Subject: Local Maxima of 2D array
Posted by robintw on Tue, 19 Jan 2010 17:48:11 GMT
View Forum Message <> Reply to Message

Hi,

Another question from me I'm afraid. I'm trying to implement a routine
which needs to be able to calculate the local maxima of a small window
moved across an array. That is, I have a large array and I will need to
move a small 3x3 array across it, each time working out what the maximum
value of that array is and storing its index (or selecting it in some
other way).

I've investigated various methods for doing this, including the dilate
method, but I can't seem to get them to work properly.

Is there any good (as in fast, efficient and elegant) way of doing this,
or will I be reduced to using for loops and lots of IF statements?

Best regards,

Robin
University of Southampton

Subject: Re: Local Maxima of 2D array
Posted by rogass on Thu, 21 Jan 2010 14:55:02 GMT
View Forum Message <> Reply to Message

On 19 Jan., 18:48, Robin Wilson <r.t.wil...@rmplc.co.uk> wrote:
> Hi,
>
> Another question from me I'm afraid. I'm trying to implement a routine
> which needs to be able to calculate the local maxima of a small window
> moved across an array. That is, I have a large array and I will need to
> move a small 3x3 array across it, each time working out what the maximum
> value of that array is and storing its index (or selecting it in some
> other way).
>
> I've investigated various methods for doing this, including the dilate
> method, but I can't seem to get them to work properly.
>
> Is there any good (as in fast, efficient and elegant) way of doing this,
> or will I be reduced to using for loops and lots of IF statements?
>
> Best regards,
>
> Robin

> University of Southampton

Dear Robin,
did you tried to reform that array to 3D and to find the MAXima and
their indices together with the keyword DIMENSION=3? Don't forget that
REFORM 'forms' rowwise.

Cheers

CR

Subject: Re: Local Maxima of 2D array
Posted by robintw on Thu, 21 Jan 2010 17:39:23 GMT

View Forum Message <> Reply to Message

> Dear Robin,
> did you tried to reform that array to 3D and to find the MAXima and
> their indices together with the keyword DIMENSION=3? Don't forget that
> REFORM 'forms' rowwise.
>
> Cheers
>
> CR

Hi Chris,

Thank you very much for your suggestion. I have looked at the
documentation for the REFORM function, but I'm not sure how to reform
the array to a suitable 3D form so that MAX will work with the
dimension=3 keyword. Could you provide some more details?

Best regards,

Robin Wilson
University of Southampton

Subject: Re: Local Maxima of 2D array
Posted by rogass on Fri, 22 Jan 2010 09:56:33 GMT

View Forum Message <> Reply to Message

On 21 Jan., 18:39, Robin Wilson <r.t.wil...@rmplc.co.uk> wrote:
>> Dear Robin,
>> did you tried to reform that array to 3D and to find the MAXima and
>> their indices together with the keyword DIMENSION=3? Don't forget that
>> REFORM 'forms' rowwise.

>
>> Cheers
>
>> CR
>
> Hi Chris,
>
> Thank you very much for your suggestion. I have looked at the
> documentation for the REFORM function, but I'm not sure how to reform
> the array to a suitable 3D form so that MAX will work with the
> dimension=3 keyword. Could you provide some more details?
>
> Best regards,
>
> Robin Wilson
> University of Southampton

Dear Robin,

basically the code without a loop could be:

```
function cr_get_windowed_extrema,b,sx_k,sy_k

 sk  = long(sx_k)*long(sy_k)
 sz  = size(b,/dimensions)
 sm  = long(sz[0])*long(sz[1])
 ind= (reform((transpose(lindgen(sz[0],sz[1])))[*],sx_k,sm/sx_k))
[0:sy_k-1,*]
 mins=  min(b[(reform((transpose((transpose(rebin(ind,sx_k,sm/sy_k,s y_k)
+$
              rebin(lindgen(1,1,sy_k),sx_k,sm/sx_k,sy_k),[0,2,1])),
[1,0,2])),sk,sm/sx_k))],$
              minind,max=maxs,subscript_max=maxind,dimension=1)
 ind2=(lindgen(sx_k,sm/sk))[*,0:*:sy_k]
 return, {mins:mins[ind2],minind:minind[ind2],maxs:maxs
[ind2],maxind:maxind[ind2]}
end
```

As potential output I got:

```
IDL> b=randomn(seed,9,9)
IDL> c=cr_get_windowed_extrema(b,3,3)
IDL> print,b
   -0.232820     -1.81190     -1.79086   -0.0838641     -1.42229
-0.569596 -0.000931759     0.197937     0.203128
   -0.742161     -1.04460     0.286660      1.59126     -1.18528
1.11088     -1.17374     -1.51570     0.156324
    0.265435     -1.02502    -0.232129     0.259060    -0.825678
```

```
-0.386492     0.275219    -0.886818    -0.210116
      1.20696    0.0987463     -1.22906    -0.155326      1.27177
-1.25504     0.650159    -0.864291    -0.915809
     0.207192    -0.544278     -1.79930    0.0309544    -0.609460
-0.348675    -0.199986     0.518268     -1.03154
      1.35320      1.08140  -0.00415816    -0.822823    -0.570877
-1.01163     -1.01084      1.87093     -1.31978
    -0.486999     0.565098     0.140825    0.0224620     0.851600
0.922738    -0.779988     0.251917     0.834798
     -1.06734      1.14913    -0.539062    -0.584468    -0.426683
0.869110     0.384573      1.50669     0.350647
     0.478418     0.458704      1.72066      1.48684    -0.250672
0.920115    -0.324874     -1.49407   -0.0624892
IDL> print,c.mins
     -1.81190     -1.79930     -1.06734
     -1.42229     -1.25504    -0.584468
     -1.51570     -1.31978     -1.49407
IDL> print,c.maxs
     0.286660      1.35320      1.72066
      1.59126      1.27177      1.48684
     0.275219      1.87093      1.50669
IDL> print,c.minind
                     1                    14                    21
                    82                    92                   102
                   166                   179                   187
IDL> print,c.maxind
                     5                    15                    26
                    84                    91                   105
                   168                   178                   184
```

Hope, it works for you :) Maybe there are some unnecessary computations, so you might optimize the code...

Cheers

CR

Subject: Re: Local Maxima of 2D array
Posted by Yngvar Larsen on Mon, 25 Jan 2010 15:55:22 GMT

View Forum Message <> Reply to Message

On Jan 19, 6:48 pm, Robin Wilson <r.t.wil...@rmplc.co.uk> wrote:
> Hi,
>
> Another question from me I'm afraid. I'm trying to implement a routine
> which needs to be able to calculate the local maxima of a small window
> moved across an array. That is, I have a large array and I will need to

> move a small 3x3 array across it, each time working out what the maximum
> value of that array is and storing its index (or selecting it in some
> other way).
>
> I've investigated various methods for doing this, including the dilate
> method, but I can't seem to get them to work properly.
>
> Is there any good (as in fast, efficient and elegant) way of doing this,
> or will I be reduced to using for loops and lots of IF statements?

Some kind of FOR loop is unavoidable, I think.

Depending of the size of your array, this code will do (most of) the
job efficiently. Elegant? Well...

```
x = [-1,  0,  1, -1, 0, 1, -1, 0, 1]
y = [-1, -1, -1,  0, 0, 0,  1, 1, 1]

; Or in general for a sliding (Kx x Ky) window
;x = lindgen(Kx)-Kx/2
;y = lindgen(Ky)-Ky/2
;x = (x[*,lindgen(Ky)])[*]
;y = (transpose(y[*,lindgen(Kx)]))[*]

sliding_3x3_max = shift(array, x[0], y[0])
for ii=1, 8 do sliding_3x3_max >= shift(array, x[ii], y[ii])
```

Note that the border case isn't handled. This is left as an exercise
for the reader :) Also, if you really need the index for each maximum
instead of the value, you must do a bit more work inside the loop.

My experience is that this method works well for operations on sliding
windows up to about 15x15, but for larger windows, the cost of the
(quite fast) SHIFT function starts to dominate when compared to the
straightforward double loop approach.

--
Yngvar

Subject: Re: Local Maxima of 2D array
Posted by karo on Wed, 27 Jan 2010 05:00:54 GMT

View Forum Message <> Reply to Message

If you have data in int or uint you could use the dilate procedure with a
3x3 array as structuring element. Dilate calculates the max and erode the
min! Have care with keywords, /GREY will be nessecary

Regards
Karsten

Am 19.01.10 18:48 schrieb "Robin Wilson" unter <r.t.wilson@rmplc.co.uk> in i6CdnU6cXJD-bcjWnZ2dnUVZ8vudnZ2d@bt.com:

> Hi,
>
> Another question from me I'm afraid. I'm trying to implement a routine
> which needs to be able to calculate the local maxima of a small window
> moved across an array. That is, I have a large array and I will need to
> move a small 3x3 array across it, each time working out what the maximum
> value of that array is and storing its index (or selecting it in some
> other way).
>
> I've investigated various methods for doing this, including the dilate
> method, but I can't seem to get them to work properly.
>
> Is there any good (as in fast, efficient and elegant) way of doing this,
> or will I be reduced to using for loops and lots of IF statements?
>
> Best regards,
>
> Robin
> University of Southampton