Subject: Re: Principal Components Analysis
Posted by davidf on Wed, 11 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Denis (denis_nadeau@hotmail.com) writes:

> I am looking for a PCA program that can handle 1500 images (2048x1024
> pixels). Does someone
> have any idea where to find this.

Star Trek? :-)

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Principal Components Analysis
Posted by Peter Mason on Fri, 13 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

On Wed, 11 Feb 1998, Denis wrote:
> I am looking for a PCA program that can handle 1500 images (2048x1024
> pixels). Does someone
> have any idea where to find this.

My first reaction on seeing this post was sort of like David Fanning's. (And you're gonna need a whole *lot* of dilithium crystals.)

(BTW, I assume that you mean an image 2048 samples across, 1024 lines down, and 1500 bands deep.)

But then, it's actually not a totally unreasonable calculation for current computers, even a PC (but a PII with plenty of memory).
I've used IDL to find eigenvectors for a 600*600 covariance matrix, and it took a few minutes on a crusty old Pentium 120. I gather that the two routines commonly used for this (TRIRED and TRIQL) are order N**3, so you'd be looking at roughly 20-30 times this wait (on the same machine). On a decent machine, I'd expect it to take around half an hour to an hour - bad but not impossible.

That said, here's a loose decription of how to go about calculating a PC transform in IDL. (I don't have any actual code handy. Besides, I'd expect that the bulk of any PCA code would be spent on logistics - getting data to and from the image and so on.)
1 Calculate the covariance matrix of the image (and keep the band beans handy)

2 Run the covar. mat. through TRIRED (trired,cvmat,d,e,/DOUBLE)
3 Run TRIRED's output through TRIQL (triql,d,e,cvmat,/DOUBLE)
  Cvmat then contains the transformation eigenvectors.
  (BTW, the J'th eigenvector is cvmat(*,J), I think.)

And to effect a transform on an image:
4 Decide on how many PC bands you want (typically by looking at the eigenvalues
  returned in D in step 3, and picking the point where they drop off to
  "insignificance"). (N)
5 Subtract the band means (from step 1) from each pixel in the image
6 Matrix multiply each mean-corrected image pixel with the first N eigenvectors
  in cvmat to get an N-band output pixel (usually done for a whole bunch of
  pixels at once, in a single matrix multiply).

So it's generally a simple operation. However, for a 2048*1024*1500 image
steps 1, 4 and 5 are going to be harder than usual - obviously, you can't fit
your entire image in memory. You'd probably want the image in "band
interleaved by line" or "band interleaved by pixel" format so that you can
easily read and work with a full line (all bands) at a time. And you'd
want all calculations done in double precision.

Covariance matrices:
If you had an image IMG(ns,nl,nb) (ns samples by nl lines by nb bands), and it
fitted comfortably in memory, you could get its covariance matrix as follows:
. img=reform(img,ns*nl,nb,/overw) ;reorganise
. band_means=total(img,1,/double)/(ns*nl)
. for i=0L,nb-1L do img(0,i)=img(*,i)-band_means(i)
. cvmat=img##transpose(img)/(ns*nl-1)
For *your* images you'd have to do this in stages, working on, say, one full
line at a time. Something like this:
. band_means=dblarr(nb)
. Loop thru the image, reading ns pixels at a time
   [acquire slice(ns,nb) and make it DOUBLE]
   band_means=temporary(band_means)+total(slice,1,/double)
. band_means=temporary(band_means)/(ns*nl)
. cvmat=dblarr(nb,nb)
. Loop thru the image, reading ns pixels at a time
   [acquire slice(ns,nb) and make it DOUBLE]
   for i=0L,nb-1L do slice(0,i)=slice(*,i)-band_means(i)
   cvmat=temporary(cvmat) + slice##transpose(slice)
. cvmat=temporary(cvmat)/(ns*nl-1)
For an image of the size you mentioned (2048*1024), you might still get
significant roundoff error. There may be a better method for calculating
the covariance matrix.

If you weren't going to do this PCA too often, I'd suggest giving the usual
eigenvector method (TRIRED followed by TRIQL) a try - it's ready to go in

IDL. I don't know how accurate it'll be for a 1500*1500 covariance matrix, but it's worth a try.
Otherwise, it might be worthwhile to pursue a routine that finds just N eigenvectors (corresponding to the N biggest eigenvalues). With a 1500-band image, and assuming that the bands are fairly well correlated, I'd expect that you'd end up using a fraction (less than 100?) of the PC bands.

Peter Mason

Subject: Re: Principal Components Analysis
Posted by Erard on Mon, 16 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <34E22A9A.58437FE5@hotmail.com>, Denis <denis_nadeau@hotmail.com> wrote:

> I am looking for a PCA program that can handle 1500 images (2048x1024
> pixels). Does someone
> have any idea where to find this.
>

There is a PCA routine in Astron, the Nasa IDL library. I'm not sure it can handle such large images, though.

--
Stï¿½phane Erard
Institut d'Astrophysique Spatiale
Orsay, France
