Homework due Nov 13, 2023 10:13 CST

For the following questions use the data loaded with:

```
library(tissuesGeneExpression)
data(tissuesGeneExpression)
```

Important note: When using the SVD in practice it is important to note that the solution to SVD is not unique. This is because $\mathbf{U}\mathbf{D}\mathbf{V}^\top = (-\mathbf{U})\,\mathbf{D}(-\mathbf{V})^\top$. In fact we can flip the sign of each column of $\mathbf{U}$ and as long as we also flip the respective column in $\mathbf{V}$ the decompostion works. Here is R code demonstrating this:

```
s = svd(e)
signflips = sample(c(-1,1),ncol(e),replace=TRUE)
signflips
```

Now we switch the sign of each column and check that we get the same answer. We do this using the function `sweep()`. If x is a matrix and `a` is a vector, then `sweep(x,1,y,FUN="*")` applies the function `FUN` to each row i `FUN(x[i,],a[i])`, in this case `x[i,]*a[i]`. If instead of 1 we use 2, `sweep()` applies this to columns. To learn about `sweep()`, read `?sweep`.

```
newu= sweep(s$u,2,signflips,FUN="*")
newv= sweep(s$v,2,signflips,FUN="*" )
all.equal( s$u %*% diag(s$d) %*% t(s$v), newu %*% diag(s$d) %*% t(newv))
```

This is important to know because different implementations of the SVD algorithm may give different signs, which can lead to the same code resulting in different answers when run in different computer systems.

SVD Exercises #1

1/1 point (graded)

Compute the SVD of `e`:

```
s = svd(e)
```

Now compute the mean of each row:

```
m = rowMeans(e)
```

What is the correlation between the first column of $\mathbf{U}$ and `m`?

-0.9999998 ✓ Answer: -0.9999998

-0.9999998

Explanation

```
cor(s$u[,1],m)
```

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

SVD Exercises #2

1/1 point (graded)

In the above question, we saw how the first column relates to the mean of the rows of `e`. Note that if we change these means, the distances between columns do not change. Here is some R code showing how changing the means does not change the distances:

```
newmeans = rnorm(nrow(e)) ##random values we will add to create new means
newe = e+newmeans ##we change the means
sqrt(crossprod(e[,3]-e[,45]))
sqrt(crossprod(newe[,3]-newe[,45]))
```

So we might as well make the mean of each row 0 since it does not help us approximate the column distances. We will define `y` as the *detrended* `e` and recompute the SVD:

```
y = e - rowMeans(e)
s = svd(y)
```

We showed that $\mathbf{UDV}^\top$ is equal to `y` up to numerical error:

```
resid = y - s$u %*% diag(s$d) %*% t(s$v)
max(abs(resid))
```

The above can be made more efficient in two ways. First, using the `crossprod()` and second not creating a diagonal matrix. Note that in R we can multiply a matrix `x` by vector `a`. The result is a matrix with row `i` equal to `x[i,]*a[i]`. Here is an example to illustrate this.

```
x=matrix(rep(c(1,2),each=5),5,2)
x
x*c(1:5)
```

Note that the above code is actually equivalent to:

```
sweep(x,1,1:5,"*")
```

This means that we don't have to convert `s$d` into a matrix to obtain $\mathbf{DV}^\top$.

Which of the following gives us the same as `diag(s$d)%*%t(s$v)`?

- ○ `s$d %*% t(s$v)[,1]`
- [✓ `s$d * t(s$v)`]
- ○ `t(s$d * s$v)`

○ `s$v * s$d`

✓

Submit You have used 2 of 2 attempts

Answers are displayed within the problem

SVD Exercises #3

0/1 point (graded)

If we define `vd = t(s$d * t(s$v))`, then which of the following is not the same as $\mathbf{UDV}^\top$:

○ `tcrossprod(s$u,vd)`

● `s$u %*% s$d * t(s$v)` ✓

○ `s$u %*% (s$d * t(s$v) )`

○ `s$u %*% t(vd)`

Explanation

Order of operations in R go left to right and `s$u %*% s$d` is multiplying non-conformable matrices

Submit You have used 2 of 2 attempts

Answers are displayed within the problem

SVD Exercises #4

1/1 point (graded)

Let `z = s$d * t(s$v)`. We showed a derivation demonstrating that because $\mathbf{U}$ is orthogonal, the distance between `e[,3]` and `e[,45]` is the same as the distance between `y[,3]` and `y[,45]`, which is the same as `z[,3]` and `z[,45]`:

```
z = s$d * t(s$v)
sqrt(crossprod(e[,3]-e[,45]))
sqrt(crossprod(y[,3]-y[,45]))
sqrt(crossprod(z[,3]-z[,45]))
```

Note that the columns `z` have 189 entries, compared to 22,215 for `e`.

What is the difference (in absolute value) between the actual distance `sqrt(crossprod(e[,3]-e[,45]))` and the approximation using only two dimensions of `z`?

40.62416

✓ Answer: 40.92771

40.62416

Explanation

```
realdistance = sqrt(crossprod(e[,3]-e[,45]))
approxdistance = sqrt(crossprod(z[1:2,3]-z[1:2,45]))
abs(realdistance - approxdistance)
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

SVD Exercises #5

1/1 point (graded)

What is the minimum number of dimensions we need to use for the approximation in SVD Exercises #4 to be within 10% or less?

7

✓ Answer: 7

7

Explanation

```
ks = 1:189
realdistance = sqrt(crossprod(e[,3]-e[,45]))
approxdistances = sapply(ks,function(k){
    sqrt(crossprod(z[1:k,3,drop=FALSE]-z[1:k,45,drop=FALSE] ))
  })
percentdiff = 100*abs(approxdistances - realdistance)/realdistance
plot(ks,percentdiff) ##take a look
min(ks[which(percentdiff < 10)])
```

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

SVD Exercises #6

1/1 point (graded)

Compute distances between sample 3 and all other samples:

```
distances = sqrt(apply(e[,-3]-e[,3],2,crossprod))
```

Recompute this distance using the 2 dimensional approximation.

What is the Spearman correlation between this approximate distance and the actual distance?

0.8598592 ✓ Answer: 0.8598592

0.8598592

Explanation

```
approxdistances = sqrt(apply(z[1:2,-3]-z[1:2,3],2,crossprod))
plot(distances,approxdistances) ##take a look
cor(distances,approxdistances,method="spearman")
```

Note that this shows how just two dimensions can be useful to get an idea about the actual distances.

Submit You have used 1 of 5 attempts

Answers are displayed within the problem