Subject: Variogram
Posted by toreto on Tue, 09 Dec 2003 08:52:48 GMT
View Forum Message <> Reply to Message

I'm working in satellite images and I'd like calculate the variogram,
Does Anybody know of variogram (or semivariogram) code based in IDL (-ENVI)?

Your assistance is much appreciated

Regards

Nacho

Subject: Re: Variogram
Posted by eoraptor on Wed, 19 May 2004 18:25:24 GMT
View Forum Message <> Reply to Message

Stephane,

Here's some code that I found in one of my IDL libraries from a long
time ago. I haven't used it in years, but the last time I did it
worked ok. You have to reshape your image into two arrays: one array
(vector) called DATA of length N which contains only the values for
each pixel, and the other array called LOCS which is of shape (d,N)
where d is the dimension of your space. For satelite images, you'll
construct LOCS(2,N) and populate it with the map coordinates of each
pixel. I'm sure there are better ways of computing the variogram
without all of the loops in the code below. It computes much faster
if you select a random sample (or select a specific subset of pixels)
of the imags and pass that along for variogram calculation rather than
the entire image.

If someone wants to re-work this procedure (remove some loops) to make
it more efficient, by all means go ahead.

```
;VGRAM2.PRO

;computes the empirical (isotropic, homogeneous) (semi)variogram from
a set of
;given data
;adapted from varigram.pro - this version is compatible with very
large data sets

;TMO  4-99

PRO vgram2, data, locs, bintype, binparms, vargram, vardist, count
```

```
;INPUT
;data    - values of data, array of length N
;locs    - locations data, array of length d by N where d = dimension
of space
;bintype - flag that specifies type of binning of values along
"distance" axis
;binparms - parameters needed to fully specify the binning for each
type
;types:  1. nbins = binparms(0), equal-sized with binsize =
binparms(1) beginning at
;          minbin = binparms(2) (maxdist = minbin +
(nbins+1)*binsize)
;       2. user specifies (in binparms) nbins+1 bin boundaries
ordered from small
;          to large
;       In either case, interpoint distances falling outside the bins
are ignored.

;OUTPUT
;vargram - values of the variogram
;vardist - distance for each value of variogram
;vargram(i) and dist(i) are the average over "point" values falling
into the ith bin.
;count - number of pairs falling into each bin, giving in idea of the
precision of the estimate.

sz = size(locs)
if sz(0) eq 1 then d = 1 else d = sz(2)
N = n_elements(data)

;construct bin boundaries
if bintype eq 1 then begin
  nbins = binparms(0) & binsize = binparms(1) & minbin = binparms(2)
  bounds = binsize*findgen(nbins+1) + minbin
endif else if bintype eq 2 then begin
  nbins = n_elements(binparms)-1 & bounds = binparms
endif

vargram = fltarr(nbins)
vardist = fltarr(nbins)
count = fltarr(nbins)
maxdist = 0.
for i=1,N-1 do begin
  for j=0,i-1 do begin
    if d eq 1 then pdist = abs(locs(i) - locs(j)) else pdist =
norm(locs(*,i) - locs(*,j))
    if pdist gt maxdist then maxdist = pdist
    subs = where(bounds le pdist, cnt)
```

```
    if cnt gt 0 and cnt le nbins then begin
      sub = max(subs)
      vardist(sub) = vardist(sub) + pdist
      vargram(sub) = vargram(sub) + .5*(data(i)-data(j))^2
      count(sub) = count(sub) + 1.
    endif
  endfor
endfor

for i=0,nbins-1 do begin
  if count(i) gt 0 then begin
    vardist(i) = vardist(i)/count(i)
    vargram(i) = vargram(i)/count(i)
  endif else vardist(i) = .5*(bounds(i) + bounds(i+1))
endfor

END
```