STAT 513/413: Lecture 6
Yet another bit of linear algebra: more decompositions

(principal components and many other things)

SVD

Let A be *arbitrary* $p \times q$ matrix

Singular value decomposition (SVD): $A = U\Lambda V^{\mathsf{T}}$

where U and V are orthogonal - $p \times p$ and $q \times q$, respectively

and Λ is $p \times q$ diagonal with diagonal elements

$\lambda_i \geqslant 0$ (singular values)


```
> A = cbind(c(1,2,3),c(2,5,4))
> A
     [,1] [,2]
[1,]    1    2
[2,]    2    5
[3,]    3    4
```

Note: for all what follows, instead of “decomposition” in some other sources you may encounter “factorization”; both mean the same, also for other decompositions/factorizations

In R

```
> sa=svd(A,nu=dim(A,1))
> sa
$d
[1] 7.6203733 0.9643188

$u
           [,1]        [,2]       [,3]
[1,] -0.2932528 -0.08121183 -0.9525793
[2,] -0.7017514 -0.65838502  0.2721655
[3,] -0.6492670  0.74828724  0.1360828

$v
           [,1]       [,2]
[1,] -0.4782649  0.8782156
[2,] -0.8782156 -0.4782649
```

Testing...

```
> t(sa$u) %*% sa$u
              [,1]          [,2]          [,3]
[1,]  1.000000e-00  0.000000e+00 -9.714451e-17
[2,]  0.000000e+00  1.000000e+00 -1.387779e-16
[3,] -9.714451e-17 -1.387779e-16  1.000000e+00
> t(sa$v) %*% sa$v
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> sa$v %*% t(sa$v)
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> sa$u %*% diag(sa$d) %*% t(sa$v)
Error in sa$u %*% diag(sa$d) : non-conformable arguments
> sa$u %*% rbind(diag(sa$d),c(0,0)) %*% t(sa$v)
     [,1] [,2]
[1,]    1    2
[2,]    2    5
[3,]    3    4
```

Economy class

Again, there are two versions of SVD: the one just introduced, and the other, "economy" version, in which:

- if $p \geqslant q$: V as above, Λ as above, but *square*, $q \times q$

and then only first q columns of U are taken: which means that U has orthonormal columns, $U^{\mathsf{T}}U = I$, but is not orthogonal, as UU^{T} may differ from I

- if $p \leqslant q$, then the other way round: Λ is $p \times p$, U is square $p \times p$ and thus orthogonal, and V has orthonormal columns, $V^{\mathsf{T}}V = I$

Economy class in R: testing again

```
> sa=svd(A)
> sa
$d
[1] 7.6203733 0.9643188
$u
           [,1]        [,2]
[1,] -0.2932528 -0.08121183
[2,] -0.7017514 -0.65838502
[3,] -0.6492670  0.74828724
$v
           [,1]       [,2]
[1,] -0.4782649  0.8782156
[2,] -0.8782156 -0.4782649
> sa$u %*% diag(sa$d) %*% t(sa$v)
     [,1] [,2]
[1,]    1    2
[2,]    2    5
[3,]    3    4
```

Testing continued

```
> sa$u %*% t(sa$u)
          [,1]        [,2]        [,3]
[1,] 0.0925926  0.25925926  0.12962963
[2,] 0.2592593  0.92592593 -0.03703704
[3,] 0.1296296 -0.03703704  0.98148148
> t(sa$u) %*% sa$u
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> sa$v %*% t(sa$v)
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> t(sa$v) %*% sa$v
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

What is it good for?

Many things. There are methods that could not be done without SVD, like correspondence analysis

One important application are principal components (for more background on the method, see Rizzo, Section 5.7) - when done directly out of centered (and perhaps scaled) matrix X rather than the variance-covariance matrix

Matrix X: the matrix of data. In the typical form, columns are *variables*. Centered matrix: sums of columns are 0. In matrix notation $\quad 1^{\mathsf{T}}X = 0$, where 1 is a column of 1's here

Note: if sums are 0, then the averages are 0 too. Centering is achieved by subtracting the average of every column. R has a convenient functions for it, `apply` and `sweep`.


```
> xcent = sweep(USArrests,2,apply(USArrests,2,mean))
> sa = svd(xcent)
```

Principal components are usually introduced via the *eigenvalue decomposition* of the (sample) variance-covariance matrix, the latter being equal - *once* X *is centered* - to $\quad \dfrac{1}{n-1}X^{\mathsf{T}}X$

Eigenvalue decomposition

Every *symmetric* (and thus square) matrix S can be written in the form

$$S = \mathrm{Q}\mathrm{L}\mathrm{Q}^{\mathsf{T}}$$

where Q is an orthogonal (thus square) matrix,

and L is a diagonal matrix (apparently, also square)

The matrix L is similar to Λ, but is different: its elements do not have to be nonnegative. (They are guaranteed nonnegative for a special class of matrices called *nonnegative definite* matrices; another special class, *positive definite* matrices, have them all positive)


```
> S=matrix(c(1,3,3,1),nrow=2,ncol=2)
> S
     [,1] [,2]
[1,]    1    3
[2,]    3    1
> eig=eigen(S)
```

Testing: you have to get it right

```
> eig
eigen() decomposition
$values
[1]  4 -2

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
> eig$vectors %*% diag(eig$values) %*% eig$vectors
     [,1] [,2]
[1,]    3   -1
[2,]    1   -3
> t(eig$vectors) %*% diag(eig$values) %*% eig$vectors
     [,1] [,2]
[1,]    1   -3
[2,]   -3    1
> eig$vectors %*% diag(eig$values) %*% t(eig$vectors)
     [,1] [,2]
[1,]    1    3
[2,]    3    1
```

The connection

Instead of doing the eigenvalue decomposition of $\dfrac{1}{n-1}X^\top X$

we just do the SVD of the *centered* (don't forget!) data matrix X:

$$X = U\Lambda V^\top \qquad \text{and then}$$

$$\frac{1}{n-1}X^\top X = \frac{1}{n-1}V\Lambda U^\top U\Lambda V^\top$$

$$= \frac{1}{n-1}V\Lambda\Lambda V^\top = V\left(\frac{1}{n-1}\Lambda^2\right)V^\top = VLV^\top$$

hence we can match $Q = V$ and $\Lambda^2 = (n-1)L$

the eigenvalues are just squares of singular values divided by $n-1$

and eigenvectors (columns of Q) are there in V

the principal components are then

$$U\Lambda = U\Lambda V^\top V = XV$$

their order depending on the magnitude of the corresponding elements on the diagonal of L (usually come already ordered)

First principal component

```
> sa$d
[1] 586.12680  99.48681  45.42598  17.37953
> sa$v
           [,1]        [,2]        [,3]        [,4]
[1,] 0.04170432 -0.04482166  0.07989066 -0.99492173
[2,] 0.99522128 -0.05876003 -0.06756974  0.03893830
[3,] 0.04633575  0.97685748 -0.20054629 -0.05816914
[4,] 0.07515550  0.20071807  0.97408059  0.07232502
> xcent %*% sa$v[,1]
Error in xcent %*% sa$v[, 1] :
  requires numeric/complex matrix/vector arguments
> as.matrix(xcent) %*% sa$v[,1]
                      [,1]
Alabama          64.802164
Alaska           92.827450
Arizona         124.068216
Arkansas         18.340035
...
```

A bit more about eigenvalue decomposition

As we have seen, there is a specific function `eigen()` in R, which comes handy when we have more general matrices to compute the eigenvalue decomposition. In fact, the eigenvalue decomposition is defined again for *every* matrix S, not only for the symmetric ones: however, it guaranteed to be *real* only for the latter - for non-symmetric matrices the eigenvalues and eigenvectors may be complex.

It should be remarked, however, that in statistics, computing eigenvalue decomposition of a non-symmetric matrix is very rarely of interest.

Eigenvalue decomposition, if available, is a useful tool for calculating various matrices and their functions. The general principle is: if you can do something for a matrix that is *diagonal*, by doing it componentwise for the elements of the diagonal, then you can generalize it to the matrices posessing *real* (and also *complex*, if you can do the same operation for complex numbers) eigenvalue decomposition

Examples

Square root of a matrix: if L is diagonal with nonnegative diagonal elements ℓ_i, then a diagonal matrix $L^{1/2}$ with diagonal elements $\sqrt{\ell_i}$ satisfies $L^{1/2}L^{1/2} = A$

Consider now a general A with eigenvalue decomposition $QLQ^\top$. Here L is diagonal, but in order to take square root, all elements must be nonnegative. (Note that there is no square root for complex numbers, but you can still find an appropriate $L^{1/2}$, albeit not uniquely defined. In fact, neither it is uniquely defined in the real case.) This is true when A is symmetric nonnegative definite, which is of particular interest in statistics: all variance-covariance matrices are symmetric nonnegative definite.

So now, $A^{1/2} = QL^{1/2}Q^\top$, as is easy to verify:

$$A^{1/2}A^{1/2} = QL^{1/2}Q^\top QL^{1/2}Q^\top = QL^{1/2}L^{1/2}Q^\top = QLQ^\top = A$$

Another example: inverse matrix. When is a diagonal matrix invertible? What does it mean for a general matrix? How do you define its inverse then?

Matrix functions

A general way of definining functions on matrices is as follows. You have a real function $f(x)$. For a diagonal matrix L with diagonal elements ℓ_i, define $f(L)$ as a diagonal matrix with diagonal elements $f(\ell_i)$. For a general matrix A with eigenvalue decomposition $QLQ^\top$, define

$$e^A = Q\, e^L\, Q^\top$$

Another example of the same principle, albeit not a special case of the previous one: determinant. What is the determinant of a diagonal matrix? How the eigenvalue decomposition can be then used to compute a determinant of a general matrix (for simplicity, with a *real* eigenvalue decomposition)?

Eigenvalues of non-symmetric matrices

```
> S=matrix(c(1,2,3,4),nrow=2,ncol=2)
> eigen(S)
eigen() decomposition
$values
[1]  5.3722813 -0.3722813

$vectors
           [,1]       [,2]
[1,] -0.5657675 -0.9093767
[2,] -0.8245648  0.4159736
> S=matrix(c(3,4,-2,-1),nrow=2,ncol=2)
> eigen(S)
eigen() decomposition
$values
[1] 1+2i 1-2i

$vectors
                     [,1]                 [,2]
[1,] 0.4082483+0.4082483i 0.4082483-0.4082483i
[2,] 0.8164966+0.0000000i 0.8164966+0.0000000i
```

Yet another decomposition: Cholesky

Let S be a *symmetric positive definite* matrix. Then it possesses a Cholesky decomposition (note that it is not uniquely defined)

$$S = R^{\top} R \qquad \text{where } R \text{ is upper triangular}$$

This is how it is in R; there are definitions in the literature requiring $S = WW^{\top}$, where W is *lower* triangular. Of course, how to obtain one from another is obvious (or isn't it?)

The Cholesky decomposition has special dedicated algorithms (in R, function `chol()`); but we can also piece it together via the eigenvalue and QR decompositions

Once we have the Cholesky decomposition, it can be used for many things. For instance, the system $Sx = b$ can be solved via repeated backsolving:

$$\text{first, solve } R^{\top} y = b; \qquad \text{second, solve } Rx = y$$

Also, the inverse matrix can be done in a similar manner