Subject: Re: Relative distances between vector elements and search for matches in other vector.
Posted by cgguido on Wed, 25 Apr 2012 15:03:35 GMT
View Forum Message <> Reply to Message

Also, I would also get nx=n_elements(xx) once and use it when I need it it.

Finally, your min for loop, I suspect one histogram could do it. but I am not sure what you're trying to do...

If you keep your loop then you could try this:

```
a = where(  abs(ratio-ratio2[jjj]) LT 0.05 ,count)
```

and also avoid appending to stel_dep adn stel_wave by greating them outside the loop and filling as needed. Then clip one at the end.

PS: histogram wise... I'd start with something approximately like:

```
nr=n_elements(ratio)
nr2=n_elements(ratio2)

bigratio=rebin(ratio,[nr,nr2],/sample)
bigratio2=transpose(rebin(ratio2,[nr2,nr], /sample))

now you have two arrays that are the same size, so you can do:
diff=bigratio-bigratio2

;good is now filled with 1s and 0s
good=abs(diff) LT 0.05

;you can count them by using DIMENSION with total:
count=total(good,2)

...
```

On Wednesday, April 25, 2012 9:03:00 AM UTC-5, Trifon Trifonov wrote:
> Hallo all
> I am new in IDL but I am increasing my level everyday and I am so happy that I discover IDL!!!
> I stuck very badly in one algorithm that I am developing for my Pipeline.
> I am searching for a routine or method to:
>
> 1st. define the relative distances between vector elements (e.g. distances between peaks in the spectrum(allready identified))

>
> 2nd. to identify the lines from another vector (bigger or smaller in array size)
> based on the criteria of the above (1st.)
>
> E.g.
>
> I have:
> RATIO_STLINES FLOAT = Array[21]
> 0.00000 -0.171038 -0.594733 -1.21922 -1.40923 -1.52994 -1.56168
-1.92495 -2.28294 -2.34803 -2.38444
> -3.15909 -3.46812 -3.61399 -3.97221 -4.09896 -4.27114 -4.43637
-5.69552 -6.05595 -6.23174
>
> RATIO_CAT FLOAT = Array[44]
> 0.00000 -0.209757 -0.249368 -0.930353 -1.05059 -1.21623 -1.21693
-1.74758 -1.75778 -1.77888 -1.78749
> -2.03075 -2.22571 -2.99081 -3.29190 -3.31060 -3.42393 -3.43413
-3.43844 -3.44364 -3.59758 -3.61648
> -3.76892 -3.80183 -3.92147 -4.26676 -4.53263 -5.02637 -5.31455
-5.35194 -5.52650 -5.59132 -5.75987
> -5.81128 -6.60440 -6.68362 -6.71653 -6.73373 -6.81295 -6.95529
-7.23017 -7.48834 -7.66608 -7.90535
>
> And the RATIO_CAT consist the lines from RATIO_STLINES (it is possible not all of them to be
in the RATIO_CAT)
>
> So RATIO_STLINES is lets say shifted respect to RATIO_CAT and I want to identify lines
based on their line ratios.
>
> I want to mention that I already made a working algorithm but I am 1000 % sure that the more
experienced people here can help me to make it much more elegant and efficient. Later on I have
a other problem on that I stuck but lets see can you help me on that and it is possible to solve it
my self... if not I will post again :(
>
>
> Here is the code:
>
> ;Finding the peak of each line on the spectrum:
>
> peak = peaks(1-chip,0.9) ; finding the peaks from JJohnson IDL.
> ddd = xaxis[peak] ;wavelengths of the finded lines.
> lines = chip[peak] ;depts of the finded lines.
> ;xx - HITRAN lines
> ;z1- VALD catalog wavelengths
> ;z2 - VALD catalog line depths
>
> ;Finding the relative distance between the HITRAN lines (nm):
> ratio=fltarr(n_elements(xx))

```
>
>  for j=0L, n_elements(xx)-1 do begin
>    ratio[j] =  xx[0] - xx[j]
>    endfor
>
>  ;Finding the relative distance between each line on the spectrum:
>  ratio2=fltarr(n_elements(ddd))
>
>  for jj=0L, n_elements(ddd)-1 do begin
>    ratio2[jj] =  ddd[0] - ddd[jj]
>  endfor
>
>  ;Finding the relative distance between the lines from the VALD catalog:
>  ratio_vald=fltarr(n_elements(z1))
>
>  for jjj=0L, n_elements(z1)-1 do begin
>    ratio_vald[jjj] =  z1[0] - z1[jjj]
>  endfor
>
>
>  ; Identify the telluric lines on the spectra. Only stellar lines will be left.
>  telluric_depts = fltarr(n_elements(xx))
>  telluric_waves = fltarr(n_elements(xx))
>  stel_dep = []
>  stel_wave = []
>  stel_ra = []
>  for jjj = 0L, n_elements(ratio2)-1 do begin
>    a = where((ratio GT (ratio2[jjj] - 0.05)) and (ratio LT (ratio2[jjj] + 0.05)),count)
>
>    if count gt 0d0 then begin
>    telluric_depts[a] = lines[jjj]
>    telluric_waves[a] = ddd[jjj]
>    endif
>
>    if count le 0d0 then begin
>    stel_dep =[stel_dep, lines[jjj]]
>    stel_wave =[stel_wave, ddd[jjj]]
>    ;stel_ra = [stel_ra,ratio2[jjj]]
>    endif
>
>  endfor
>
>  ;Finding the relative distance between the lines that left from the science spectrum:
>
>  ratio_stlines = fltarr(n_elements(stel_wave))
>  for k = 0L, n_elements(stel_wave) - 1 do begin
>    ratio_stlines[k] =  stel_wave[0] - stel_wave[k]
>  endfor
```

>
>
> And I will stop here because I have more problems below...
>
> Thank you very much in advance!
