

Singular Value Decomposition Demo

Malay Basu (malaykbasu@gmail.com)

2023-11-20

This is a demonstration of singular value decomposition. Check the talk for the theoretical details of the SVD. This demo just shows how the SVD can be used for image compression.

We need to install imager library to open an image.

```
ifelse(require(imager), suppressPackageStartupMessages(library(imager)), BiocManager::install("imager"))

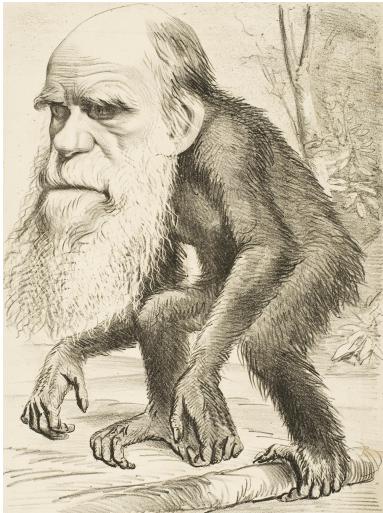
## Loading required package: imager
## Loading required package: magrittr
##
## Attaching package: 'imager'
## The following object is masked from 'package:magrittr':
##       add
## The following objects are masked from 'package:stats':
##       convolve, spectrum
## The following object is masked from 'package:graphics':
##       frame
## The following object is masked from 'package:base':
##       save.image
## [1] "imager"
```

Now we will download an image file for our use.

```
download.file(url = "https://upload.wikimedia.org/wikipedia/commons/6/6f/Editorial_cartoon_depicting_Ch
```

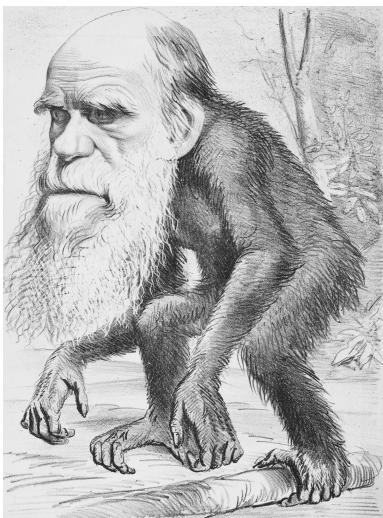
Here is the image that we will use.

```
darwin <- load.image("darwin.jpg")
plot(darwin, axes = F)
```



The original image is a mixture of 3 channels. To find a proper variance for the overall image, we will convert it to grayscale first.

```
darwin_gray <- grayscale(darwin)
plot(darwin_gray, axes=F)
```



Check the dimension of this image.

```
dim(darwin_gray)
```

```
## [1] 2068 2782      1      1
```

Now we'll save the grayscale image as PNG and check the size.

```
save.image(darwin_gray, "darwin_gray_before_svd.png")
```

Check the file size.

```
file.info("darwin_gray_before_svd.png")$size/1024^2
## [1] 3.820333
```

SVD will decompose the image into 3 matrices. The middle diagonal matrix holds the variance information. We will do an SVD of the image and plot a scree plot.

```

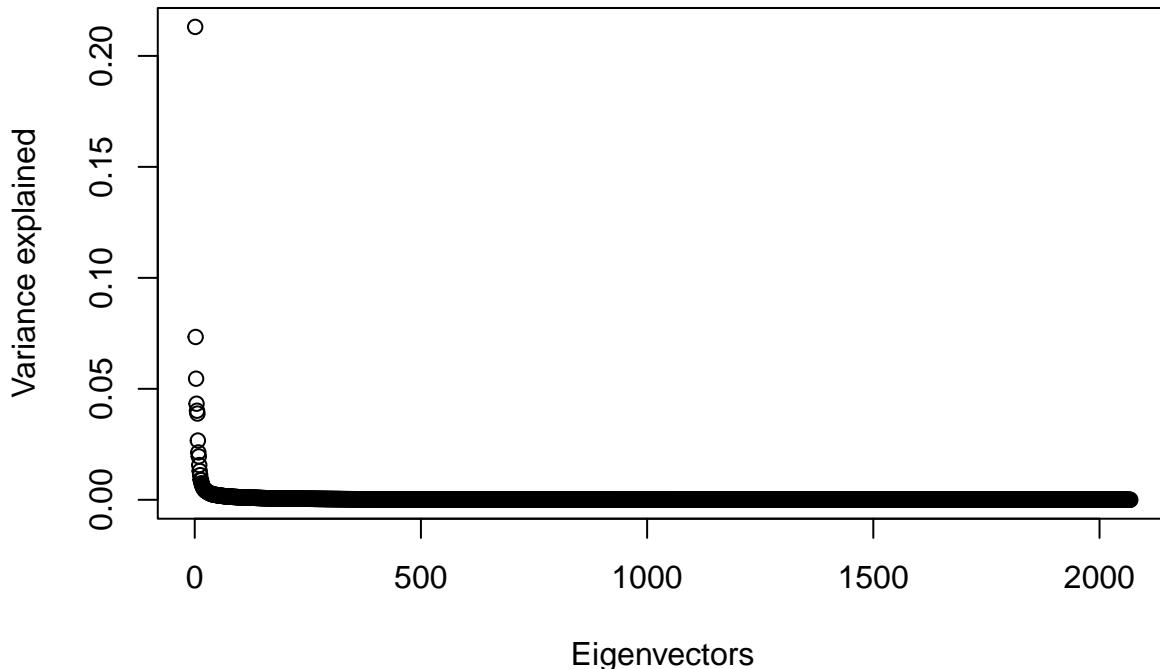
svd1 <- svd(scale(darwin_gray))
str(svd1)

## List of 3
## $ d: num [1:2068] 1107 649 560 499 480 ...
## $ u: num [1:2068, 1:2068] 0.0389 0.0389 0.0388 0.0387 0.0386 ...
## $ v: num [1:2782, 1:2068] 0.00799 0.0084 0.00975 0.00917 0.0085 ...

d is the diagolar vector. Let's plot the scree plot.

plot( svd1$d^2/sum(svd1$d^2), xlab= "Eigenvectors", ylab="Variance explained")

```



Let's plot using various number of SVD.

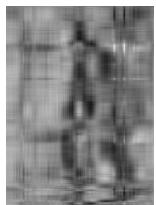
```

par(mfrow=c(2,3))

for (i in c(5, 10, 25, 50, 100, length(svd1$d))) {
  reconstructed <- svd1$u[, 1:i] %*% diag(svd1$d[1:i]) %*% t(svd1$v[, 1:i])
  plot(as.cimg(reconstructed), axes=F, main=paste("Eigenvectors", i, sep=" "))
}

```

Eigenvectors 5



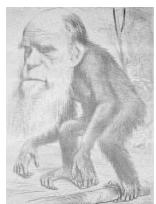
Eigenvectors 10



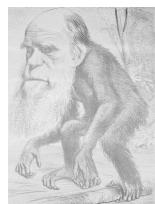
Eigenvectors 25



Eigenvectors 50



Eigenvectors 100

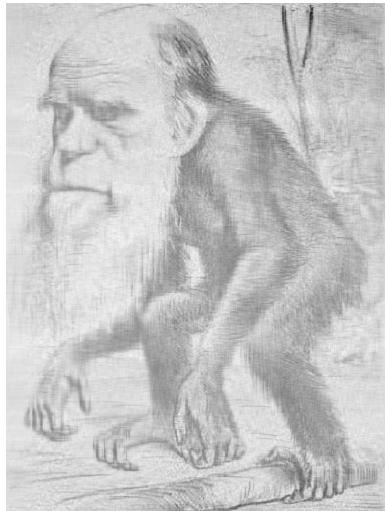


Eigenvectors 2068



We can see that approximately 50 eigenvectors can reconstruct the full image.

```
compressed <- svd1$u[, 1:50] %*% diag(svd1$d[1:50]) %*% t(svd1$v[, 1:50])
plot(as.cimg(compressed), axes=F)
```



```
dim(compressed)
```

```
## [1] 2068 2782
```

Save the file and check the size.

```
save.image(as.cimg(compressed), "darwin_gray_after_svd.png")
```

Size of the file.

```
file.info("darwin_gray_after_svd.png")$size/1024^2
```

```
## [1] 2.142785
```