Subject: Re: Principle Componets Analysis
Posted by Jeff N. on Thu, 23 Aug 2007 19:32:24 GMT
View Forum Message <> Reply to Message

On Aug 23, 2:27 pm, David Fanning <da...@dfanning.com> wrote:
> Folks,
>
> Is it just me, or is this turning into an ENVI forum...
>
> I have three bands of a Landsat image. They come in
> three separate files (GEOTIFF). I have opened all three
> in ENVI. I wish to perform a principle components analysis
> of these three images, and capture the first principle
> component for further analysis.
>
> ENVI seems to want to have these three images in the
> same file or something. In any case, I can't work out
> how to get these three images into the PC apparatus.
> Perhaps if I could make a new ENVI image of these three
> bands? What if I wanted to do the analysis with four image
> bands?
>
> Any help appreciated.
>
> Thanks,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/

David,

You can easily combine different images into a single cube in ENVI.
Just go to File -> Save File As -> Envi Standard. Note that this
"File" menu is in the main ENVI menu, not the file menu for a display
window. You can do it for any number of bands, even different
combinations of bands in different images, and once you combine them
the PCA should work just fine.

Jeff

Subject: Re: Principle Componets Analysis
Posted by David Fanning on Thu, 23 Aug 2007 19:49:04 GMT
View Forum Message <> Reply to Message

Jeff N. writes:

> You can easily combine different images into a single cube in ENVI.
> Just go to File -> Save File As -> Envi Standard. Note that this
> "File" menu is in the main ENVI menu, not the file menu for a display
> window. You can do it for any number of bands, even different
> combinations of bands in different images, and once you combine them
> the PCA should work just fine.

Ah, fabulous. Piece of cake!

In case anyone is interested, here is a reference to a
FABULOUS explanation of priciple components analysis.
After reading it, I was convinced I could do PCA in IDL
if I couldn't get the darn ENVI stuff to work. But, in
the end, it was easier to let ENVI do the heavy lifting.

http://tinyurl.com/3aaeb6

Cheers,

David

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

Subject: Re: Principle Componets Analysis
Posted by Jeff N. on Thu, 23 Aug 2007 20:04:33 GMT
View Forum Message <> Reply to Message

On Aug 23, 3:49 pm, David Fanning <da...@dfanning.com> wrote:
> Jeff N. writes:
>> You can easily combine different images into a single cube in ENVI.
>> Just go to File -> Save File As -> Envi Standard. Note that this
>> "File" menu is in the main ENVI menu, not the file menu for a display
>> window. You can do it for any number of bands, even different
>> combinations of bands in different images, and once you combine them
>> the PCA should work just fine.
>
> Ah, fabulous. Piece of cake!
>
> In case anyone is interested, here is a reference to a
> FABULOUS explanation of priciple components analysis.
> After reading it, I was convinced I could do PCA in IDL
> if I couldn't get the darn ENVI stuff to work. But, in

> the end, it was easier to let ENVI do the heavy lifting.
>
> http://tinyurl.com/3aaeb6
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/

I've gone through that tutorial many times myself, almost as much as
the histogram tutorial! :)

Jeff

Subject: Re: Principle Componets Analysis
Posted by Jeff N. on Thu, 23 Aug 2007 21:32:10 GMT
View Forum Message <> Reply to Message

On Aug 23, 3:49 pm, David Fanning <da...@dfanning.com> wrote:
> Jeff N. writes:
>> You can easily combine different images into a single cube in ENVI.
>> Just go to File -> Save File As -> Envi Standard. Note that this
>> "File" menu is in the main ENVI menu, not the file menu for a display
>> window. You can do it for any number of bands, even different
>> combinations of bands in different images, and once you combine them
>> the PCA should work just fine.
>
> Ah, fabulous. Piece of cake!
>
> In case anyone is interested, here is a reference to a
> FABULOUS explanation of priciple components analysis.
> After reading it, I was convinced I could do PCA in IDL
> if I couldn't get the darn ENVI stuff to work. But, in
> the end, it was easier to let ENVI do the heavy lifting.
>
> http://tinyurl.com/3aaeb6
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/

Now that I think about it, I remember trying a while back going through that tutorial and trying to get the same answer using PCOMP....but never could. Has anyone else tried this? Did you get good agreement?

Jeff

Subject: Re: Principle Componets Analysis
Posted by KRDean on Fri, 24 Aug 2007 02:42:32 GMT
View Forum Message <> Reply to Message

I find the book Image Analysis, Classification and Change Detection in Remote Sensing, with Algorothms for ENIV/IDL by Morton Canty handy. He provides ENVI/IDL code to do the PCAs.

Kelly Dean
Fort Collins, Colorado

On Aug 23, 12:27 pm, David Fanning <da...@dfanning.com> wrote:
> Folks,
>
> Is it just me, or is this turning into an ENVI forum...
>
> I have three bands of a Landsat image. They come in
> three separate files (GEOTIFF). I have opened all three
> in ENVI. I wish to perform a principle components analysis
> of these three images, and capture the first principle
> component for further analysis.
>
> ENVI seems to want to have these three images in the
> same file or something. In any case, I can't work out
> how to get these three images into the PC apparatus.
> Perhaps if I could make a new ENVI image of these three
> bands? What if I wanted to do the analysis with four image
> bands?
>
> Any help appreciated.
>
> Thanks,
>
> David
> --

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

Subject: Re: Principle Componets Analysis
Posted by David Fanning on Fri, 24 Aug 2007 02:59:51 GMT
View Forum Message <> Reply to Message

kBob writes:

> I find the book Image Analysis, Classification and Change Detection
> in Remote Sensing, with Algorothms for ENIV/IDL by Morton Canty handy.
> He provides ENVI/IDL code to do the PCAs.

Well, I'm ashamed to say, I had read part's of Mort's book
earlier in the week and found I needed, well, more remedial
help. Quite frankly, I didn't understand a word of it. :-(

The Lindsay Smith tutorial, on the other hand, was crystal
clear. So much so that I came back to my office and wrote up
the example in IDL, just to see if I could follow it.

It turns out, that the PCOMP function in IDL gives essentially
the same answer as the tutorial (this for Jeff's benefit), but
the values are scaled slightly differently. However they
plot on exactly the same line in the end. Here is the code
I used.

```
; Method according to the Lindsay Smith tutorial:
; http://tinyurl.com/3aaeb

x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]

xmean = x - Mean(x)
ymean = y - Mean(y)
Window, XSIZE=600, YSIZE=800
!P.MULTI=[0,1,2]
Plot, xmean, ymean, PSYM=7

dataAdjust = Transpose([ [xmean], [ymean] ])
covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)

Print, 'EIGENVALUES: ', eigenvalues
Print, 'EIGENVECTORS: '
Print, eigenvectors
```

```
rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
;rowFeatureVector = eigenvectors
finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
!P.MULTI=0



; Method using PCOMP in IDL library.
data = Transpose([[x],[y]])
r = PCOMP(data, /COVARIANCE, NVARIABLES=1, EIGENVALUES=ev, /STANDARDIZE)
Print, 'IDL EIGENVALUES: ', ev


; Compare methods.
Window, 1
PLOT, r
OPLOT, finalData, LINESTYLE=2

Window, 2
PLOT, r + Mean(x), r + Mean(y), PSYM=2
OPLOT, finalData + Mean(x), finalData + Mean(y), PSYM=7
END
```

This is really nice stuff and has me EXTREMELY jazzed about
the potential of it. :-)

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: Principle Componets Analysis
Posted by Ben Panter on Fri, 24 Aug 2007 10:06:30 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

> ; Method according to the Lindsay Smith tutorial:
> ; http://tinyurl.com/3aaeb

US Senate Roll Call Votes 107th Congress?

I think you're missing the 6:

http://tinyurl.com/3aaeb6

cheers,

Ben

--
Ben Panter, Edinburgh, UK.
Email false, http://www.benpanter.co.uk
or you could try ben at ^^^^^^^^^^^^^^^

Subject: Re: Principle Componets Analysis
Posted by yp on Fri, 24 Aug 2007 10:21:46 GMT
View Forum Message <> Reply to Message

On Aug 24, 3:59 am, David Fanning <n...@dfanning.com> wrote:
> kBob writes:
>> I find the book Image Analysis, Classification and Change Detection
>> in Remote Sensing, with Algorothms for ENIV/IDL by Morton Canty handy.
>> He provides ENVI/IDL code to do the PCAs.
>
> Well, I'm ashamed to say, I had read part's of Mort's book
> earlier in the week and found I needed, well, more remedial
> help. Quite frankly, I didn't understand a word of it. :-(
>
> The Lindsay Smith tutorial, on the other hand, was crystal
> clear. So much so that I came back to my office and wrote up
> the example in IDL, just to see if I could follow it.
>
> It turns out, that the PCOMP function in IDL gives essentially
> the same answer as the tutorial (this for Jeff's benefit), but
> the values are scaled slightly differently. However they
> plot on exactly the same line in the end. Here is the code
> I used.
>
> ; Method according to the Lindsay Smith tutorial:
> ;http://tinyurl.com/3aaeb
>
> x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
> y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]
>
> xmean = x - Mean(x)
> ymean = y - Mean(y)

> Window, XSIZE=600, YSIZE=800
> !P.MULTI=[0,1,2]
> Plot, xmean, ymean, PSYM=7
>
> dataAdjust = Transpose([[xmean], [ymean]])
> covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
> eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)
>
> Print, 'EIGENVALUES: ', eigenvalues
> Print, 'EIGENVECTORS: '
> Print, eigenvectors
>
> rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
> ;rowFeatureVector = eigenvectors
> finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
> Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
> !P.MULTI=0
>
> ; Method using PCOMP in IDL library.
> data = Transpose([[x],[y]])
> r = PCOMP(data, /COVARIANCE, NVARIABLES=1, EIGENVALUES=ev, /STANDARDIZE)
> Print, 'IDL EIGENVALUES: ', ev
>
> ; Compare methods.
> Window, 1
> PLOT, r
> OPLOT, finalData, LINESTYLE=2
>
> Window, 2
> PLOT, r + Mean(x), r + Mean(y), PSYM=2
> OPLOT, finalData + Mean(x), finalData + Mean(y), PSYM=7
> END
>
> This is really nice stuff and has me EXTREMELY jazzed about
> the potential of it. :-)
>
> 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.")

Hi David,
Yes, both methods are essentially same except that the data in
Method#1 are NOT standardised. You will get exactly same result if you

do
xmean = (x - Mean(x) / Stddev(x)
ymean = (y - Mean(y) / STddev(y)

--yas

Subject: Re: Principle Componets Analysis
Posted by yp on Fri, 24 Aug 2007 10:24:47 GMT
View Forum Message <> Reply to Message

On Aug 24, 11:21 am, Yaswant Pradhan <Yaswant.Prad...@gmail.com>
wrote:
> On Aug 24, 3:59 am, David Fanning <n...@dfanning.com> wrote:
>
>
>
>> kBob writes:
>>> I find the book Image Analysis, Classification and Change Detection
>>> in Remote Sensing, with Algorothms for ENIV/IDL by Morton Canty handy.
>>> He provides ENVI/IDL code to do the PCAs.
>
>> Well, I'm ashamed to say, I had read part's of Mort's book
>> earlier in the week and found I needed, well, more remedial
>> help. Quite frankly, I didn't understand a word of it. :-(
>
>> The Lindsay Smith tutorial, on the other hand, was crystal
>> clear. So much so that I came back to my office and wrote up
>> the example in IDL, just to see if I could follow it.
>
>> It turns out, that the PCOMP function in IDL gives essentially
>> the same answer as the tutorial (this for Jeff's benefit), but
>> the values are scaled slightly differently. However they
>> plot on exactly the same line in the end. Here is the code
>> I used.
>
>> ; Method according to the Lindsay Smith tutorial:
>> ;http://tinyurl.com/3aaeb
>
>> x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
>> y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]
>
>> xmean = x - Mean(x)
>> ymean = y - Mean(y)
>> Window, XSIZE=600, YSIZE=800
>> !P.MULTI=[0,1,2]
>> Plot, xmean, ymean, PSYM=7
>

```
>>  dataAdjust = Transpose([ [xmean], [ymean] ])
>>  covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
>>  eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)
>
>>  Print, 'EIGENVALUES: ', eigenvalues
>>  Print, 'EIGENVECTORS: '
>>  Print, eigenvectors
>
>>  rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
>>  ;rowFeatureVector = eigenvectors
>>  finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
>>  Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
>>  !P.MULTI=0
>
>>  ; Method using PCOMP in IDL library.
>>  data = Transpose([[x],[y]])
>>  r = PCOMP(data, /COVARIANCE, NVARIABLES=1, EIGENVALUES=ev, /STANDARDIZE)
>>  Print, 'IDL EIGENVALUES: ', ev
>
>>  ; Compare methods.
>>  Window, 1
>>  PLOT, r
>>  OPLOT, finalData, LINESTYLE=2
>
>>  Window, 2
>>  PLOT, r + Mean(x), r + Mean(y), PSYM=2
>>  OPLOT, finalData + Mean(x), finalData + Mean(y), PSYM=7
>>  END
>
>>  This is really nice stuff and has me EXTREMELY jazzed about
>>  the potential of it. :-)
>
>>  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.")
>
>  Hi David,
>  Yes, both methods are essentially same except that the data in
>  Method#1 are NOT standardised. You will get exactly same result if you
>  do
>  xmean = (x - Mean(x) / Stddev(x)
>  ymean = (y - Mean(y) / STddev(y)
>
```

> --yas

whoops... missed a parenthesis, should read xmean = (x - Mean(x)) /
Stddev(x) and likewise.

Subject: Re: Principle Componets Analysis
Posted by Mort Canty on Fri, 24 Aug 2007 11:22:38 GMT
View Forum Message <> Reply to Message

David Fanning schrieb:
> kBob writes:
>
>> I find the book Image Analysis, Classification and Change Detection
>> in Remote Sensing, with Algorothms for ENIV/IDL by Morton Canty handy.
>> He provides ENVI/IDL code to do the PCAs.
>
> Well, I'm ashamed to say, I had read part's of Mort's book
> earlier in the week and found I needed, well, more remedial
> help. Quite frankly, I didn't understand a word of it. :-(
>

Aw jeez, David, I understood *your* book :-)

Mort

Subject: Re: Principle Componets Analysis
Posted by David Fanning on Fri, 24 Aug 2007 12:24:23 GMT
View Forum Message <> Reply to Message

mort canty writes:

> Aw jeez, David, I understood *your* book :-)

Well, you know, once the light goes on, a LOT of
things start to make sense. I'm going to re-visit it. :-)

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: Principle Componets Analysis
Posted by David Fanning on Fri, 24 Aug 2007 15:46:49 GMT
View Forum Message <> Reply to Message

Yaswant Pradhan writes:

> Yes, both methods are essentially same except that the data in
> Method#1 are NOT standardised. You will get exactly same result if you
> do
> xmean = (x - Mean(x) / Stddev(x)
> ymean = (y - Mean(y) / STddev(y)

Well, not exactly. Did you run the example with this change?
I get something quite a bit different, although still "correct"
I think.

```
;********************************************************** ***
; Method according to the Lindsay Smith tutorial:
; http://tinyurl.com/3aaeb6

x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]

xmean = (x - Mean(x)) / STDDEV(x, /DOUBLE)
ymean = (y - Mean(y)) / STDDEV(y, /DOUBLE)
Window, XSIZE=600, YSIZE=800
!P.MULTI=[0,1,2]
Plot, xmean, ymean, PSYM=7

dataAdjust = Transpose([ [xmean], [ymean] ])
covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)

Print, 'EIGENVALUES: ', eigenvalues
Print, 'EIGENVECTORS: '
Print, eigenvectors

rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
;rowFeatureVector = eigenvectors
finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
!P.MULTI=0



; Method using PCOMP in IDL library.
data = Transpose([[x],[y]])
r = PCOMP(data, /COVARIANCE, NVARIABLES=1, $
   EIGENVALUES=ev, /STANDARDIZE)
```

```
Print, 'IDL EIGENVALUES: ', ev


; Compare methods.
Window, 1
PLOT, r
OPLOT, finalData, LINESTYLE=2;, COLOR=FSC_Color('yellow')

Window, 2
PLOT, r + Mean(x), r + Mean(y), PSYM=2
OPLOT, finalData + Mean(x), finalData + Mean(y), $
  PSYM=7;, COLOR=FSC_Color('yellow')
END
 ;************************************************************ ***
```

The curves in Window 1 are worse than they were without
making the change you suggest.

Cheers,

David

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

Subject: Re: Principle Componets Analysis
Posted by David Fanning on Fri, 24 Aug 2007 15:53:35 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> mort canty writes:
>
>> Aw jeez, David, I understood *your* book :-)
>
> Well, you know, once the light goes on, a LOT of
> things start to make sense. I'm going to re-visit it. :-)

Well, I had another look at that PCA section this morning.
I'm pretty sure I must have gotten hold of the German version
of the book. :-(

Cheers,

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

Subject: Re: Principle Componets Analysis
Posted by David Streutker on Fri, 24 Aug 2007 16:44:40 GMT
View Forum Message <> Reply to Message

For a reason I haven't quite figured out, the results of PCOMP and the
ENVI results differ by a factor of the square root of the eigenvalue
for the corresponding band.

This works for me:

```
x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]

;xmean = x - Mean(x)
;ymean = y - Mean(y)
Window, XSIZE=600, YSIZE=800
!P.MULTI=[0,1,2]
Plot, xmean, ymean, PSYM=7

;dataAdjust = Transpose([ [xmean], [ymean] ])
dataAdjust = Transpose([ [x], [y] ])
covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)

Print, 'EIGENVALUES: ', eigenvalues
Print, 'EIGENVECTORS: '
Print, eigenvectors

rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
;rowFeatureVector = eigenvectors
finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
!P.MULTI=0

; Method using PCOMP in IDL library.
data = Transpose([[x],[y]])
;r = PCOMP(data, /COVARIANCE, NVARIABLES=1, EIGENVALUES=ev, /
STANDARDIZE)
r = PCOMP(data, /COVARIANCE, EIGENVALUES=ev)
Print, 'IDL EIGENVALUES: ', ev
```

```
; Compare methods.
Window, 1
;PLOT, r
PLOT, r[0,*] / sqrt(ev[0])
OPLOT, finalData, LINESTYLE=2

Window, 2
PLOT, r[0,*] / sqrt(ev[0]) + Mean(x), r[0,*] / sqrt(ev[0]) + Mean(y),
PSYM=2
;PLOT, r + Mean(x), r + Mean(y), PSYM=2
OPLOT, finalData + Mean(x), finalData + Mean(y), PSYM=7
END
```

Subject: Re: Principle Componets Analysis
Posted by David Streutker on Fri, 24 Aug 2007 16:50:47 GMT
View Forum Message <> Reply to Message

On Aug 24, 10:44 am, David Streutker <dstreut...@gmail.com> wrote:

```
>  For a reason I haven't quite figured out, the results of PCOMP and the
>  ENVI results differ by a factor of the square root of the eigenvalue
>  for the corresponding band.
>
>  This works for me:
>
>  x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
>  y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]
>
>  ;xmean = x - Mean(x)
>  ;ymean = y - Mean(y)
>  Window, XSIZE=600, YSIZE=800
>  !P.MULTI=[0,1,2]
>  Plot, xmean, ymean, PSYM=7
>
>  ;dataAdjust = Transpose([ [xmean], [ymean] ])
>  dataAdjust = Transpose([ [x], [y] ])
>  covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
>  eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)
>
>  Print, 'EIGENVALUES: ', eigenvalues
>  Print, 'EIGENVECTORS: '
>  Print, eigenvectors
>
>  rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
>  ;rowFeatureVector = eigenvectors
>  finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
>  Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
>  !P.MULTI=0
```

>
> ; Method using PCOMP in IDL library.
> data = Transpose([[x],[y]])
> ;r = PCOMP(data, /COVARIANCE, NVARIABLES=1, EIGENVALUES=ev, /
> STANDARDIZE)
> r = PCOMP(data, /COVARIANCE, EIGENVALUES=ev)
> Print, 'IDL EIGENVALUES: ', ev
>
> ; Compare methods.
> Window, 1
> ;PLOT, r
> PLOT, r[0,*] / sqrt(ev[0])
> OPLOT, finalData, LINESTYLE=2
>
> Window, 2
> PLOT, r[0,*] / sqrt(ev[0]) + Mean(x), r[0,*] / sqrt(ev[0]) + Mean(y),
> PSYM=2
> ;PLOT, r + Mean(x), r + Mean(y), PSYM=2
> OPLOT, finalData + Mean(x), finalData + Mean(y), PSYM=7
> END

And by "ENVI results", I mean of course the the IDL programmatic (non-PCOMP) method, which are equivalent. (Sorry, I've got ENVI on the brain this morning.)

Subject: Re: Principle Componets Analysis
Posted by yp on Fri, 24 Aug 2007 17:30:48 GMT
View Forum Message <> Reply to Message

On Aug 24, 4:46 pm, David Fanning <da...@dfanning.com> wrote:
> Yaswant Pradhan writes:
>> Yes, both methods are essentially same except that the data in
>> Method#1 are NOT standardised. You will get exactly same result if you
>> do
>> xmean = (x - Mean(x) / Stddev(x)
>> ymean = (y - Mean(y) / STddev(y)
>
> Well, not exactly. Did you run the example with this change?
> I get something quite a bit different, although still "correct"
> I think.
>
> ;** ***
> ; Method according to the Lindsay Smith tutorial:
> ;http://tinyurl.com/3aaeb6
>
> x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
> y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]

```
>
>  xmean = (x - Mean(x)) / STDDEV(x, /DOUBLE)
>  ymean = (y - Mean(y)) / STDDEV(y, /DOUBLE)
>  Window, XSIZE=600, YSIZE=800
>  !P.MULTI=[0,1,2]
>  Plot, xmean, ymean, PSYM=7
>
>  dataAdjust = Transpose([ [xmean], [ymean] ])
>  covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
>  eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)
>
>  Print, 'EIGENVALUES: ', eigenvalues
>  Print, 'EIGENVECTORS: '
>  Print, eigenvectors
>
>  rowFeatureVector = eigenvectors[0,*] ; Take first principle component.
>  ;rowFeatureVector = eigenvectors
>  finalData = Transpose(rowFeatureVector) ## Transpose(dataAdjust)
>  Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
>  !P.MULTI=0
>
>  ; Method using PCOMP in IDL library.
>  data = Transpose([[x],[y]])
>  r = PCOMP(data, /COVARIANCE, NVARIABLES=1, $
>     EIGENVALUES=ev, /STANDARDIZE)
>  Print, 'IDL EIGENVALUES: ', ev
>
>  ; Compare methods.
>  Window, 1
>  PLOT, r
>  OPLOT, finalData, LINESTYLE=2;, COLOR=FSC_Color('yellow')
>
>  Window, 2
>  PLOT, r + Mean(x), r + Mean(y), PSYM=2
>  OPLOT, finalData + Mean(x), finalData + Mean(y), $
>     PSYM=7;, COLOR=FSC_Color('yellow')
>  END
>   ;********************************************************** ***
>
>  The curves in Window 1 are worse than they were without
>  making the change you suggest.
>
>  Cheers,
>
>  David
>
>  --
>  David Fanning, Ph.D.
```

> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/

Not exactly, what I was looking at in your code is whether you get the same eigenvalues/vecors or not. I've changed the code slightly for dimension compatibility. What I would look at to compare both methods is - (i) eigenvalues (maginitude), (ii) sign (direction) of the components.

```
x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1]
y = [2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]


xmean = (x - Mean(x)) / STDDEV(x, /DOUBLE)
ymean = (y - Mean(y)) / STDDEV(y, /DOUBLE)
;Window, XSIZE=600, YSIZE=800
;!P.MULTI=[0,1,2]
;Plot, xmean, ymean, PSYM=7


dataAdjust = Transpose([ [xmean], [ymean] ])
;dataAdjust = ([ [xmean], [ymean] ])
covArray = Correlate(dataAdjust, /COVARIANCE, /DOUBLE)
eigenvalues = EIGENQL(covArray, EIGENVECTORS=eigenvectors, /DOUBLE)


Print, 'EIGENVALUES: ', eigenvalues
Print, 'EIGENVECTORS: '
Print, eigenvectors


;rowFeatureVector = eigenvectors[0,*] ; Take first principle
component.
rowFeatureVector = transpose(eigenvectors)
finalData = (dataAdjust) ## (rowFeatureVector)
;Plot, finaldata+Mean(x), finaldata+mean(y), PSYM=7
;!P.MULTI=0


; Method using PCOMP in IDL library.
data = Transpose([[x],[y]])
r = PCOMP(data, /COVARIANCE, EIGENVALUES=ev, /STANDARDIZE, /DOUBLE);,
NVARIABLES=1)
Print, 'IDL EIGENVALUES: ', ev


; Compare methods.
Window, 1, title='1st component'
```

```
PLOT, r[0,*]
OPLOT, finalData[0,*], LINESTYLE=2;, COLOR=FSC_Color('yellow')
OPLOT, [0,10],[0,0]


Window, 2, title='2nd component'
PLOT, finalData[1,*], LINESTYLE=2;, COLOR=FSC_Color('yellow')
OPLOT, r[1,*]
OPLOT, [0,10],[0,0]
```

Subject: Re: Principle Componets Analysis
Posted by Mort Canty on Sat, 25 Aug 2007 10:14:10 GMT
View Forum Message <> Reply to Message

David Fanning schrieb:
> David Fanning writes:
>
>> mort canty writes:
>>
>>> Aw jeez, David, I understood *your* book :-)
>> Well, you know, once the light goes on, a LOT of
>> things start to make sense. I'm going to re-visit it. :-)
>
> Well, I had another look at that PCA section this morning.
> I'm pretty sure I must have gotten hold of the German version
> of the book. :-(
>
> Cheers,
>
> David

Ah well. But do remember, it's a textbook. You won't follow the notation in Chapter 3 without slogging through Chapters 1 and 2 first. Anyway, please don't review it for Amazon :-)

Mort

Subject: Principal Componets Analysis
Posted by David Fanning on Tue, 04 Sep 2007 00:33:37 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Well, I'm ashamed to say, I had read part's of Mort's book
> earlier in the week and found I needed, well, more remedial

> help. Quite frankly, I didn't understand a word of it. :-(
>
> The Lindsay Smith tutorial, on the other hand, was crystal
> clear (http://tinyurl.com/3aaeb6). So much so that I came
> back to my office and wrote up the example in IDL, just
> to see if I could follow it.

Well, I'm back, this time with a spelling of "principal"
that might even make this discussion of Principal Components
Analysis (PCA) available to people who can spell correctly
on Google searches. (And I have bookmarked the difference
between "principle" as a fundamental truth or law, and
"principal" as the first in rank, in my dictionary. How
you get to be my age without knowing this is beyond me.)

I've been pretty much obsessed with PCA for the past couple
of weeks. For one thing, I felt badly about telling Mort
I couldn't understand a word of his excellent book. (It
does grow on you when you finally find yourself up to
speed again on some of the mathematical notations.)

Anyhow, I've wanted to understand this. Probably for the
same reason I've been studying Spanish so diligently
lately: it just seems like it might come in handy some
day. Does anyone know if they use PCA in Costa Rica?

PCA *is* fairly straightforward. At least after you
understand it, it is. Getting there is something else.
I suspect there are more of you out there like me who
would appreciate a--shall we say--less mathematical
approach to the subject.

I found MUCH became clear after reading the above
mentioned Lindsay Smith tutorial:

http://tinyurl.com/3aaeb6

But there were still a few unresolved problems for me.
One of these was why there are two ways to do PCA in
IDL, and why you don't get the same answer when you use
them. I discovered, eventually, that you DO get the same
answer, but this took me a whole lot longer to figure
out than it probably should have.

I also wanted to understand the use of PCA for images,
so I looked into that a little bit, too.

All this to say that I have written what I am calling

a PCA Tutorial, although that is probably a lofty title
for a piece of writing that is more like the blind leading
the blind. :-)

I would appreciate feedback on this from those of you
who know a lot more about it than I do. I tried to let
the Smith tutorial do the heavy theoretical lifting.
What I wanted to know was how to do this in IDL. So that
is the focus here. (I did find what I think is an error
in the Smith tutorial, for what it is worth.)

You can find the tutorial here:

  http://www.dfanning.com/code_tips/pca.html

Any and all comments welcome.

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 rjp23 on Tue, 20 Jun 2017 11:39:19 GMT
View Forum Message <> Reply to Message

This is a bit of an extreme bump but I've just stumbled upon this and need to do something similar.

The disclaimer here (http://www.idlcoyote.com/code_tips/pca.html) bothers me a bit: [Editor Note: After writing an article on EOF Analysis, which is identical to Principal Component Analysis, I have a feeling this section of the article might be slightly misleading. Let's just say, I wouldn't do it this way if I were doing the analysis again now.]

I don't know if David's still around but could someone elaborate a bit more on that? What's the issue with the article that'd be done differently?

Thanks

On Tuesday, September 4, 2007 at 1:33:37 AM UTC+1, David Fanning wrote:
> David Fanning writes:
>

>> Well, I'm ashamed to say, I had read part's of Mort's book
>> earlier in the week and found I needed, well, more remedial
>> help. Quite frankly, I didn't understand a word of it. :-(
>>
>> The Lindsay Smith tutorial, on the other hand, was crystal
>> clear (http://tinyurl.com/3aaeb6). So much so that I came
>> back to my office and wrote up the example in IDL, just
>> to see if I could follow it.
>
> Well, I'm back, this time with a spelling of "principal"
> that might even make this discussion of Principal Components
> Analysis (PCA) available to people who can spell correctly
> on Google searches. (And I have bookmarked the difference
> between "principle" as a fundamental truth or law, and
> "principal" as the first in rank, in my dictionary. How
> you get to be my age without knowing this is beyond me.)
>
> I've been pretty much obsessed with PCA for the past couple
> of weeks. For one thing, I felt badly about telling Mort
> I couldn't understand a word of his excellent book. (It
> does grow on you when you finally find yourself up to
> speed again on some of the mathematical notations.)
>
> Anyhow, I've wanted to understand this. Probably for the
> same reason I've been studying Spanish so diligently
> lately: it just seems like it might come in handy some
> day. Does anyone know if they use PCA in Costa Rica?
>
> PCA *is* fairly straightforward. At least after you
> understand it, it is. Getting there is something else.
> I suspect there are more of you out there like me who
> would appreciate a--shall we say--less mathematical
> approach to the subject.
>
> I found MUCH became clear after reading the above
> mentioned Lindsay Smith tutorial:
>
> http://tinyurl.com/3aaeb6
>
> But there were still a few unresolved problems for me.
> One of these was why there are two ways to do PCA in
> IDL, and why you don't get the same answer when you use
> them. I discovered, eventually, that you DO get the same
> answer, but this took me a whole lot longer to figure
> out than it probably should have.
>
> I also wanted to understand the use of PCA for images,
> so I looked into that a little bit, too.

>
> All this to say that I have written what I am calling
> a PCA Tutorial, although that is probably a lofty title
> for a piece of writing that is more like the blind leading
> the blind. :-)
>
> I would appreciate feedback on this from those of you
> who know a lot more about it than I do. I tried to let
> the Smith tutorial do the heavy theoretical lifting.
> What I wanted to know was how to do this in IDL. So that
> is the focus here. (I did find what I think is an error
> in the Smith tutorial, for what it is worth.)
>
> You can find the tutorial here:
>
>   http://www.dfanning.com/code_tips/pca.html
>
> Any and all comments welcome.
>
> 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.")
