
Subject: Re: singular value decomposition
Posted by [H T Onishi](#) on Thu, 01 Jul 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is not a complete answer but perhaps adds some insight.

First of all I believe that your U from Matlab is in error. The .46 should be -.46.

Second, if you define x as below and then do `svdc,x,w,u,v,/col` and then stuff w into the diagonal elements of a 4x4 array, say ww, and finally compute `u # ww # transpose(v)`, you will get x back.

```
x = u # ww # transpose(v)
```

as it should.

Unfortunately the # operator in IDL does the row/col reversal from "standard" matrix multiplication.

However, if you compute `transpose(v) ## w ## u` then you will also get x back. Since ## computes the "standard" matrix multiply this means that

u from Matlab = `transpose(v)` from IDL

v from Matlab = `transpose(u)` from IDL

both of which are true for the non-singular vectors with the exception of sign differences. I don't think that the sign differences are important since these vectors will span the same subspaces of a 4 dimensional real vector space.

Finally, regarding the column vectors in U from Matlab which do not correspond to the column vectors in `transpose(V)` from IDL, these vectors span the two dimensional null space for this particular linear transformation and they are therefore arbitrary within this null space as long as they are normalized and orthogonal. Another way of putting this is that the two null col vectors in U can be combined linearly to give the two col vectors in `transpose(V)`. Try this out. (This is how I decided that the .46 should be -.46)

Howard Onishi

Dave Bazell <bazell@home.com> wrote in message
news:377AD107.745DA866@home.com...

> I am trying to use the IDL routine SVDC to do principal component
> analysis. In order to understand SVD better I was doing an example I
> found online. However, the IDL SVD routine gives me different results

```

> than the online example.
>
> x = [[1,2],[3,4],[5,6],[7,8]]
>
> matlab, which uses linalg gives (to two decimal places):
>
> [U,S,V] = svd(x) where X = U S transpose(V)
>
> U = .15  .82  -.39  -.38
>      .35  .42   .24  .80
>      .55  .02   .70  .46
>      .74  -.38  -.54  .04
>
> S = 14.3  0
>      0    .62
>
> V = .64  -.77
>      .77  .64
>
> IDL gives
>
> svdc, x,w,u,v,/column
>
> w = 14.2691  0.626828
>
> u = -0.641423  -0.767187
>      -0.767187  0.641423
>      0.00000  0.00000
>      0.00000  0.00000
>
> v = -0.152483  -0.349918  -0.547354  -0.744789
>      0.822647  0.421375  0.0201032  -0.381169
>      0.547723  -0.730297  -0.182574  0.365149
>      0.00000  0.408249  -0.816496  0.408248
>
> clearly the eigenvalues are the same but the u and v matrices are
> exchanged. But what really bothers me is that some values are changed
> from positive to negative. And the IDL V does not have the same values
> as the MATLAB U.
>
> What am I doing wrong? Even if I leave out the /column in the call to
> svdc, I don't get the right answers.
>
> The eigenvalues do not correspond to the eigenvalues returned by the IDL
> routine pcomp which calculates principal components. I thought PCA
> could be done using SVD but I don't see the correspondence.
>
> Any help would be appreciated.

```

>
> Thanks.
>
> Dave
> bazell@home.com
>
