Subject: Re: majority voting
Posted by Mort Canty on Thu, 12 Feb 2009 15:45:09 GMT
View Forum Message <> Reply to Message

Allan Whiteford schrieb:
> mort canty 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?
>>
>> Cheers,
>>
>> Mort
>
> Hi Mort,
>
> It might be less efficient than JD's histogram solution (I didn't check)
> but the following also fits the problem specification:
>
> x=[[0,1,1,2,1],$
> [0,2,1,1,1],$
> [1,0,2,2,1]]
>
> voters=(size(x,/dim))[1]
> items=(size(x,/dim))[0]
> max_label=max(x)+1
>
> f=intarr(max_label,items)
> ++f[max_label*(indgen(voters*items) / voters)+ $
> reform(transpose(x),voters*items)]
> junk=max(f,idx,dim=1)
> print,idx - max_label*findgen(items)
>
> Note that the above solution will also blow up when you end up with

> sparse arrays (e.g. if you have someone voting for label 1000000 then f
> will end up being an items x 1000000 array even if nobody votes for any
> labels between 3 and 1000000).
>
> I think all the discussions on finding the mode (either in 1D or nD)
> probably pre-dated the ++ operator. It could be that using the
> vectorised ++ operator is a better way to do it - I doubt it though,
> normally if histogram can do something then histogram will be the best
> way! You'd also need to introduce a clumsy offset to deal with negative
> selections (Not an issue for you here but would be if finding the mode
> in a more general way).
>
> It would make David's 1D example from his webpage into something like this:
>
> array = [1, 1, 2 , 4, 1, 3, 3, 2, 4, 5, 3, 2, 2, 1, 2, 6,-3]
> f=intarr(max(array)-min(array)+1)
> f[array-min(array)]++
> junk=max(f,idx)
> mode=idx + min(array)
> print,mode
>
> again, with no idea on what would be more efficient. If you're doing
> analysis on measurements (typically non-integers) then you'd need to
> invoke histogram anyway to bin them before trying to find the mode.
>
> Thanks,
>
> Allan

Hi Allan,

Thanks! Not only have I learned that something called vectorized ++
exists, but that it bumps the indexed value multiple times if that index
is repeated. Live and learn! But where the hell is all that on the IDL Help?

Anyway, what I had in mind was trying to program an ensemble image
classifier, so that the items are rows of pixels (lots and lots), the
labels are land cover classes (contiguous small integers) and the voters
are classifiers (e.g. neural networks, also not too many). Hence the
wish to avoid the loop over items. I certainly got my money's worth :-)

Mort