Subject: histogram & reverse_indices
Posted by Ken Mankoff on Tue, 09 Apr 2002 19:15:20 GMT
View Forum Message <> Reply to Message

Hi,

As you may have guessed from the subject, I have a question about histogram and reverse indices...

I have a 2D array made up of n quadruplets. Ex:
```
array =[[1,1,1,2], $
 [1,1,1,1], $
 [3,4,3,2], $
 [3,3,0,0], $
 [5,5,0,5]]
```

I want my algorithm to do the following: Return the index of all the quadruplets that have at least 3 out of 4 numbers equal to each other (i.e. for the above array, it should return [0,1,4].

I can do it in a for-loop as follows:

```
for i=0,n_elements(array[0,*])-1 do begin
  quad = array[*,i]
  hist = histogram( quad )
  hist = hist[ where( hist ne 0 ) ]
  if ( max( hist ) gt 3 then print, 'good' else print, 'bad'
endfor
```

But I think there is a way to do this without a for loop. Either using reverse_indices, or where(), I just cannot see it. Can you?

  -k.

--
Kenneth Mankoff
LASP://303.492.3264
http://lasp.colorado.edu/~mankoff/
http://lasp.colorado.edu/snoe/
http://lasp.colorado.edu/mars/
http://lasp.colorado.edu/marsrobot/

Subject: Re: histogram & reverse_indices
Posted by Wayne Landsman on Tue, 09 Apr 2002 20:47:45 GMT
View Forum Message <> Reply to Message

Ken Mankoff wrote:

>
> I have a 2D array made up of n quadruplets. Ex:
> array =[[1,1,1,2], $
> [1,1,1,1], $
> [3,4,3,2], $
> [3,3,0,0], $
> [5,5,0,5]]
>
> I want my algorithm to do the following: Return the index of all the
> quadruplets that have at least 3 out of 4 numbers equal to each other
> (i.e. for the above array, it should return [0,1,4]

Here's a non-loop solution for the specific case, although it is a solution that is difficult to generalize, and which may be less understandable and slower than simply using a loop.

The idea is that if 3 out of 4 numbers are equal to each other, then that number is either the minimum or the maximum of the quadruplet. So we first get the min and max of each quadruplet.

```
   amin = min(array,dimen=1,max=amax)        ;V5.5 needed
```

Now reform/rebin the min and max vectors into a 2d arrays

```
amax = rebin(reform(amax,1,5),4,5)
amin  = rebin(reform(amin,1,5),4,5)
```

Now find which values in the array are equal to either the minimum or the maximum. Total along rows
to determine if 3 or more values in a quadruplet meet this condition:

```
print,where( (total((array EQ amin),1) GE 3) or $
                  (total((array EQ amax),1) GE 3))
```

---> [0,1,4]

--Wayne
landsman@mpb.gsfc.nasa.gov

Subject: Re: histogram & reverse_indices
Posted by Craig Markwardt on Wed, 10 Apr 2002 02:35:23 GMT
View Forum Message <> Reply to Message

Wayne Landsman <landsman@mpb.gsfc.nasa.gov> writes:

> Ken Mankoff wrote:

>
...
>> I want my algorithm to do the following: Return the index of all the
>> quadruplets that have at least 3 out of 4 numbers equal to each other
>> (i.e. for the above array, it should return [0,1,4]
>
> Here's a non-loop solution for the specific case, although it is a
> solution that is difficult to generalize, and which may be less
> understandable and slower than simply using a loop.

Ooof, Wayne beat me to the punch. It looks like a good technique.

Craig

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

Subject: Re: histogram & reverse_indices
Posted by the_cacc on Wed, 10 Apr 2002 10:02:37 GMT
View Forum Message <> Reply to Message

Only in IDL could we have such FOR paranoia... May the FORs not be with you.

Subject: Re: histogram & reverse_indices
Posted by Ken Mankoff on Wed, 10 Apr 2002 14:26:06 GMT
View Forum Message <> Reply to Message

On 9 Apr 2002, Craig Markwardt wrote:

> Wayne Landsman <landsman@mpb.gsfc.nasa.gov> writes:
>
>> Ken Mankoff wrote:
>>
> ...
>>> I want my algorithm to do the following: Return the index of all the
>>> quadruplets that have at least 3 out of 4 numbers equal to each other
>>> (i.e. for the above array, it should return [0,1,4]
>>
>> Here's a non-loop solution for the specific case, although it is a
>> solution that is difficult to generalize, and which may be less
>> understandable and slower than simply using a loop.

>
> Ooof, Wayne beat me to the punch. It looks like a good technique.
>

Yep, that works. Thanks!

-k.
