for loop

```
################################################################

FUNCTION hampel, x, y, dx, THRESHOLD=threshold

Compile_Opt idl2

IF N_Elements(threshold) EQ 0 THEN threshold = 3

s0 = FltArr(N_Elements(y))
y0 = FltArr(N_Elements(y))
yy = y

lower_Boundary = Value_Locate(x,x-dx)+1 ; indices of lower boundaries
upper_Boundary = Value_Locate(x,x+dx)   ; indices of upper boundaries

FOR i=0,N_Elements(y)-1 DO BEGIN
  IF lower_Boundary[i] EQ upper_Boundary[i] THEN BEGIN
    ; only one point in gap
    y0[i] = y[i]
    s0[i] = !Value.F_NAN
  ENDIF ELSE BEGIN
    ; Two or more points in gap
    y_temp = y[lower_Boundary[i]:upper_Boundary[i]]
    y0[i] = Median(y_temp) ; median filtering
    s0[i] = 1.4826*Median(Abs(y_temp - y0[i])) ; estimating uncertainty
  ENDELSE
ENDFOR

gp = Where(Abs(y - y0) LT threshold*s0)  ;index of good points
ol = Where(Abs(y - y0) GE threshold*s0,n)  ;index of outliers
```

```
yy[ol] = y0[ol] ; replace outliers

result = Create_Struct('y'    ,yy, $
                       'sigma',s0, $
                       'gp'   ,gp, $
                       'ol'   ,ol, $
                       'n'    ,n) ; number of outliers

RETURN, result

END

##################################################################
```

Den fredag den 22. marts 2013 18.29.31 UTC+1 skrev bobgst...@gmail.com:
> 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).
>

I've looked into it, and I really don't see a way of using histogram, since it involves binning of the data, and my data aren't binned - rather they are subject to a moving window running smoothly over the data set.

Cheers,
Søren