Dimensionality Reduction

Charles Wallis

Motion capsure of Hand postures has been selected for a dimensionality reduction, and I will be performing PCA and LDA dimensionality reduction.

https://archive-beta.ics.uci.edu/ml/datasets/motion+capture+hand+postures The Dataset contains 5 classes of hand postures, and 11 points of tracking on the hand, the markings on hand have a 3d position corresponding to them, from X0, Y0, Z0, up to X11, Y11, Z11 having fewer variables here would mean we would have to guess the hand posture by looking at fewer fingers

```
pos <- read.csv("postures.csv", na.strings = c("?"), header=FALSE)
i <- c(1:17)
pos <- pos[-c(1), i]
pos <- na.omit(pos)
pos[ , i] <- apply(pos[ , i], 2, function(j) as.numeric(as.character(j)))
```

```
set.seed(1234)
i <- sample(1:nrow(pos), nrow(pos)*0.8, replace=FALSE)
train <- pos[i,]
test <- pos[-i,]

str(train)
```

```
## 'data.frame':    59980 obs. of  17 variables:
##  $ V1 : num  2 3 5 5 5 1 2 5 2 5 ...
##  $ V2 : num  10 2 8 13 12 14 4 8 11 2 ...
##  $ V3 : num  59.7 108.1 78 13 -16.4 ...
##  $ V4 : num  141.6 45.9 77.5 106.4 87.3 ...
##  $ V5 : num  33.79 -68.84 -95.29 3.22 10.11 ...
##  $ V6 : num  71 71.5 75.6 -20.2 50.3 ...
##  $ V7 : num  85.8 129 53.3 86.3 87.8 ...
##  $ V8 : num  -1.39 -38.49 -95.1 -31.21 10.81 ...
##  $ V9 : num  100.96 69.36 62.94 7.47 15.39 ...
##  $ V10: num  133.4 80.5 17 130.8 89.5 ...
##  $ V11: num  10.9 -26.7 -82.6 -51.8 19.2 ...
##  $ V12: num  90.5 17.6 89 23.4 76.4 ...
##  $ V13: num  76.9 78.5 83.5 134.1 79.4 ...
##  $ V14: num  -22.8 -50.8 -11.9 -35.9 -10.8 ...
##  $ V15: num  120.7 98 -10.4 51.3 14.5 ...
##  $ V16: num  111.3 26.2 129.8 131.6 140 ...
##  $ V17: num  -15.46 -66.37 3.61 -26.19 -23.79 ...
##  - attr(*, "na.action")= 'omit' Named int [1:3120] 9 42 43 44 51 55 80 87 88 89 ...
##   ..- attr(*, "names")= chr [1:3120] "10" "43" "44" "45" ...
```

PCA

```
library(caret)
```

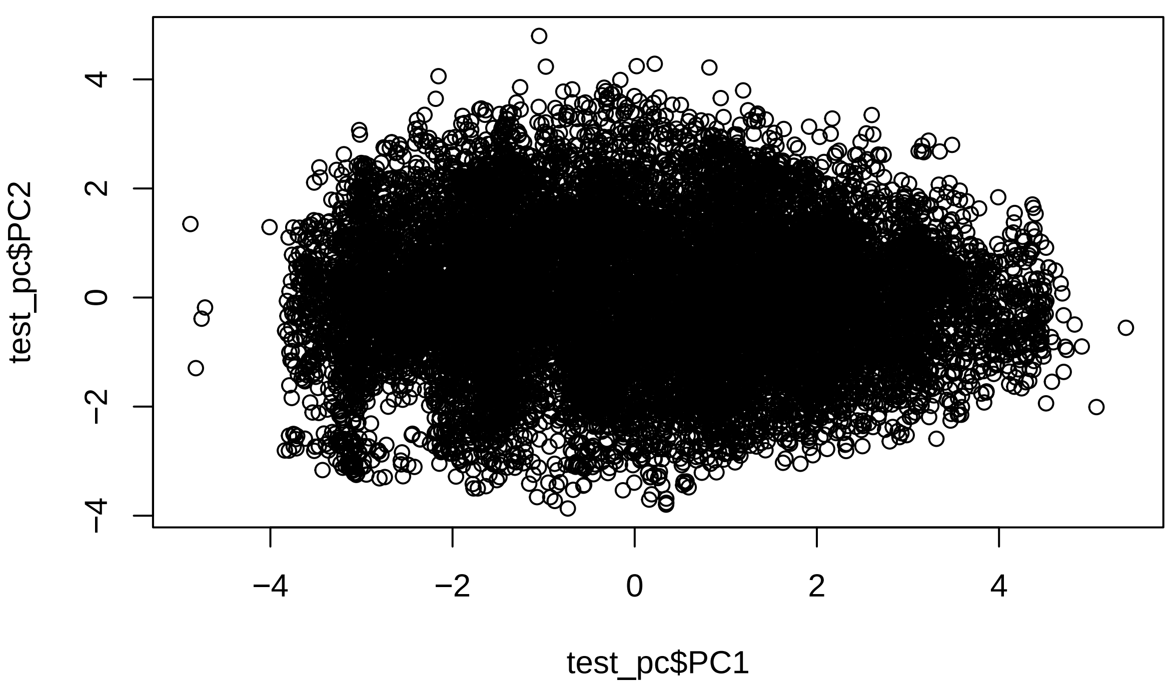
```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
pca_out <- preProcess(train[,1:17], method=c("center", "scale", "pca"))
pca_out
```

```
## Created from 59980 samples and 17 variables
##
## Pre-processing:
##   - centered (17)
##   - ignored (0)
##   - principal component signal extraction (17)
##   - scaled (17)
##
## PCA needed 14 components to capture 95 percent of the variance
```

```
train_pc <- predict(pca_out, train[,])
test_pc <- predict(pca_out, test[,])
plot(test_pc$PC1, test_pc$PC2, pch=c(23,22)[unclass(test_pc$v1)])
```

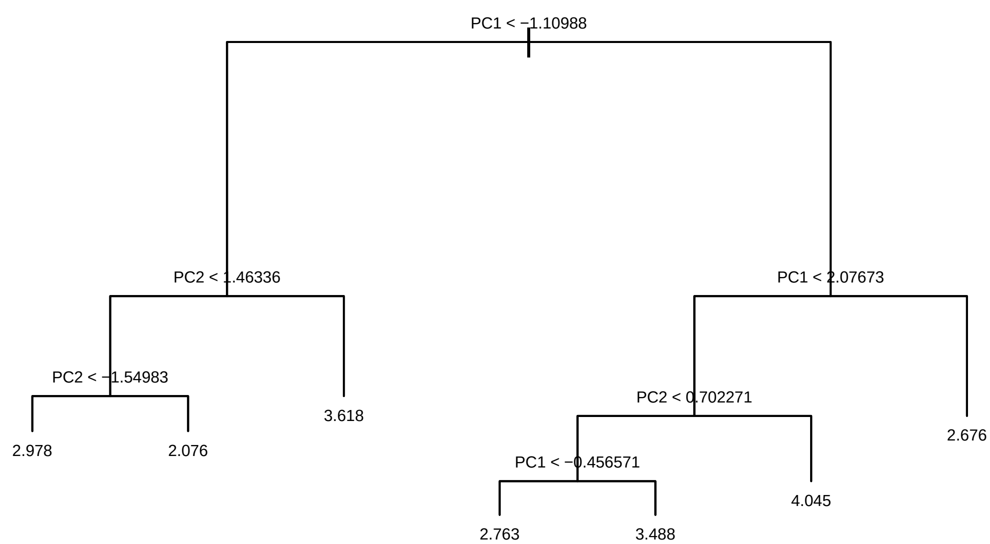


PCA Accuracy

```r
train_df <- data.frame(train_pc$PC1, train_pc$PC2, train$V1)
test_df <- data.frame(test_pc$PC1, test_pc$PC2, test$V1)
library(class)
set.seed(1234)
pred <- knn(train=train_df[,1:2], test=test_df[,1:2], cl=train_df[,3], k=17)
mean(pred==test$V1)
```

```
## [1] 0.5514137
```

```r
library(tree)
colnames(train_df) <- c("PC1", "PC2", "Class")
colnames(test_df) <- c("PC1", "PC2", "Class")
set.seed(1234)
tree1 <- tree(Class~., data=train_df)
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```




```r
pred <- predict(tree1, newdata=test_df, type="vector")
```

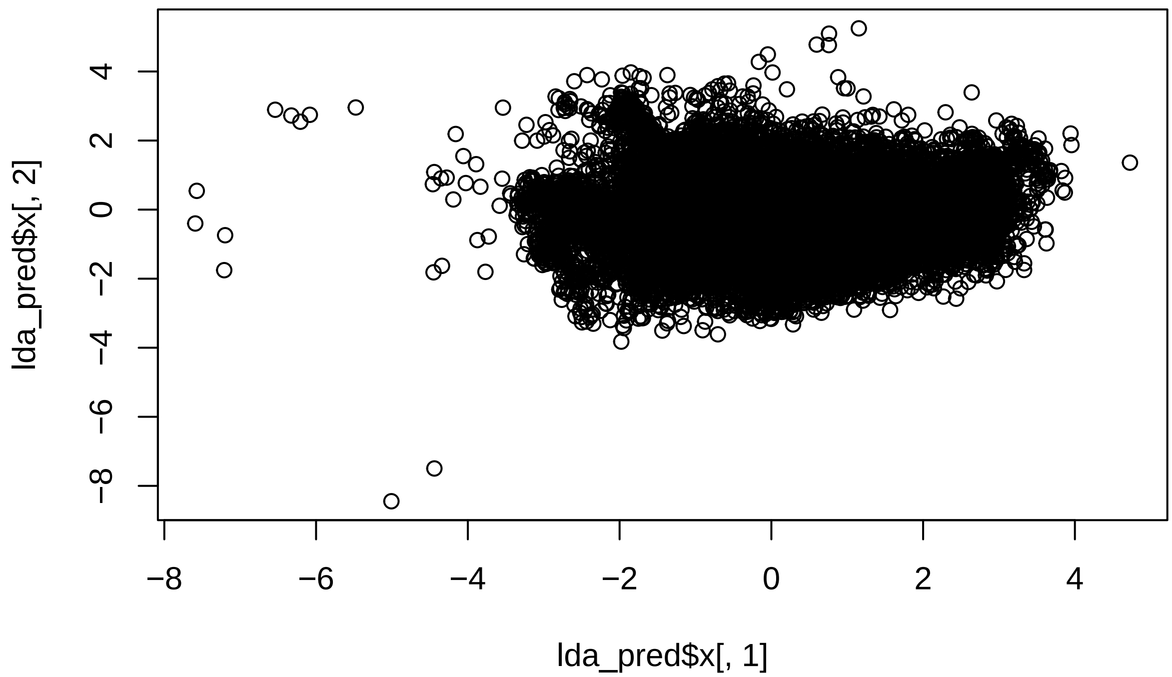
```r
glm1 <- glm(PC1~., data=train_pc, family="gaussian")
summary(glm1)
```

```
##
```

```
## Call:
## glm(formula = PC1 ~ ., family = "gaussian", data = train_pc)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.937  -1.535  -0.077   1.546   5.472
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.115e-16  7.780e-03       0        1
## PC2         -2.966e-15  5.947e-03       0        1
## PC3          1.831e-15  6.399e-03       0        1
## PC4          1.983e-15  6.501e-03       0        1
## PC5          1.693e-15  6.578e-03       0        1
## PC6          5.760e-16  6.679e-03       0        1
## PC7          3.034e-14  7.438e-03       0        1
## PC8         -5.477e-14  8.139e-03       0        1
## PC9          1.292e-14  9.772e-03       0        1
## PC10        -4.305e-15  9.936e-03       0        1
## PC11         8.142e-15  1.011e-02       0        1
## PC12        -2.125e-15  1.017e-02       0        1
## PC13         7.870e-14  1.123e-02       0        1
## PC14         2.561e-14  1.384e-02       0        1
##
## (Dispersion parameter for gaussian family taken to be 3.63018)
##
##     Null deviance: 217687  on 59979  degrees of freedom
## Residual deviance: 217687  on 59966  degrees of freedom
## AIC: 247563
##
## Number of Fisher Scoring iterations: 2
```

LDA

```
library(MASS)
lda1 <- lda(V1~., data=train)
lda_pred <- predict(lda1, newdata=test, type="class")
plot(lda_pred$x[,1], lda_pred$x[,2], pch=c(23,21,22)[unclass(lda_pred$V1)])
```



LDA Accuracy

```
mean(lda_pred$class==test$V1)
```

```
## [1] 0.6192318
```

Analysis

As seen above, the accuracy of PCA logistic regression was 0.55, while LDA method had an accuracy of 0.62. Since PCA is unsupervised and LDA is supervised, we can see that the results are more accurate on LDA since the classes are known