
Subject: Re: Minimum between two arrays; no, not just Min(A-B)

Posted by [JD Smith](#) on Tue, 18 Sep 2007 22:53:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Tue, 18 Sep 2007 07:19:00 -0700, leatherback wrote:

> Hi All,
>
> I am working with time series data, and image scenes. For each point in my
> time series I try to find the correct satellite scene to use. So I have
> two arrays, one with dates of observations, one with image scene dates.
>
> These dates are in long integer (Unix timestamps).
>
> I *could* of course replicate my image scenes array to the length of the
> timeseries array, calculate the difference for each date, and find out
> where it is, something along these lines:
>
> [/color]
[color=blue]> datelist = rebin(dates, n_elements(dates), n_elements(fixes.timestamp))[/color]
[color=blue]> [/color]
[color=blue]> for j=0, n_elements(dates)-1 do begin[/color]
[color=blue]> datelist[j,*] = datelist[j,*]-fixes.timestamp[/color]
[color=blue]> endfor[/color]
[color=blue]> [/color]
>
> Or even rebin both arrays, and subtract them. En then loop through them
> using the MIN / WHERE functions to find out where the minimum differences
> are.
>
> However, this does not feel right. Besides that.. the timeseries is
> potentially 100,000+ observations. Add to that a few hundred image dates,
> and you are filling your memory for something quite simple. Anybody here
> have a good idea how to vectorize this problem?

If they are in sorted order, you can use VALUE_LOCATE, which is happy
to search for the location of many dates within many satellite
timestamps, all at once. Then select which ever of the two is closer,
like so:

```
IDL> nv=100
IDL> obs=randomu(sd,nv) & sat=randomu(sd,nv) & sat=sat[sort(sat)]
IDL> v=value_locate(sat,obs)
IDL> m=min(abs(sat[[1#v,1#v+1]]-rebin(1#obs,2,nv)),pos,DIMENSION= 1)
IDL> v= nv-1 < (v + pos mod 2) > 0
IDL> print,mean(abs(sat[v]-obs))
```

JD
