The Great SVD Mystery

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

P312, 2012

The Great SVD Mystery

Introduction

- Earlier in the course, I warned you about how, over the years, people have occasionally gotten into trouble by ignoring various indeterminacies and ambiguities in seemingly well-established quantities.
- In Rencher's homework problem 2.23, and in the author's treatment of the Singular Value Decomposition (SVD), this kind of situation is illustrated beautifully, so I thought we'd digress, have some fun, and discover what went wrong in the author's treatment of the topic.

The SVD

- For a rank k matrix $\boldsymbol{A}$, of order $n \times p$, the *singular value decomposition*, or *SVD*, is a decomposition of A as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}' \tag{1}$$

  where $\boldsymbol{U}$ is $n \times k$, $\boldsymbol{D}$ is $k \times k$, and $\boldsymbol{V}$ is $p \times k$.
- Rencher goes on, as many authors do, to state that $\boldsymbol{D} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$ consists of diagonal elements that are the square root of the non-zero eigenvalues of $\boldsymbol{A}\boldsymbol{A}'$, and $\boldsymbol{U}$ and $\boldsymbol{V}$ are normalized eigenvectors of $\boldsymbol{A}\boldsymbol{A}'$ and $\boldsymbol{A}'\boldsymbol{A}$, respectively, so that, of course, $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{V}'\boldsymbol{V} = \boldsymbol{I}$.
- This would seem to furnish several easy ways to compute the SVD of $\boldsymbol{A}$. For example, the most direct might seem to be to follow Rencher's prescription exactly, using an eigenvalue routine.
- We are fortunate, because R has a routine *svd()* that will provide us with a correct SVD solution.
- Let's try it on problem 2.23 in Rencher.

An Example

Symmetric, but not Square

```
> A <- matrix(c(4,7,-1,8,-5,-2,4,2,-1,3,-3,6),4,3)
> A

     [,1] [,2] [,3]
[1,]    4   -5   -1
[2,]    7   -2    3
[3,]   -1    4   -3
[4,]    8    2    6

> svd1 <- svd(A)
> svd1

$d
[1] 13.161210  6.999892  3.432793

$u
           [,1]       [,2]        [,3]
[1,] -0.2816569  0.7303849 -0.42412326
[2,] -0.5912537  0.1463017 -0.18371213
[3,]  0.2247823 -0.4040717 -0.88586638
[4,] -0.7214994 -0.5309048  0.04012567

$v
           [,1]        [,2]       [,3]
[1,] -0.8557101  0.01464091 -0.5172483
[2,]  0.1555269 -0.94610374 -0.2840759
[3,] -0.4935297 -0.32353262  0.8073135
```

An Example

```
> ## test it out
> ## note that d is provided as a vector
> ## so when computing UDV', need to construct d
> svd1$u %*% diag(svd1$d) %*% t(svd1$v)

     [,1] [,2] [,3]
[1,]    4   -5   -1
[2,]    7   -2    3
[3,]   -1    4   -3
[4,]    8    2    6
```

- It worked perfectly. Now, let's try do reproduce the above SVD using the description in Rencher.

An Example

- We start by taking the eigendecomposition of $\boldsymbol{A}\boldsymbol{A}'$.

```
> decomp <- eigen(A %*% t(A))
> decomp
$values
[1] 1.732174e+02 4.899849e+01 1.178407e+01 8.047847e-15

$vectors
           [,1]       [,2]        [,3]       [,4]
[1,] -0.2816569  0.7303849 -0.42412326  0.4553316
[2,] -0.5912537  0.1463017 -0.18371213 -0.7715340
[3,]  0.2247823 -0.4040717 -0.88586638 -0.0379443
[4,] -0.7214994 -0.5309048  0.04012567  0.4426835
```

- Remember that, because $\boldsymbol{A}$ is only rank 3, we need to grab only the first 3 eigenvectors!

```
> U <- decomp$vectors[,1:3]
```

An Example

- Next we decompose $A'A$

```
> decomp <- eigen(t(A) %*% A)
> decomp
$values
[1] 173.21745  48.99849  11.78407

$vectors
           [,1]        [,2]       [,3]
[1,]  0.8557101 -0.01464091 -0.5172483
[2,] -0.1555269  0.94610374 -0.2840759
[3,]  0.4935297  0.32353262  0.8073135
> V <- decomp$vectors
> D <- diag(sqrt(decomp$values[1:3]))
```

- We are all set to go!
- Let's try it out.

An Example
Oops!

```
> U %*% D %*% t(V)

          [,1]      [,2]      [,3]
[1,] -2.493848  5.827188 -1.350780
[2,] -6.347599  2.358302 -4.018258
[3,]  4.145900 -2.272253 -1.910074
[4,] -8.142495 -2.078259 -5.777596
```

- Oops! This did not work. Why not?
- Let me be even more directive. Here is an approach that does work. We simply calculate $\boldsymbol{V}'$ a different way, that is, $\boldsymbol{V}' = \boldsymbol{D}^{-1}\boldsymbol{U}'\boldsymbol{A}$ and transpose the result.

An Example
Oops!

```
> Vprime <- solve(D) %*% t(U) %*% A
> U %*% D %*% Vprime

     [,1] [,2] [,3]
[1,]    4   -5   -1
[2,]    7   -2    3
[3,]   -1    4   -3
[4,]    8    2    6

> ##It worked!
> V <- t(Vprime)
```

- Why did one approach work, and the other not work? Try to solve the problem before looking at the following slides.

Non-Uniqueness of Eigenvectors

- When authors (like Rencher) speak of "the eigenvectors" of a symmetric matrix, they are mis-characterizing the situation.
- Eigenvectors are unique only up to a reflection, i.e., multiplication by ± 1.
- Consider any $p \times k$ matrix $\boldsymbol{X}$.
- Define a *reflector matrix* $\boldsymbol{R}$ as a diagonal matrix with all diagonal elements equal to ± 1.

Non-Uniqueness of Eigenvectors

- Consider all the possible reflections $\boldsymbol{X}\boldsymbol{R}$ of the columns of $\boldsymbol{X}$.
- There are 2^{k-1} possible reflections of $\boldsymbol{X}$ that are not equal to $\boldsymbol{X}$. Index the reflection by the specific reflector matrix $\boldsymbol{R}_j$.
- Now consider any nonsingular diagonal matrix $\boldsymbol{D}$. It is easy to verify that $\boldsymbol{R}_j\boldsymbol{D}\boldsymbol{R}_j = \boldsymbol{D}$ for any choice of the 2^{k-1} reflector matrices. Moreover, it is also the case that $\boldsymbol{D}^{-1}\boldsymbol{R}_j\boldsymbol{D} = \boldsymbol{R}_j$. On the other hand, for two different reflector matrices $\boldsymbol{R}_j$, $\boldsymbol{R}_k$, it will never be the case that $\boldsymbol{R}_k\boldsymbol{D}\boldsymbol{R}_j = \boldsymbol{D}$.
- So of course, if a symmetric matrix $\boldsymbol{W}$ has an Eckart-Young decomposition $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$, it is also the case that $\boldsymbol{W} = \boldsymbol{V}_j\boldsymbol{D}\boldsymbol{V}_j'$, where $\boldsymbol{V}_j = \boldsymbol{V}\boldsymbol{R}_j$, since $\boldsymbol{V}_j\boldsymbol{D}\boldsymbol{V}_j' = \boldsymbol{V}(\boldsymbol{R}_j\boldsymbol{D}\boldsymbol{R}_j)\boldsymbol{V}' = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$.
- Which specific $\boldsymbol{V}_j$ is generated is a semi-random event that depends on precisely how the program generates the eigenvectors.

Solving the Mystery

- The SVD is not unique either.
- To see why, suppose that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}'$. Then clearly, $\boldsymbol{A} = \boldsymbol{U}_j\boldsymbol{D}\boldsymbol{V}_j'$, where $\boldsymbol{U}_j = \boldsymbol{U}\boldsymbol{R}_j$ and $\boldsymbol{V}_j = \boldsymbol{V}\boldsymbol{R}_j$. Note that the same $\boldsymbol{R}_j$ is applied to both matrices.
- For any valid pair $\boldsymbol{U}$, $\boldsymbol{V}$, there are 2^{k-1} other pairs of the form $\boldsymbol{U}_j$, $\boldsymbol{V}_j$.

Solving the Mystery

- Now we are in a position to see what can go wrong with the solution as described by Rencher. Suppose $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}'$, for a specific $\boldsymbol{U}$ and $\boldsymbol{V}$.
- When you take the eigendecomposition of $\boldsymbol{A}\boldsymbol{A}'$, all you can be sure of is that you obtained eigenvectors $\boldsymbol{U}_j = \boldsymbol{U}\boldsymbol{R}_j$ for *some* $\boldsymbol{R}_j$, with the identity matrix among the possibilities for $\boldsymbol{R}$ in this case, and eigenvalues $\boldsymbol{D}^2$. You can take square roots to obtain $\boldsymbol{D}$, but your $\boldsymbol{U}$ may not be the same as the "correct" $\boldsymbol{U}$.
- When you take the eigendecomposition of $\boldsymbol{A}'\boldsymbol{A}$, your $\boldsymbol{V}\boldsymbol{R}_k$ may not be permuted from the "correct" $\boldsymbol{V}$ by the same $\boldsymbol{R}_j$ that permuted $\boldsymbol{U}$. That is, $\boldsymbol{R}_j$ may not be equal to $\boldsymbol{R}_k$. Suppose you follow Rencher's directions. When you try to reconsitute $\boldsymbol{A}$ from your "solution" as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{R}_j\boldsymbol{D}\boldsymbol{R}_k\boldsymbol{V}'$, you will find it is incorrect (unless you are lucky and $\boldsymbol{R}_j = \boldsymbol{R}_k$).

Solving the Mystery

- On the other hand, suppose you have your $\boldsymbol{U}_j = \boldsymbol{U}\boldsymbol{R}_j$ and the correct $\boldsymbol{D}$. Then, if you compute $\boldsymbol{V}'$ as $\boldsymbol{V}' = \boldsymbol{D}^{-1}\boldsymbol{U}_j'\boldsymbol{A}$, you will obtain

$$
\begin{aligned}
\boldsymbol{V}' &= \boldsymbol{D}^{-1}\boldsymbol{R}_j(\boldsymbol{U}'\boldsymbol{U})\boldsymbol{D}\boldsymbol{V}' && (2)\\
&= (\boldsymbol{D}^{-1}\boldsymbol{R}_j\boldsymbol{D})\boldsymbol{V}' && (3)\\
&= \boldsymbol{R}_j\boldsymbol{V}' && (4)
\end{aligned}
$$

  since $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}$, and $\boldsymbol{D}^{-1}\boldsymbol{R}_j\boldsymbol{D} = \boldsymbol{R}_j$.
- Notice now, with this approach, you obtain a $\boldsymbol{U}$ and $\boldsymbol{V}$ that have, in effect, been permuted by the same $\boldsymbol{R}_j$, so regardless of which permutation the $\boldsymbol{R}_j$ was, this method will produce a correct solution.