Subject: Re: Find Closest Coincident Measurements In Time And Space Between Two Data Sets
Posted by Jeremy Bailin on Thu, 16 Oct 2008 01:32:25 GMT
View Forum Message <> Reply to Message

On Oct 13, 8:14 am, "|Rob|" <r08...@gmail.com> wrote:

> I was hoping that somebody could help and apply some magic to this
> problem.
>
> I have two sets of satellite measurement data and for dataset A want
> to find the value in dataset B that are closest to it in time and
> space.
>
> If that's easy enough to speed it it'd also be useful to find ALL
> values in databaset B that were within a certain spatial and temporal
> distance of each point in dataset A (for example, 5deg lat/lon and 2
> hours).
>
> As both datasets are quite large I have a feeling that this problem is
> quite similar to the one found herehttp://www.dfanning.com/code_tips/slowloops.html
> but I don't think I understand it quite well enough to solve and
> eliminate the FOR loops.
>
> My (working but slow) code is below.
>
> Any help would be great.
>
> Cheers
>
> ;datasetA structure contains lat, long, time and values
> ;datasetB has lat, long, time and value arrays
>
> count=0L
> final_lat=fltarr(100000)
> final_lon=fltarr(100000)
> final_datasetA=fltarr(100000)
> final_datasetB=fltarr(100000)
> final_time=strarr(100000)
> final_date=strarr(100000)
>
> ;value in hms of datasetB meas. either side of datasetA meas. time
> where time is still considered coincident
>
> time_margin=010000
>
> ;value in degrees of datasetB meas. either side of datasetA meas.
> location where location is still considered coincident
>

```
>  space_margin=2.  ;degrees lat/long
>
>  print, ‘Starting to loop through dataset A to find coincident dataset
>  B points'
>  FOR i=0L, n_elements(datasetA.lats)-1  DO BEGIN
>
>  ;find measurement in datasetB that is within time and space margin of
>  datasetA
>  coin=where(datasetB_hms GE datasetA.time[i]-time_margin AND
>  datasetB_hms LE datasetA.time[i]+time_margin AND $
>            datasetB_lat GE datasetA.lats[i]-space_margin AND
>  datasetB_lat LE datasetA.lats[i]+space_margin AND $
>            datasetB_lon GE datasetA.lons[i]-space_margin AND
>  datasetB_lon LE datasetA.lons[i]+space_margin)
>
>  ;account for no coincident matches between datasets
>  if coin[0] NE -1 THEN BEGIN
>  final_lat[count]=datasetA.lats[i]
>  final_lon[count]=datasetA.lons[i]
>  final_datasetA[count]=datasetA.value[i]
>
>  ;store mean of datasetB values that are coincident to datasetA
>  measurments
>  final_datasetB[count]=mean(datasetB_values[coin])
>  final_time[count]=datasetA.time[i]
>  final_date[count]=datasetA.date
>  count=count+1L
>  END
>  ENDFOR
>  print, 'Finished looping through datasetA data to find coincident
>  datasetB data'
```

While not the full solution, you might find WITHINSPHRAD useful:
http://astroconst.org/jbiu/jbiu-doc/astro/withinsphrad.html
within JBIU:
http://web.astroconst.org/jbiu

-Jeremy.