Subject: Re: Principal Componets Analysis
Posted by David Fanning on Tue, 04 Sep 2007 01:06:00 GMT
View Forum Message <> Reply to Message

David Fanning writes:

Principal "Componets" Analysis!? It's probably just as well
that no one is going to find that tutorial. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Principal Componets Analysis
Posted by Jeff N. on Tue, 04 Sep 2007 16:02:46 GMT
View Forum Message <> Reply to Message

David,

I'm reading through the tutorial, and spotted something you might want to fix in your discussion of CORRELATE. It's not true that negative values returned from that function are uncorrelated. You actually describe the reason why this is true in your tutorial: "a change in one vector will predict an opposite change in the other." if one vector predicts a change in the other, it's correlated. The negative sign just means that the change is an opposite change, like you mentioned. How well two variables are correlated depends on the magnitude of the number: 0 is not correlated, 1 is perfectly correlated (so -1 is perfectly negatively correlated).

Jeff

Subject: Re: Principal Componets Analysis
Posted by David Fanning on Tue, 04 Sep 2007 16:23:50 GMT
View Forum Message <> Reply to Message

Jeff N. writes:

> I'm reading through the tutorial, and spotted something you might want

> to fix in your discussion of CORRELATE. It's not true that negative
> values returned from that function are uncorrelated. You actually
> describe the reason why this is true in your tutorial: "a change in
> one vector will predict an opposite change in the other." if one
> vector predicts a change in the other, it's correlated. The negative
> sign just means that the change is an opposite change, like you
> mentioned. How well two variables are correlated depends on the
> magnitude of the number: 0 is not correlated, 1 is perfectly
> correlated (so -1 is perfectly negatively correlated).

Oh, my goodness. I must have been asleep when I wrote that.
One of the problems with working at 3AM. :-(

Fixed now. Thanks.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Principal Componets Analysis
Posted by wlandsman on Wed, 05 Sep 2007 14:45:40 GMT

View Forum Message <> Reply to Message

On Sep 3, 8:33 pm, David Fanning <n...@dfanning.com> wrote:
>
> You can find the tutorial here:
>
>   http://www.dfanning.com/code_tips/pca.html
>
> Any and all comments welcome.

Well, a minor historical comment about the different PCA
conventions.   The pcomp.pro procedure was introduced into IDL in
1996, but prior to that Immanuel Freedman had written a procedure
pca.pro (http://idlastro.gsfc.nasa.gov/ftp/pro/math/pca.pro) based on
a FORTRAN program by Fionn Murtagh.

When pcomp.pro was introduced, it took me a long time to prove that
pca.pro and pcomp.pro gave the same results   Below are the notes I
wrote at the time:

****

The intrinsic IDL function PCOMP duplicates most most of the functionality of PCA, but uses different conventions and normalizations. Note the following:

(1) PCOMP requires a N_ATTRIB x N_OBJ input array; this is the transpose of what PCA expects
(2) PCA uses standardized variables for the correlation matrix: the input vectors are set to a mean of zero and variance of one and divided by sqrt(n); use the /STANDARIZE keyword to PCOMP for a direct comparison.
(3) PCA (unlike PCOMP) normalizes the eigenvectors by the square root of the eigenvalues.
(4) PCA returns cumulative percentages; the VARIANCES keyword of PCOMP returns the variance in each variable
(5) PCOMP divides the eigenvalues by (1/N_OBJ-1) when the covariance matrix is used.

****

And, yes, I verified that pca.pro also reproduces the results in your tutorial, but it requires even more adjustment than does pcomp.pro !