Subject: Faster approach to calculating z-scores
Posted by JessW on Tue, 08 Feb 2011 20:41:49 GMT
View Forum Message <> Reply to Message

Hi all,

I would appreciate any ideas about how to efficiently calculate Z-scores for a (large) stack of images. I have a working program that seems unnecessarily clunky.

I have 3 files:

- Stack of NDVI images (ns = 5000, nl = 5000, nb = 20)
- A single-band classification map (7 classes) with the same dimensions as the NDVI stack
- A look-up table with the mean and standard deviation for each class for each time period (7classes x 20bands = 140 entries).

Right now my program runs as follows:

For each class in the classification map, I retrieve the indices of all pixels of that class. I use histograms and ENVI_GET_SLICE to extract the values of the relevant pixels in each line. I extract the class's mean and standard deviation data from the look-up table, rebin it so that it matches the dimensions of the extracted data, and perform the z-score calculation. I read the resulting data into a Z-score array.

It works, but I can't help but think there's a more efficient approach. Right now it takes 10 minutes to go through a 5,000,000 pixel subset. (My computer balks at processing the whole image--but that's another topic).

The relevant code is below, in case my explanation of the process was confusing. I appreciate any suggestions!

```
FOR i=0,nClasses-1 DO BEGIN

; Extract the class's mean/std data from the look-up table
        rowStart=(classes[i]-1)*nBands
        rowEnd=rowStart + nBands - 1
        tableSubset=table(*,rowStart:rowEnd)

; Get indices of all pixels of class [i] from the classification image
       classPix = where(classScene EQ classes[i])

; Transform the single indices to row/column format
        indexPix = Array_indices(classScene, classPix)
```

```
; Reform into cols/line arrays
      sampleCols=reform(indexPix[0,*])
      sampleRows=reform(indexPix[1,*])

; Calculate histogram of lines
      h=histogram(sampleLines, MIN=0, REVERSE_INDICES=ri)

; Loop through histogram bins
      FOR j=0, N_elements(h)-1 DO BEGIN
            IF ri[j+1] GT re[j] THEN BEGIN
               nElements = h[j]
               currLines = ri[ri[j]:ri[j+1]-1]
               sampleColsInLines = sampleCols[currLines]

; Create and populate array with pixel values extracted from the NDVI
stack
               pixVals = FLTARR(nElements, nBands]
               pixVals =
Get_Sample_Spectra(fid1,j,samplecolsInLine, allNDVIBands)

; Expand the look-up table data to the dimensions of the extracted
data array
               tableSubsetMean = rebin(tableSubset[0,*],
nElements, nBands)
               tableSubsetStd = rebin(tableSubset[1,*],
nElements,nBands)

; Calculate Z-score for the extracted pixels
              pixVals = (pixVals - tableSubsetMean)/
tableSubsetStd

; Write the Z-scores to the appropriate locations in the Zscore array
               Zscores[sampleColsInLine, j, *] = pixVals

         ENDIF
    ENDFOR
ENDFOR



FUNCTION get_sample_spectra, fid, line, sampleCols, bandsSelected
  Slice=ENVI_GET_SLICE(FID=fid, line=line,pos=bandsSelected)
  RETURN, Slice[sampleCols,*]
END
```