Subject: Re: majority voting
Posted by ben.bighair on Wed, 11 Feb 2009 22:03:59 GMT
View Forum Message <> Reply to Message

On Feb 11, 4:53 pm, Mort Canty <m.ca...@fz-juelich.de> wrote:
> ben.bighair schrieb:
>
>
>
>> On Feb 11, 11:14 am, mort canty <m.ca...@fz-juelich.de> wrote:
>>> Hi all,
>
>>> Given a 2-D array such as
>
>>> 0 1 1 2 1
>>> 0 2 1 1 1
>>> 1 0 2 2 1
>
>>> where the entries are labels, the columns represent items and the rows
>>> are voters, I want a IDL function that returns the majority vote labels.
>>> So here I should get
>
>>> 0 ? 1 2 1
>
>>> as output, where ? = "don't care". There must _not_ be a loop over
>>> columns. I've got a clumsy solution, but I'm sure there's an elegant one
>>> somewhere?
>
>> Hi,
>
>> This is incomplete as it doesn't flag the "don't care" crowd. I can't
>> noodle that part out without column looping. Looping would make it
>> easy to use something like...
>
>> for i = 0, ncol-1 do dontCare[i] = ARRAY_EQUAL(votes[i,*],votes[i,0])
>
>> but by your rules, that is out of bounds.
>
>> ***BEGIN
>> x=[[0,1,1,2,1],$
>> [0,2,1,1,1],$
>> [1,0,2,2,1]]
>> sz = SIZE(x, /DIM)
>> votes = [[TOTAL(x EQ 0, 2)],$
>> [TOTAL(x EQ 1, 2)], $
>> [TOTAL(x EQ 2, 2)]]
>> mx = MAX(votes, mxIdx,dim = 2)
>> majority = (array_indices(sz, mxIdx, /dim))[1,*]

```
>>  print, majority
>>  ***END
>
>>  Cheers,
>>  Ben
>
>  Thanks Ben. What I meant by "don't care" is that I don't care which of
>  the labels that got equal votes is output. I think my solution is
>  essentially the same as yours, certainly not more elegant:
>
>  function majority_vote, A, num_labels
>      n = n_elements(A[*,0])
>      B = intarr(n,num_labels)
>      for i=0,num_labels-1 do begin
>         C = A*0
>         idx = where(A eq i,count)
>         if count gt 0 then C[idx] = 1
>         B[*,i] = total(C,2)
>      endfor
>      void = max(B,labels,dimension=2)
>      return, labels/n
>  end
>
>  I was probably hoping for some HISTOGRAM magic :-)
>  Mort
```

Oooo. Histogram... well my mojo just isn't working that way today.
But deep inside Jean's addition is sort_nd (I had never heard of it
before - slick!) which has a histogram *with* reverse indices. Double
the mojo!

Cheers,
Ben
