
Subject: Re: Avoiding loop stats
Posted by [JD Smith](#) on Fri, 19 Jan 2007 18:11:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 19 Jan 2007 09:43:59 -0800, yp wrote:

> IDL Gurus,
> There is perhaps a smart solution to this problem, but I could not
> figure out.
> I have a series of EO images (2D) stacked over time which makes the
> data a 3D array of [4000, 2000, 900] i.e., [lon,lat,time]
> I need to compute various statistical parameters at each pixel over
> time and produce each of them as [4000,2000] array.
>
>
> for i=0,4000L-1 do for j=0,2000L-1 do data_st(i,j)=st_func(data(i,j,*))
>
> where, data=FLTARR[4000,2000,900]
> data_st is the output from a function 'st_func' which works with vector
> data only.
>
> Is there a way to do this avoiding the 4000x2000 loop? It is painfully
> slow on windows.

Yes, recode st_func to work on the full data cube at once. Sometimes this is easier said than done. Let's imagine your st_func just calculates the standard deviation. Unfortunately, IDL's built-in statistics functions are almost all array-unaware, but some things are easy to do "by hand":

```
s=size(data,/DIMEN)
mean=total(data,3)/s[2]
stddev=sqrt(total((data-rebin(mean,s))^2)/(s[2]-1))
```

Higher moments could be built as well. The "threadable" or array-aware statistics/math functions (since IDL v5.6, anyway), are MIN, MAX, MEDIAN, TOTAL, PRODUCT, SMOOTH, and CONVOL (any other I'm missing?).

ITTVIS could invest a small amount of effort to improve this state of affairs. For instance, it would be trivial to rewrite MOMENT.PRO to take a DIMENSION keyword, such that VARIANCE, STDDEV, MEAN, SKEWNESS, KURTOSIS could all be array-aware.

JD

Subject: Re: Avoiding loop stats
Posted by news.qwest.net on Fri, 19 Jan 2007 20:00:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

"yp" <Yaswant.Pradhan@gmail.com> wrote in message
news:1169228639.255277.164120@38g2000cwa.googlegroups.com...
> IDL Gurus,
...
> Is there a way to do this avoiding the 4000x2000 loop? It is painfully
> slow on windows.

no, Fortran would be a better solution for a problem like this.
I bet orders of magnitude faster.

Subject: Re: Avoiding loop stats
Posted by [Foldy Lajos](#) on Fri, 19 Jan 2007 20:26:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 19 Jan 2007, yp wrote:

> IDL Gurus,
> There is perhaps a smart solution to this problem, but I could not
> figure out.
> I have a series of EO images (2D) stacked over time which makes the
> data a 3D array of [4000, 2000, 900] i.e., [lon,lat,time]
> I need to compute various statistical parameters at each pixel over
> time and produce each of them as [4000,2000] array.
>
>
> for i=0,4000L-1 do for j=0,2000L-1 do data_st(i,j)=st_func(data(i,j,*))
>
> where, data=FLTARR[4000,2000,900]
> data_st is the output from a function 'st_func' which works with vector
> data only.
>
> Is there a way to do this avoiding the 4000x2000 loop? It is painfully
> slow on windows.
> thanks in advance
>

Some speedup can be achieved with optimized memory access (if you have
enough memory for two copies of data):

```
temp=transpose(data, [2,0,1])  
for j=0l,2000l-1 do for i=0l,4000l-1 do data_st[i,j]=st_func(temp[*i,j])
```

regards,
lajos

Subject: Re: Avoiding loop stats
Posted by [yp](#) on Sat, 20 Jan 2007 03:07:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi JD,
I have already written only few the simplest stats functions (stdev, covar, etc) inspired by NASA average.pro function to tackle the dimension issue. I could not spend much time to write e.g., a full regression analysis. I was expecting a clever syntactic way to play with the IDL in-built functions. But it is good to know that I must rewrite the functions.
Thanks for your suggestion,
yas

On Jan 19, 6:11 pm, JD Smith <jdsm...@as.arizona.edu> wrote:
> Yes, recode st_func to work on the full data cube at once. Sometimes
> this is easier said than done. Let's imagine your st_func just
> calculates the standard deviation. Unfortunately, IDL's built-in
> statistics functions are almost all array-unaware, but some things are
> easy to do "by hand":
>
> s=size(data,/DIMEN)
> mean=total(data,3)/s[2]
> stddev=sqrt(total((data-rebin(mean,s))^2)/(s[2]-1))
>
> Higher moments could be built as well. The "threadable" or
> array-aware statistics/math functions (since IDL v5.6, anyway), are
> MIN, MAX, MEDIAN, TOTAL, PRODUCT, SMOOTH, and CONVOL (any other I'm
> missing?).
>
> ITTVIS could invest a small amount of effort to improve this state of
> affairs. For instance, it would be trivial to rewrite MOMENT.PRO to
> take a DIMENSION keyword, such that VARIANCE, STDDEV, MEAN, SKEWNESS,
> KURTOSIS could all be array-aware.
>
> JD- Hide quoted text -- Show quoted text -

Subject: Re: Avoiding loop stats
Posted by [MarioIncandenza](#) on Mon, 22 Jan 2007 16:29:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm missing something. You are working with a FLTARR(4000,2000,900) on Windows? That is >10x the size of the largest array I can create in IDL under Windows or Linux. Is this 64-bit IDL? for Windows?

yp wrote:

> IDL Gurus,
> There is perhaps a smart solution to this problem, but I could not
> figure out.
> I have a series of EO images (2D) stacked over time which makes the
> data a 3D array of [4000, 2000, 900] i.e., [lon,lat,time]
> I need to compute various statistical parameters at each pixel over
> time and produce each of them as [4000,2000] array.
>
>
> for i=0,4000L-1 do for j=0,2000L-1 do data_st(i,j)=st_func(data(i,j,*))
>
> where, data=FLTARR[4000,2000,900]
> data_st is the output from a function 'st_func' which works with vector
> data only.
>
> Is there a way to do this avoiding the 4000x2000 loop? It is painfully
> slow on windows.
> thanks in advance

Subject: Re: Avoiding loop stats
Posted by [Jean H.](#) on Mon, 22 Jan 2007 17:20:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ed Hyer wrote:

> I'm missing something. You are working with a FLTARR(4000,2000,900) on
> Windows? That is >10x the size of the largest array I can create in IDL
> under Windows or Linux. Is this 64-bit IDL? for Windows?

agreed...

4000 * 2000 * 900 * 32bits = 26.82 Gb and the max memory one
could have under windows is 4 Gb..

but you could use an associated variable in this case..

Jean

>
> yp wrote:
>
>> IDL Gurus,
>> There is perhaps a smart solution to this problem, but I could not

```
>> figure out.  
>> I have a series of EO images (2D) stacked over time which makes the  
>> data a 3D array of [4000, 2000, 900] i.e., [lon,lat,time]  
>> I need to compute various statistical parameters at each pixel over  
>> time and produce each of them as [4000,2000] array.  
>>  
>>  
>> for i=0,4000L-1 do for j=0,2000L-1 do data_st(i,j)=st_func(data(i,j,*))  
>>  
>> where, data=FLTARR[4000,2000,900]  
>> data_st is the output from a function 'st_func' which works with vector  
>> data only.  
>>  
>> Is there a way to do this avoiding the 4000x2000 loop? It is painfully  
>> slow on windows.  
>> thanks in advance  
>  
>
```
