
Subject: Re: EOF analysis

Posted by [siumtesfai](#) on Wed, 12 Aug 2015 19:10:28 GMT

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

Hello all,

I am waiting for your suggestion . This is a follow up question

I followed David EOF analysis which worked for him. I have only changes data set from NCEP/NCAR . I used Geopotential Height . I am wondering why the variance of the first mode of my PC is higher (53 %) . Literature says PC1 explain around 20 % variance.

Please help if you can see any error the source code below

Best regards

On Tuesday, August 11, 2015 at 12:06:18 PM UTC-4, siumt...@gmail.com wrote:

> Dear All,

>

>

> I have tried to use EOF analysis on my data. Any suggestion on how to fix my code below using NCEP/NCAR Sea level pressure.

>

>

> Best regards

>

> ;=====

>

> filename=['ncep1']

>

> FOR imodel = 0 ,n_elements(filename)-1 do begin

>

> source ='F:\Data\CMIP5\geopot\1979-2012\zg_Amon_'

> file=source+filename(imodel)+'_r1i1p1_combine_za.sav'

>

> restore,file

> print,file

>

> check,time

>

>

> xt=where(time GE 1950. and time LT 2006)

> time=time(xt)

> lon=long

> xp=where(ref_press eq 1000)

> new_data1=reform(new_data1(*,*,xp,xt))

>

> ;=====

```

> ; Interpolate models to NCEP resolution (2.5 by 2.5)
> ; =====
>
> if imodel eq 0 then sst=fltarr(n_elements(filename),144,73,n_elements(time))
>
>
> FOR timeindx=0,n_elements(time)-1 do begin
>
>     ice=new_data1[*,*,timeindx]
>     nx = 144
>     ny = 73
>     slon = findgen(144)*2.5
>     slat = findgen(73)*2.5-90
>     x = Interpol(Findgen(N_Elements(lon)), lon, slon)
>     y = Interpol(Findgen(N_Elements(lat)), lat, slat)
>     xx = Rebin(x, nx, ny, /SAMPLE)
>     yy = Rebin(Reform(y, 1, ny), nx, ny, /SAMPLE)
>     newwind = INTERPOLATE(ice, xx, yy,missing=1e20)
>
>
>
>     sst(imodel,*,*,timeindx)=newwind
>
> ENDFOR ; timeindx loop
>
>
> lat=slat
> lon=slon
>
> ;=====
> ntime=n_elements(time)/12
>
> ; over NH 20N-90N
>
> xlat=where(lat GE 20 and lat LE 90)
> xlon=where(lon GE 0 and lon LE 360)
>
> tempdata=reform(sst(imodel,xlon,xlat,*))
>
>
> xmonth=where(time GE 1950 and time LT 2006)
> data=tempdata(*,*,xmonth)
> lon=lon(xlon)
> lat=lat(xlat)
>
> ; Calculate and apply cosine weighting to the values.
> dims = Size(data, /Dimensions)
> nlon = dims[0] & nlat = dims[1] & ntime = dims[2]

```

```

> dlon = Abs(lon[1]-lon[0])
> dlat = Abs(lat[1]-lat[0])
> weights = Sqrt(Cos((lat - dlat/2.) * !DtoR))
>
> fac=!pi/180.
>
> FOR k=0,nlon-1 do begin
>   FOR m=0,nlat-1 do begin
>     FOR j=0,ntime-1 DO begin
>
>       data[k,m,j] = data[k,m,j] * weights(m)
>
>     ENDFOR
>   ENDFOR
> ENDFOR
>
>
> data = Reform(data, nlon*nlat, ntime, /OVERWRITE)
> data_anomalies = data
>
> FOR jj=0,nlon*nlat-1 DO data_anomalies[jj,*] = data[jj,*] - Mean(data[jj,*])
>
> matrix = (1/ntime-1) * (Double(data_anomalies) ## Transpose(data_anomalies))
>
> LA_SVD, matrix, W, U, V
>
> dims = Size(data_anomalies, /Dimensions)
> eof1 = FltArr(dims[1], dims[0])
>
> FOR j=0,dims[1]-1 DO BEGIN
>   t = Transpose(data_anomalies) ## U[j,*]
>   eof1[j,*] = t / SQRT(Total(t^2))
> ENDFOR
>
> pc = FltArr(dims[1], dims[1])
>
> FOR j=0,dims[1]-1 DO pc[j,*] = data_anomalies ## eof1[j,*]
>
>
> percent_variance = W / TOTAL(W) * 100.0
>
> mode = 1
> theEOF = eof1[mode-1,*]
>
> theEOF = Reform(theEOF, nlon, nlat, /OVERWRITE)
>
> pctmp=reform(pc(0,*))
> pc1=fltarr(ntime)

```

```
>  
> FOR k=0,ntime-1 DO pc1[k] = pctmp[k] / stddev(pctmp) ; PC1 is the AO index  
>  
> AO=pc1  
>  
>  
> ENDFOR ; end of main loop  
>  
> END
```
