No evidence that PCA is biased: a response to Elhaik

Daniel Shriner

2023-07-11

This document presents an R port of MATLAB code.

The original data matrix is unusual in that centering does not reduce the rank.

```
library(Matrix)
#create original data matrix
A <- matrix(c(1,0,0,0,1,0,0,0,1,0,0,0),byrow=TRUE,nrow=4,ncol=3,
dimnames=list(c("Red","Green","Blue","Black"),c("R","G","B")))
A
```

```
##       R G B
## Red   1 0 0
## Green 0 1 0
## Blue  0 0 1
## Black 0 0 0
```

```
rankMatrix(A)[1]
```

```
## [1] 3
```

```
#center
B <- scale(A,center=TRUE,scale=FALSE)
B
```

```
##           R     G     B
## Red    0.75 -0.25 -0.25
## Green -0.25  0.75 -0.25
## Blue  -0.25 -0.25  0.75
## Black -0.25 -0.25 -0.25
## attr(,"scaled:center")
##    R    G    B
## 0.25 0.25 0.25
```

```
rankMatrix(B)[1]
```

```
## [1] 3
```

```
#estimate marker covariance matrix
C <- cov(B)
C
```

```
##             R           G           B
## R  0.25000000 -0.08333333 -0.08333333
## G -0.08333333  0.25000000 -0.08333333
## B -0.08333333 -0.08333333  0.25000000
```

The following commands show that PCA preserves variance.

```
#variance before PCA
sum(diag(C))
```

```
## [1] 0.75
```

```
#variance after PCA
D <- eigen(C)$values
D
```

```
## [1] 0.33333333 0.33333333 0.08333333
```

```
sum(D)
```

```
## [1] 0.75
```

The following commands show that PCA preserves distances.

```
#calculate PC score matrix T
V <- eigen(C)$vectors
V
```

```
##            [,1]       [,2]       [,3]
## [1,]  0.0000000  0.8164966 -0.5773503
## [2,] -0.7071068 -0.4082483 -0.5773503
## [3,]  0.7071068 -0.4082483 -0.5773503
```

```
T <- B%*%V
T
```

```
##                [,1]          [,2]       [,3]
## Red   -2.775558e-17  8.164966e-01 -0.1443376
## Green -7.071068e-01 -4.082483e-01 -0.1443376
## Blue   7.071068e-01 -4.082483e-01 -0.1443376
## Black -2.775558e-17  1.110223e-16  0.4330127
```

```
#distances before PCA
dist(A,diag=TRUE,upper=TRUE)
```

```
##            Red    Green     Blue    Black
## Red   0.000000 1.414214 1.414214 1.000000
## Green 1.414214 0.000000 1.414214 1.000000
## Blue  1.414214 1.414214 0.000000 1.000000
## Black 1.000000 1.000000 1.000000 0.000000
```

```
#distances after PCA
dist(T,diag=TRUE,upper=TRUE)
```

```
##            Red    Green     Blue    Black
## Red   0.000000 1.414214 1.414214 1.000000
## Green 1.414214 0.000000 1.414214 1.000000
## Blue  1.414214 1.414214 0.000000 1.000000
## Black 1.000000 1.000000 1.000000 0.000000
```

Over the three dimensions of the data, the estimated distances exactly equal the true distances. The foregoing analyses did not involve any dimension reduction. By decoupling PCA from dimension reduction, it can be seen that there is no evidence that the decorrelation accomplished by PCA is biased.

Now consider the two-dimensional subspace of the original data, with Black projected onto the plane defined by Red, Green, and Blue. The rank of the centered data matrix should be two.

```
#solve sqrt((1-x)^2+(0-y)^2+(0-z)^2)=sqrt(2/3) constrained by x=y=z
#Black projects to (1/3,1/3,1/3)
A2 <- matrix(c(1,0,0,0,1,0,0,0,1,1/3,1/3,1/3),byrow=TRUE,nrow=4,ncol=3,
dimnames=list(c("Red","Green","Blue","Black"),c("R","G","B")))
A2
```

```
##               R         G         B
## Red   1.0000000 0.0000000 0.0000000
## Green 0.0000000 1.0000000 0.0000000
## Blue  0.0000000 0.0000000 1.0000000
## Black 0.3333333 0.3333333 0.3333333
```

```
rankMatrix(A2)[1]
```

```
## [1] 3
```

```
B2 <- scale(A2,center=TRUE,scale=FALSE)
B2
```

```
##                R          G          B
## Red    0.6666667 -0.3333333 -0.3333333
## Green -0.3333333  0.6666667 -0.3333333
## Blue  -0.3333333 -0.3333333  0.6666667
## Black  0.0000000  0.0000000  0.0000000
## attr(,"scaled:center")
##         R         G         B
## 0.3333333 0.3333333 0.3333333
```

```
rankMatrix(B2)[1]
```

```
## [1] 2
```

For the two-dimensional subspace, PCA preserves variance.

```
C2 <- cov(B2)
C2
```

```
##            R          G          B
## R  0.2222222 -0.1111111 -0.1111111
## G -0.1111111  0.2222222 -0.1111111
## B -0.1111111 -0.1111111  0.2222222
```

```
#variance before PCA
sum(diag(C2))
```

```
## [1] 0.6666667
```

```
#variance after PCA
D2 <- eigen(C2)$values
sum(D2)
```

```
## [1] 0.6666667
```

For the two-dimensional subspace, PCA preserves distances.

```
V2 <- eigen(C2)$vectors
T2 <- B2%*%V2
T2
```

```
##                [,1]       [,2]          [,3]
## Red   -2.775558e-17  0.8164966 -1.110223e-16
## Green -7.071068e-01 -0.4082483 -2.775558e-17
## Blue   7.071068e-01 -0.4082483  0.000000e+00
## Black  0.000000e+00  0.0000000  0.000000e+00
```

```
#distances before PCA
dist(A2,diag=TRUE,upper=TRUE)
```

```
##             Red     Green      Blue     Black
## Red   0.0000000 1.4142136 1.4142136 0.8164966
## Green 1.4142136 0.0000000 1.4142136 0.8164966
## Blue  1.4142136 1.4142136 0.0000000 0.8164966
## Black 0.8164966 0.8164966 0.8164966 0.0000000
```

```
#distances after PCA
dist(T2,diag=TRUE,upper=TRUE)
```

```
##             Red     Green      Blue     Black
## Red   0.0000000 1.4142136 1.4142136 0.8164966
## Green 1.4142136 0.0000000 1.4142136 0.8164966
## Blue  1.4142136 1.4142136 0.0000000 0.8164966
## Black 0.8164966 0.8164966 0.8164966 0.0000000
```

Over the top two dimensions of the data, the estimated distances exactly equal the true distances. There remains no evidence that PCA is biased.

In contrast, dimension reduction potentially involves a bias-variance tradeoff. For instance, suppose only the top two dimensions after PCA are retained despite a nonzero third eigenvalue.

```
#variance before dimension reduction
sum(diag(C))
```

```
## [1] 0.75
```

```
#variance after dimension reduction
sum(D[1:2])
```

```
## [1] 0.6666667
```

```
#distances before dimension reduction
dist(A,diag=TRUE,upper=TRUE)
```

```
##            Red    Green     Blue    Black
## Red   0.000000 1.414214 1.414214 1.000000
## Green 1.414214 0.000000 1.414214 1.000000
## Blue  1.414214 1.414214 0.000000 1.000000
## Black 1.000000 1.000000 1.000000 0.000000
```

```
#distances after dimension reduction
dist(T[,1:2],diag=TRUE,upper=TRUE)
```

```
##             Red     Green      Blue     Black
## Red   0.0000000 1.4142136 1.4142136 0.8164966
## Green 1.4142136 0.0000000 1.4142136 0.8164966
## Blue  1.4142136 1.4142136 0.0000000 0.8164966
## Black 0.8164966 0.8164966 0.8164966 0.0000000
```

In this instance, the variance over the top two dimensions is less than the variance over all three dimensions, and the distances estimated over the top two dimensions do not equal the original distances over all three dimensions. With real genetic data, analysis of population structure might take the form of partitioning variance into between-population, within-population, and individual components. Distances estimated over a specific subspace may more accurately reflect a specific variance component than the original distances. Whether estimated distances are biased requires an explicit statement of expectations.