Subject: Re: speed-up computation of kernel-based "statistics"
Posted by ben.bighair on Thu, 19 Apr 2012 01:50:00 GMT
View Forum Message <> Reply to Message

On Wednesday, April 18, 2012 5:28:28 AM UTC-4, lbusett wrote:
> Hi all,
>
> I have two large (20000*20000) images. The first one is a
> classification, with discrete values from 1 to 10, while the second
> one contains values of a variable of interest. For each pixel, I have
> to compute the 5th and the 95th percentile of the values of the
> variable in a 801*801 window (centered on the selected pixel), for the
> pixels of the same class of the center pixel.
>

Hi,

It looks like you are doing a lot of ordering and sorting - IDL's histogram function can really help you here. Let's call the two 801x801 subsets "class" and "data".

```
; first compute the histogram of your class info.
; Make sure that you are using the data type of 'class' to define min and max.
h = histogram(class, min = 1, max = 10, reverse_indices = ri)

; now use David Fanning's REVERSE_INDICES to get the pixel locations for pixels
; that have the same class as your 'selected pixel'
selectedPixel = 3  ; suppose your center pixel is class=3

; the histogram bin values were specified as 1,2,3,4, ... 10 which means the
; selected pixel class value can be used as an index into the histogram
idx = reverse_indices(ri, selectedPixel - 1, count = n)

; I think n has to be at least one (for the selected pixel)

; now collect those pixels from your 'data' subset - they'll have the same pixel
; addresses
d = data[idx]
```

I don't have IDL in front of me just now so I could have botched part of it. From here your next step is to order d and find your quantiles. I'm a bit fuzzy on this part but maybe something like this...

```
ds = d[sort(d)]
ix = n*[0.05, 0.95]  ; n is from above, the count of elements in d
p = ds[ix]  ; so p should have the values for 5th and 95th quantiles.
```

Cheers,
Ben