Subject: Avoiding a for cicle
Posted by Ricardo Fonseca on Thu, 06 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi

I'm looking for a more efficient way of implementing the following (i.e. avoiding the for cycle) which is a routine for finding local maximuns

```
; Data is a 1D Array

s = Size(Data)

nx = s[1]

max_pos = [-1]

for i = 1, nx-1 do $
  if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
     max_pos = [[max_pos],i]

; and then throw away the first element...
```

Any ideas? Thanks in advance, Ricardo

Subject: Re: Avoiding a for cicle
Posted by davidf on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astro.cornell.edu) writes:
>
>> Alright code slingers... new challenge... find location of all peaks in a region
>> of n points (n odd), monotonically decreasing away from the peak. I.e. find
>> peaks of width n.
>>
>
> Since no one will take my challenge I'm forced to claim the prize for myself:
>
> wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)

Uh, I don't get it. :-(

Cheers,

David

P.S. Never mind tryin' to learn me. I don't got that long.

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by John-David T. Smith on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

> Alright code slingers... new challenge... find location of all peaks in a region
> of n points (n odd), monotonically decreasing away from the peak. I.e. find
> peaks of width n.
>

Since noone will take my challenge I'm forced to claim the prize for myself:

wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Avoiding a for cicle
Posted by davidf on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

> I think I can attend, but only if the "entertainment" is as good as I
> hear it was last year.

Springsteen was booked. We're still looking.

Cheers,

David

P.S. Oh, you mean *that* entertainment. No, we won't

be doing that again any time soon. Nearly our whole
"orphan's fund" went to pay for damages. :-(

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by Craig Markwardt on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) writes:

> Craig and JD,
>
> I'm forwarding both of your names to the Nominating Committee
> for this year's Distinguished Contribution in Mathematics
> award, given annually (or whenever there is a worthy
> candidate) at the IDL Expert Programmers Association
> blowout in October. I don't think there has ever been
> a tie for this award before, but you two guys certainly
> deserve it. You can both count on my vote and I'll be
> lobbying the other members as well.

I think I can attend, but only if the "entertainment" is as good as I
hear it was last year.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Avoiding a for cicle
Posted by davidf on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig and JD,

I'm forwarding both of your names to the Nominating Committee

for this year's Distinguished Contribution in Mathematics
award, given annually (or whenever there is a worthy
candidate) at the IDL Expert Programmers Association
blowout in October. I don't think there has ever been
a tie for this award before, but you two guys certainly
deserve it. You can both count on my vote and I'll be
lobbying the other members as well.

Well done!

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by Craig Markwardt on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

"J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>> This one does exactly what was requested, which I'm not sure of about
>> your solution, J.D. On the other hand, your solution may be more
>> physically meaningful since it involves smoothing.
>>
>
> Nice entry Craig. But unfortunatly it doesn't *alway* do exactly
> what was requested. It works fine for n=5, but for n>5 (7,9,...),
> the index is off. Part of the reason is in the way convol works.
> For n=5, nh=2, and convol subscripts are left of center (t+i-1 for
> i=0,1). For n=7, nh=3, and convol subscripts are (t+i-1 for
> i=0,1,2), which is centered. For n=9, nh=4, and you again have
> (t+i-2 for i=0,1,2,3), left of center again... and so on. The
> additional complication is that you are finding the center of the
> *wrapped* up-going and down-going entries... the center of the peak
> is offset from this by (nh+1)/2 (which happens to equal 1 when n=5
> or n=3). Adding this rather than 1 to the returned indices would
> make it correct. Here's an example for n=9 (^=up-goin,
> v=down-going):
> ...

Arghh. I stand corrected. This turns out to be the kind of "feature"

in CONVOL which we can turn off, by setting the keyword CENTER=0. I
actually read the documentation for the function, intended to put
CENTER=0 in, but forgot. I think now it works as advertised.

```
dd = d(1:*)-d
nh = (n-1)/2
wh = where(convol((dd GT 0) AND (dd(nh:*) LT 0), bytarr(nh)+1, nh, center=0) $
        EQ 1, count) + 1
```

> Now I'll explain my solution, which does indeed produce indentical
> results ...

I must admit your entry was the cruftier. Nice explanation.

Here is the algorithm I used to make my random spiky_data.sav. You
will find a 13-peak about every 4000 data points with this.

```
m = 1024L
d = randomn(seed,m)
for i = 1L, m-1 do d(i) = d(i-1)+d(i)
```

Craig

--

---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Avoiding a for cicle
Posted by Craig Markwardt on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) writes:
> Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:
>
>> dd = d(1:*)-d
>> nh = (n-1)/2
>> wh = where(convol((dd GT 0) AND (dd(nh:*) LT 0), bytarr(nh)+1, nh) EQ 1, ct)+1
>>
>> For the goobledy-gook impaired (aka DF :-),
>> dd is the first difference of the data
>> nh is the half-width of the peak
>> (dd GT 0) AND (dd(nh:*) LT 0) locates up-going followed by down-going points
>> convol(...) locates runs of length nh
>>
>> This one does exactly what was requested, which I'm not sure of about

>> your solution, J.D. On the other hand, your solution may be more
>> physically meaningful since it involves smoothing.
>
> Alright, now that Craig has oriented me a little bit,
> I find that I, uh..., have a *need* for this sort of thing. :-)
>
> I presume you gentlemen are testing these little theories of yours
> on a test data set. Could you supply such a data set for the
> rest of us to fool around with? And if you gave us just a little
> hint about how such a thing might be useful to *you*, that might
> help too. I might even take a stab at writing an article about
> it all, especially if I feel like it has been a day or two since
> I really embarrassed myself.

Okay, try this:

ftp://cow.physics.wisc.edu/pub/craigm/spiky_data.sav

It's the cumulative sum of normally distributed random deviates, so it
has lots of peaks and valleys to practice on.

Personally, I was responding to the challenge that J.D. put forth.
I've never used this snippet "for real." I just did it today. I said
that smoothing might be more appropriate for real life situations
because real-life data often has noise. My algorithm does not really
tolerate noise.

Peak finding has obvious uses. Need I say more? I personally don't
do too much of it. I do have time series with peaks, but I know where
to expect the peaks so I can just fit an amplitude.

For a noisy signal with many potential (but unknown) peaks I would
probably perform a cross correlation between the signal and a
template, and then threshold. This prevents a single noisy point from
ruining an otherwise nice peak.

For a noisy signal with a single peak, an algorithm such as IDL's
GAUSSFIT(), or my own MPFITPEAK() might be worthwhile. Those two
algorithms are different; I assert mine is better :-)

Craig

MPFITPEAK is found at
http://cow.physics.wisc.edu/~craigm/idl/idl.html

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu

Subject: Re: Avoiding a for cicle
Posted by John-David T. Smith on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>
>>> Alright code slingers... new challenge... find location of all peaks in a region
>>> of n points (n odd), monotonically decreasing away from the peak. I.e. find
>>> peaks of width n.
>>>
>>
>> Since noone will take my challenge I'm forced to claim the prize for myself:
>>
>> wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)
>
> Slow down there whipper-snapper! Gotta let these things stew for a
> while. Though multiline, this is my best effort:
>
> dd = d(1:*)-d
> nh = (n-1)/2
> wh = where(convol((dd GT 0) AND (dd(nh:*) LT 0), bytarr(nh)+1, nh) EQ 1, ct)+1
>
> For the goobledy-gook impaired (aka DF :-),
> dd is the first difference of the data
> nh is the half-width of the peak
> (dd GT 0) AND (dd(nh:*) LT 0) locates up-going followed by down-going points
> convol(...) locates runs of length nh
>
> This one does exactly what was requested, which I'm not sure of about
> your solution, J.D. On the other hand, your solution may be more
> physically meaningful since it involves smoothing.
>

Nice entry Craig. But unfortunatly it doesn't *alway* do exactly what was requested. It works fine for n=5, but for n>5 (7,9,...), the index is off. Part of the reason is in the way convol works. For n=5, nh=2, and convol subscripts are left of center (t+i-1 for i=0,1). For n=7, nh=3, and convol subscripts are (t+i-1 for i=0,1,2), which is centered. For n=9, nh=4, and you again have (t+i-2 for i=0,1,2,3), left of center again... and so on. The additional complication is that you are finding the center of the *wrapped* up-going and down-going entries... the center of the peak is offset from this by (nh+1)/2 (which happens to equal 1 when n=5 or n=3). Adding this rather than 1

to the returned indices would make it correct. Here's an example for n=9 (^=up-goin, v=down-going):

```
 | |
^^^^vvvv
 | |
```

You want the second marked location, the first down-going point. Convol will instead return the first marked location, so you have to offset by the half-width with the convol centering taken into account. Offsetting by 1 as you have done delivers the point between the two marked.

Now I'll explain my solution, which does indeed produce indentical results to yours (after being fixed as above, and except for the null case -- yours always returns (nh-1)/2, instead of -1, though you could test the count to avoid this).

It only works for n>=5 (but for n=3 you can use the first half of it as posted originally -- I left out the test on n which would select between them). Let's examine the parts:

d gt ((m=median(d,3)))

This is the solution to the original n=3 case, modified to assign m to the median 3 result. "d" is the data in which you are finding peaks, and n is the odd peak width you are trying to find.

smooth((d eq m)*(n-2),n-2) eq n-3

This looks like it must be smoothing the data somehow. It is not. Consider first the term (d eq m). Just as (d gt m) indicates a local 3-point maximum, (d eq m) indicates a local 3 point line. I.e., the point has one neighbor greater and one neighbor less than it. It lives on a "slope", not on a peak (or valley). So, what is the definition of an n point peak? It has a central point which is a 3-pt peak, surrounded on either side by n-3 points which are 3-pt slopes. Here is a picture demonstrating this for an n=7 peak:

```
  *     ^
 * *   / \
* *  /   \
*   *
```

The first is the data, the second indicates the kind of data, peak or slope. So, now we can consider the full smooth() expression. Smooth(), as you know, takes rolling averages. But averages are really just sums. We would like a way to find all 3-pt peaks, with n-3 3-pt "slopes" surrounding it. We can do a moving count of such slopes with smooth! (This is a trick I use all the time...

boxcar counting.) To avoid integer truncation of the average, we first multiply the boolean vector of slopes by (n-2). E.g. if you had 11011 in the above n=7 example of (d eq m), you'd smooth 55055 instead, with width 5, to find the value 4 at the central position. This is just the number we wanted! So, putting it all together, we want a single 3-pt peak, with n-3 3-pt slopes around it. We've seen how to find peaks and slopes, and we can count slopes with smooth. So we have all the ingredients.

"Aha," you might say,"but I have found the critical flaw in your reasoning... (d eq m) detects not a single kind of slope, but two kinds -- both / and \. To ensure a peak, you need only /'s before and only \'s after the peak! Any other combination is not an n point peak!"

A seemingly bullet-proof argument. But think carefully... If we know a point is a 3 point peak, an adjacent slope to the left is necessarily a /, and an adjacent slope to the right is necessarily a \, since part of the slope includes the peak itself! Moving two away from the peak, any slope adjacent to a \ slope is itself a \ slope, by a similar argument, and any slope adjacent to a / is itself a / slope. You can't switch slopes without going through a valley or a peak! So this test is sufficient to find them all.

To try this on some data, just generate some random numbers:

```
IDL> d=randomu(sd,1000)
IDL> n=7
IDL> wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)
```

To find all 7 point peaks in a stream of random numbers. There will be a few. You can examine them either one at a time, or by constructing a flag array to print side by side. Here's an example of the data for one peak:

```
IDL> print,wh
        38       344       412       706       906
IDL> print,rotate(d[wh[0]-n/2:wh[0]+n/2],1)
    0.271930
    0.404679
    0.460323
    0.787722
    0.432555
    0.411609
   0.0525138
```

You can see the central peak at 0.787722. The numbers of such peaks in random data decreases rapidly with n... try finding a 13 point peak... you'll need 10 million points before you have a good chance of getting some!

As far as real applications, you would probably not use this for spectral lines, which often don't follow the strict rules of monotonically decreasing from a central value. Instead you'd convolve with a gaussian kernel of appropriate

width, or do something similar. But I'm sure there are instances when finding
"real" peaks is necessary.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Avoiding a for cicle
Posted by davidf on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

> "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>
>>> Alright code slingers... new challenge... find location of all peaks in a region
>>> of n points (n odd), monotonically decreasing away from the peak. I.e. find
>>> peaks of width n.
>>>
>>
>> Since noone will take my challenge I'm forced to claim the prize for myself:
>>
>> wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)
>
> Slow down there whipper-snapper! Gotta let these things stew for a
> while. Though multiline, this is my best effort:
>
> dd = d(1:*)-d
> nh = (n-1)/2
> wh = where(convol((dd GT 0) AND (dd(nh:*) LT 0), bytarr(nh)+1, nh) EQ 1, ct)+1
>
> For the goobledy-gook impaired (aka DF :-),
>   dd is the first difference of the data
>   nh is the half-width of the peak
>   (dd GT 0) AND (dd(nh:*) LT 0) locates up-going followed by down-going points
>   convol(...) locates runs of length nh
>
> This one does exactly what was requested, which I'm not sure of about
> your solution, J.D. On the other hand, your solution may be more
> physically meaningful since it involves smoothing.

Alright, now that Craig has oriented me a little bit,
I find that I, uh..., have a *need* for this sort of thing. :-)

I presume you gentlemen are testing these little theories of yours
on a test data set. Could you supply such a data set for the
rest of us to fool around with? And if you gave us just a little
hint about how such a thing might be useful to *you*, that might
help too. I might even take a stab at writing an article about
it all, especially if I feel like it has been a day or two since
I really embarrassed myself.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by davidf on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

> For the goobledy-gook impaired (aka DF :-),

Bless you, Craig. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by Craig Markwardt on Wed, 12 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

"J.D. Smith" <jdsmith@astro.cornell.edu> writes:

>> Alright code slingers... new challenge... find location of all peaks in a region
>> of n points (n odd), monotonically decreasing away from the peak. I.e. find
>> peaks of width n.
>>
>
> Since noone will take my challenge I'm forced to claim the prize for myself:
>
> wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)

Slow down there whipper-snapper! Gotta let these things stew for a
while. Though multiline, this is my best effort:

```
dd = d(1:*)-d
nh = (n-1)/2
wh = where(convol((dd GT 0) AND (dd(nh:*) LT 0), bytarr(nh)+1, nh) EQ 1, ct)+1
```

For the goobledy-gook impaired (aka DF :-),
dd is the first difference of the data
nh is the half-width of the peak
(dd GT 0) AND (dd(nh:*) LT 0) locates up-going followed by down-going points
convol(...) locates runs of length nh

This one does exactly what was requested, which I'm not sure of about
your solution, J.D. On the other hand, your solution may be more
physically meaningful since it involves smoothing.

Craig

--

---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------