Lab3 ModernStats

Yuval Benjamini

29 May 2019

In this document I would like to show how to do some basic operations on the fmri data.

The data for the regression:

```
data_sets = load("~/Dropbox/Courses/52525_Modern/52525_2019/Labs/Lab3/fMRIclass_new.RData")
for (i in data_sets){
    print(sprintf("%s, %d x %d", i, dim(get(i))[1],dim(get(i))[2]))
}
```

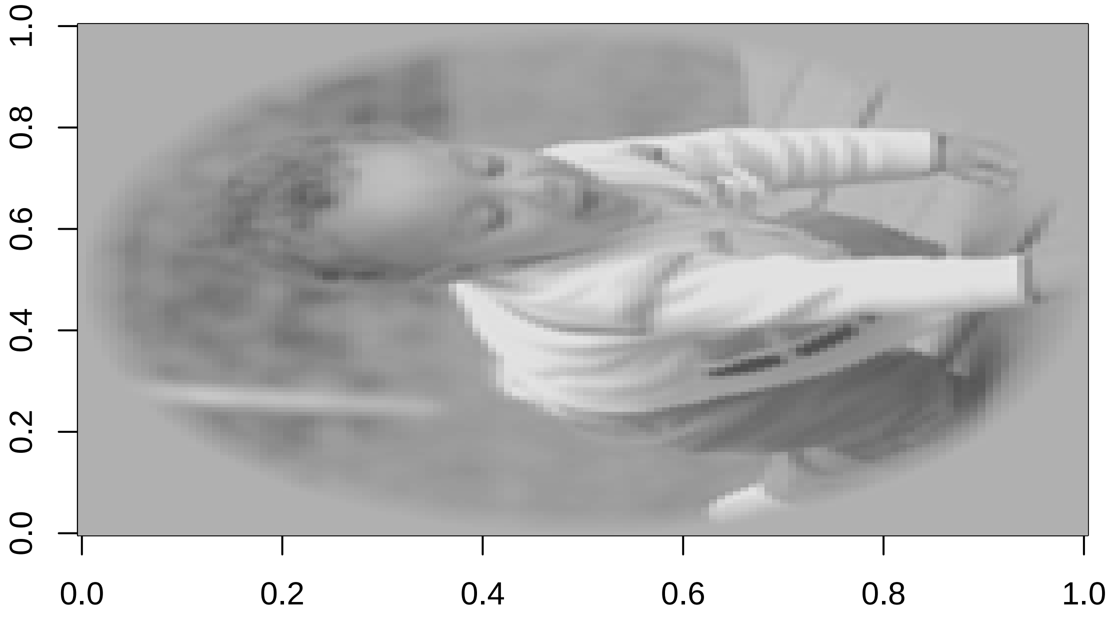
```
## [1] "train_resp, 1750 x 15"
## [1] "feature_train, 1750 x 10921"
## [1] "feature_valid, 120 x 10921"
```

The data sets are as follows:

- train_resp: Training set responses (1750 images, 15 voxels)
- feature_train: Training set features (1750 images, 10921 features)
- feature_valid: Validation set features (120 images, 10921 features)

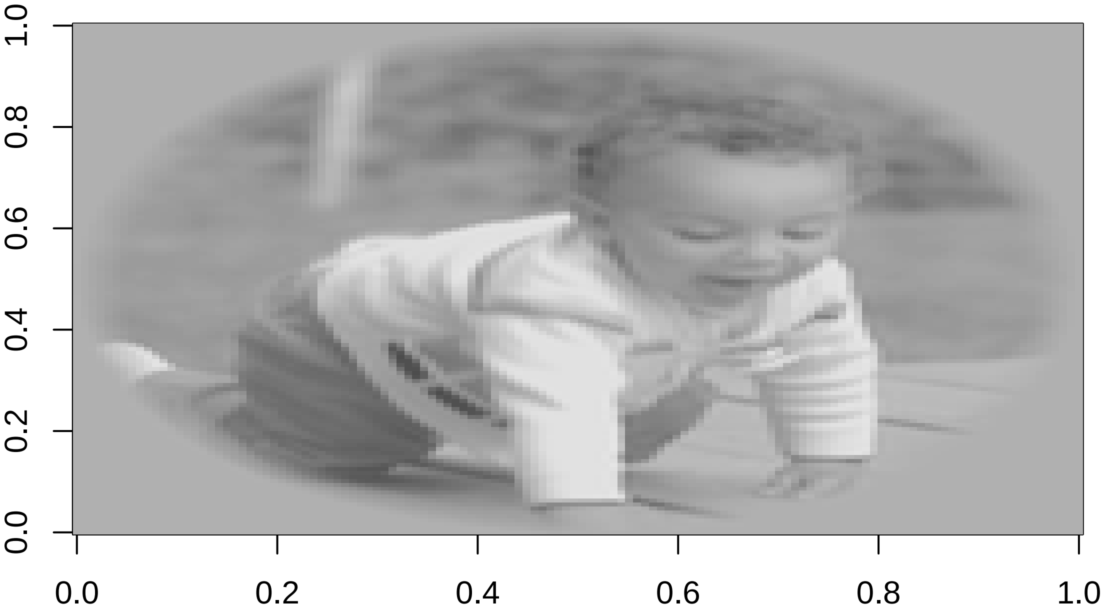
Viewing the images

```
load("stim.RData")
image(matrix(stim[1,], nrow = 128),col = grey.colors(100))
```



We see that images require rotation… Here is the right way:

```
image(t(matrix(stim[1,], nrow = 128)[128:1,]),col = grey.colors(100))
```



Generating the features

The features are computed by taking a pair of wavelets phases and covolving each with the image.

```
load("wavpyr.RData")
```

Note the original images and features were computed in 512x512 resolution, whereas here everything is 128×128, so reconstruction will not be exact.

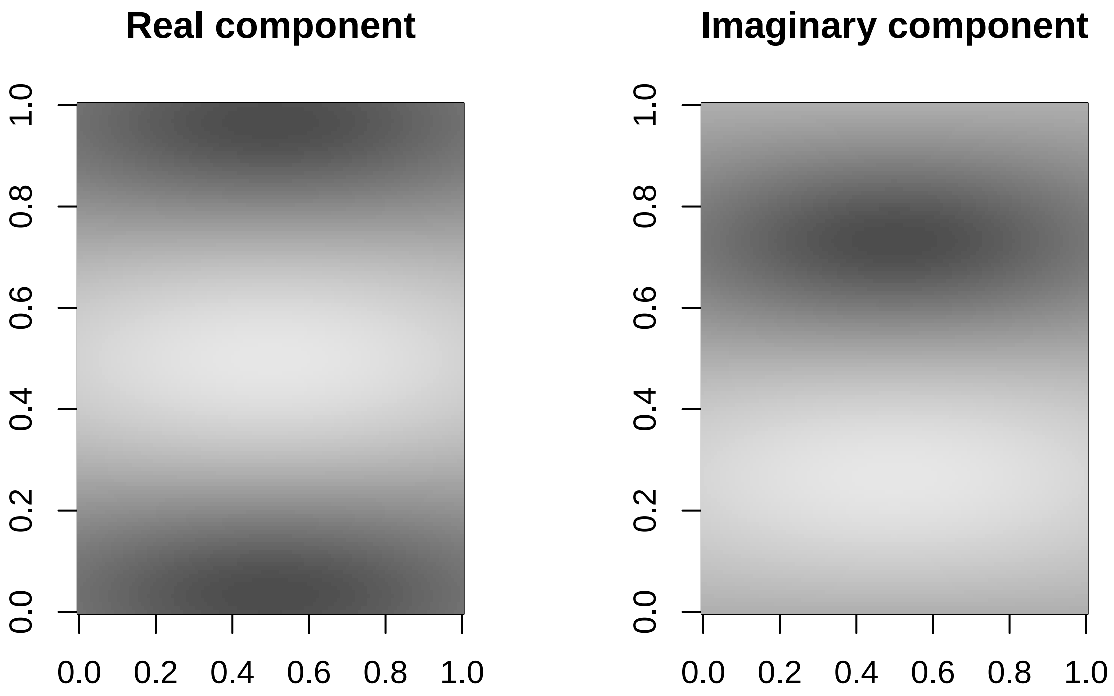
Each wavlet filter is composed of two phase shifted sinusoids multiplied by an exponential.

One of the filters is in the Real component, and the other is the Imaginary component. They have the same orientation, spatial frequency, and center.

```
print(wav.pyr[1:10,2])
```

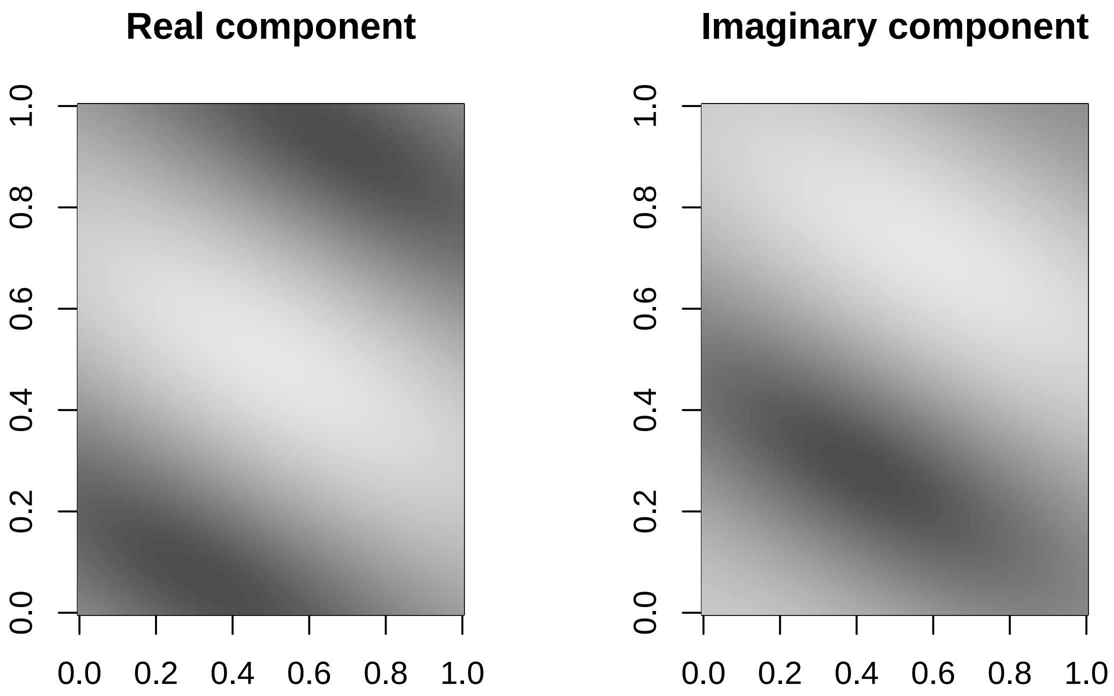
```
##  [1] -0.005162710-0.000111908i -0.005207396-0.000339560i
##  [3] -0.005240373-0.000571830i -0.005261273-0.000808096i
##  [5] -0.005269764-0.001047710i -0.005265548-0.001289997i
##  [7] -0.005248368-0.001534259i -0.005218003-0.001779779i
##  [9] -0.005174277-0.002025821i -0.005117052-0.002271630i
```

```
par(mfcol = c(1,2))
image(t(matrix(Re(wav.pyr[,2]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Real component")
image(t(matrix(Im(wav.pyr[,2]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Imaginary component
```



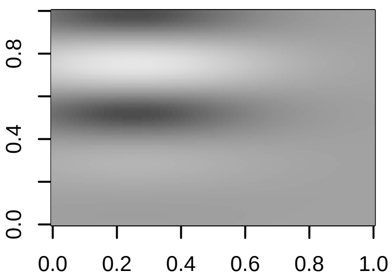
The first 8 features (after the bias term) are just different orientations:

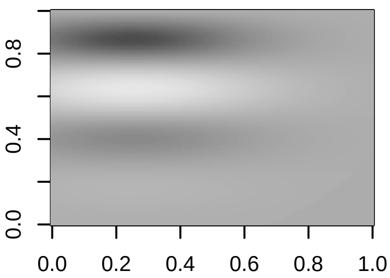
```
par(mfcol = c(1,2))
image(t(matrix(Re(wav.pyr[,9]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Real component")
image(t(matrix(Im(wav.pyr[,9]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Imaginary component
```

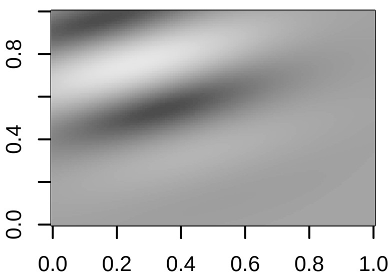


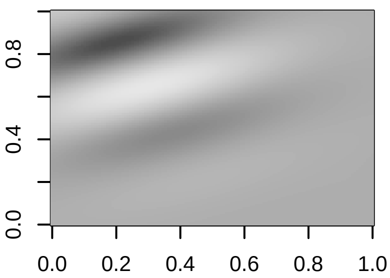
The next 4*8 are of a smaller scale, with 4 different centers and 8 orientations:

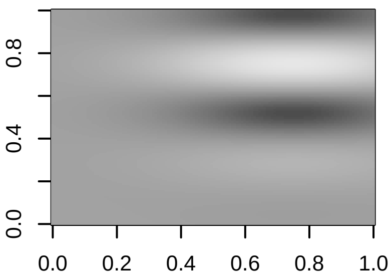
```
par(mfrow = c(3,2))
image(t(matrix(Re(wav.pyr[,10]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Real component")
image(t(matrix(Im(wav.pyr[,10]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Imaginary componen
image(t(matrix(Re(wav.pyr[,11]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Real component")
image(t(matrix(Im(wav.pyr[,11]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Imaginary componen
image(t(matrix(Re(wav.pyr[,18]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Real component")
image(t(matrix(Im(wav.pyr[,18]), nrow = 128)[128:1,]),col = grey.colors(100),main = "Imaginary componen
```

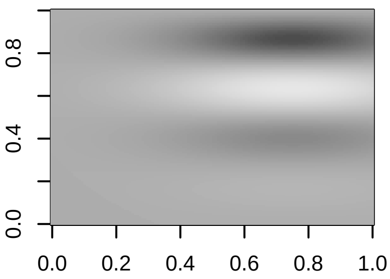

Real component


Imaginary component


Real component


Imaginary component


Real component


Imaginary component

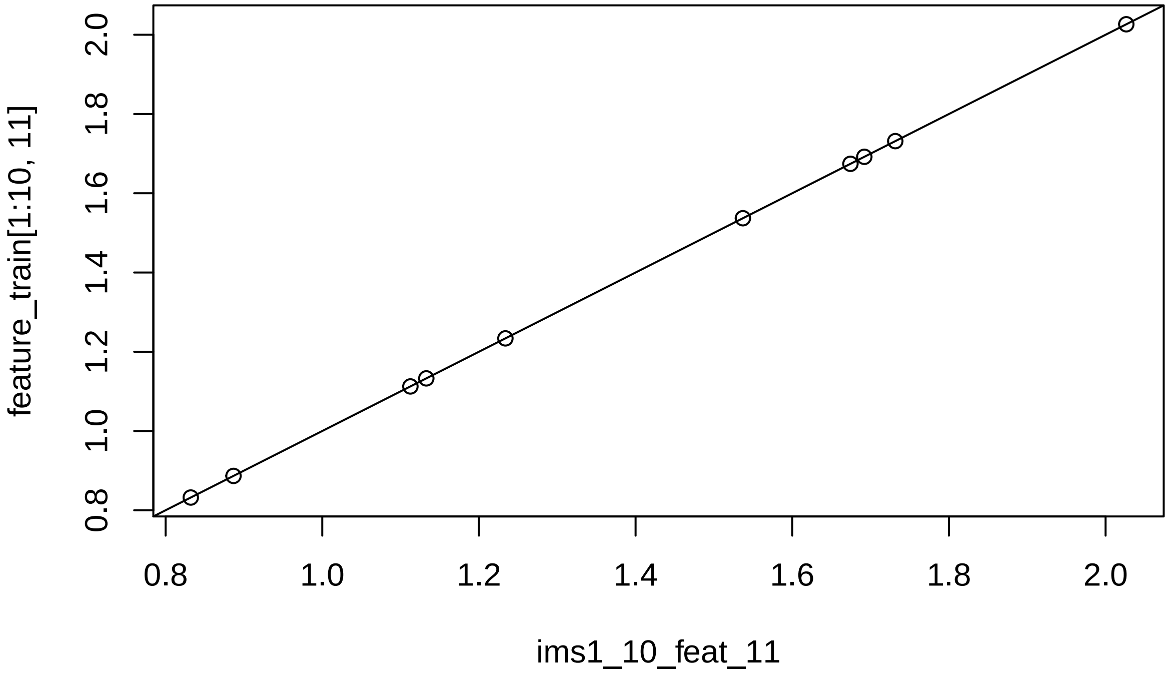
Features computation

The feature computation uses the following equation:

$$X_{ij} = log(|Images_i \cdot WaveletPyramid_j| + 1), \quad |c| = \sqrt{Re^2(c) + Im^2(c)}$$

For example:

```
ims1_10_feat_11 = log(abs(stim[1:10,]%*%wav.pyr[,11])  + 1)
plot(ims1_10_feat_11,feature_train[1:10,11])
abline(0,1)
```


feature_train[1:10, 11]
ims1_10_feat_11