SVM Classification

Spencer Gray

2022-10-23

Data Setup

In this notebook we will try and guess whether the observed data corresponds to a star, galaxy or a quasar. Training the SVM model requires a value target, so we define a factor to each classification. For this example I will assign 0 to star, 1 to galaxy, 2 to quasar. This dataset is also quite large at 100,000 samples so I am going to limit its size to 15,00 with 12,000 in the training set and 3,000 in the test.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(1234)

star <- read.csv("star_classification.csv")

star$class[which(star$class=="STAR")] <- 0
star$class[which(star$class=="GALAXY")] <- 1
star$class[which(star$class=="QSO")] <- 2
star$class <- as.factor(star$class)

train <- sample_n(star, 12000)
test <- sample_n(star, 3000)


dim(star)
```

```
## [1] 100000     18
```

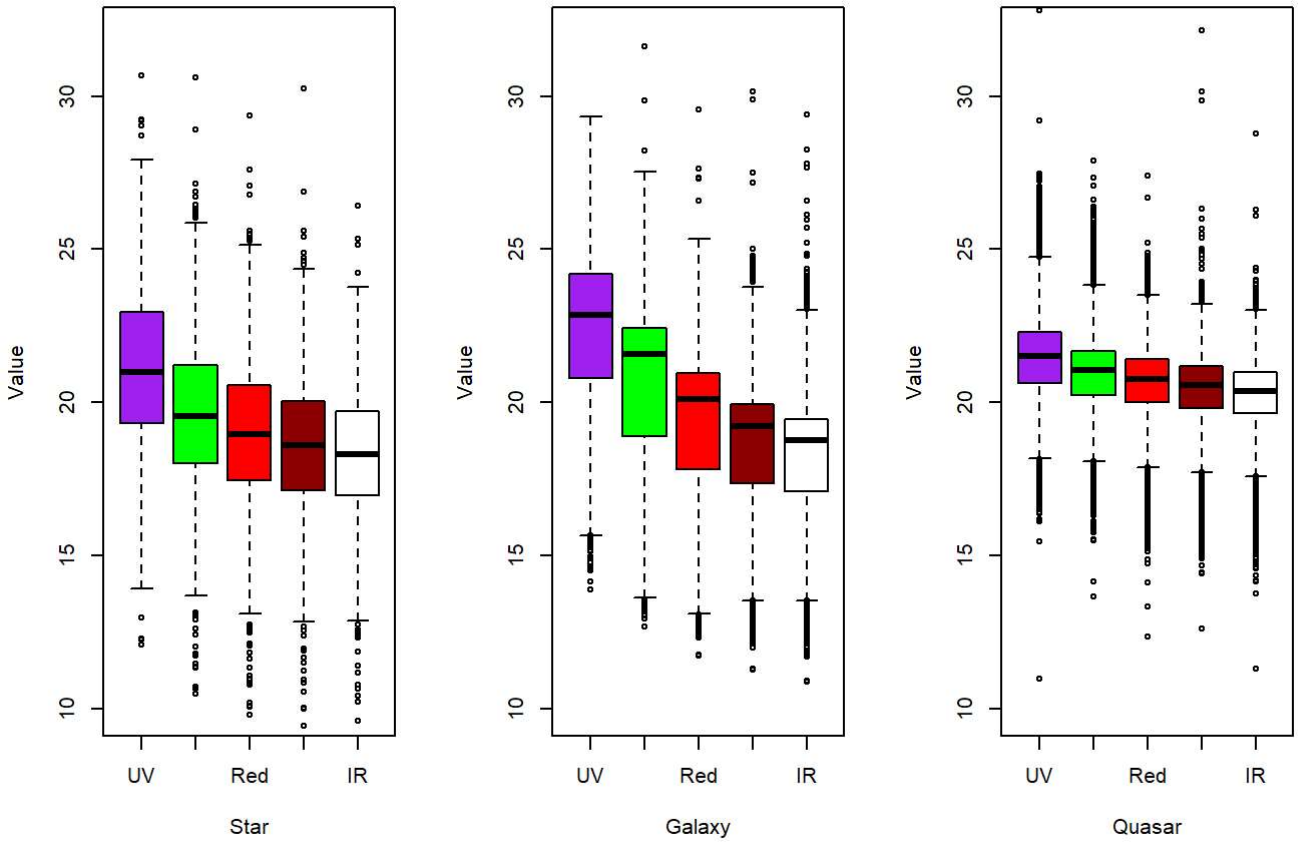
```
head(star)
```

```
##         obj_ID    alpha      delta        u        g        r        i        z
## 1 1.237661e+18 135.6891 32.4946318 23.87882 22.27530 20.39501 19.16573 18.79371
## 2 1.237665e+18 144.8261 31.2741849 24.77759 22.83188 22.58444 21.16812 21.61427
## 3 1.237661e+18 142.1888 35.5824442 25.26307 22.66389 20.60976 19.34857 18.94827
## 4 1.237663e+18 338.7410 -0.4028276 22.13682 23.77656 21.61162 20.50454 19.25010
## 5 1.237680e+18 345.2826 21.1838656 19.43718 17.58028 16.49747 15.97711 15.54461
## 6 1.237680e+18 340.9951 20.5894763 23.48827 23.33776 21.32195 20.25615 19.54544
##   run_ID rerun_ID cam_col field_ID  spec_obj_ID class  redshift plate   MJD
## 1   3606      301       2       79 6.543777e+18     1 0.6347936  5812 56354
## 2   4518      301       5      119 1.176014e+19     1 0.7791360 10445 58158
## 3   3606      301       2      120 5.152200e+18     1 0.6441945  4576 55592
## 4   4192      301       3      214 1.030107e+19     1 0.9323456  9149 58039
## 5   8102      301       3      137 6.891865e+18     1 0.1161227  6121 56187
## 6   8102      301       3      110 5.658977e+18     2 1.4246590  5026 55855
##   fiber_ID
## 1      171
## 2      427
## 3      299
## 4      775
## 5      842
## 6      741
```

Graphing Color Channels

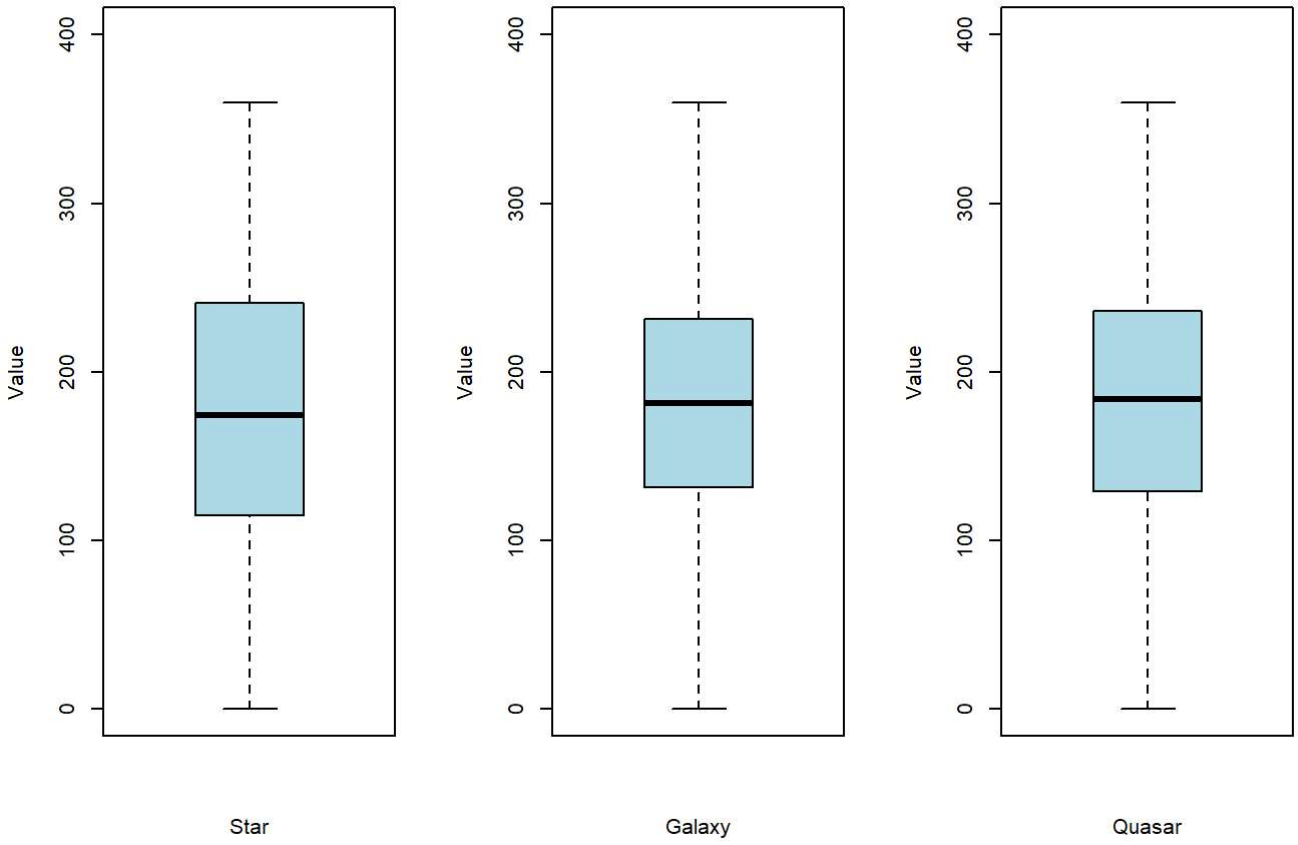
Boxplot of UV, Green, Red, Near IR, and IR channels of quasars, stars. We could be seeing a more tight box with Quasars due to their rarity (less samples) relative to galaxies and stars.

```
par(mfrow=c(1,3))

boxplot(star$u[star$class == 0], star$g[star$class == 0], star$r[star$class == 0], star$i[star$c
lass == 0], star$z[star$class == 0], names=c("UV", "Green", "Red", "Near IR", "IR"), ylim= c(10,
32), ylab=("Value"), xlab=("Star"), col=c("purple", "green", "red", "darkred", "white"))

boxplot(star$u[star$class == 1], star$g[star$class == 1], star$r[star$class == 1], star$i[star$c
lass == 1], star$z[star$class == 1], names=c("UV", "Green", "Red", "Near IR", "IR"), ylim= c(10,
32), ylab=("Value"), xlab=("Galaxy"), col=c("purple", "green", "red", "darkred", "white"))

boxplot(star$u[star$class == 2], star$g[star$class == 2], star$r[star$class == 2], star$i[star$c
lass == 2], star$z[star$class == 2], names=c("UV", "Green", "Red", "Near IR", "IR"), ylim= c(10,
32), ylab=("Value"), xlab=("Quasar"), col=c("purple", "green", "red", "darkred", "white"))
```



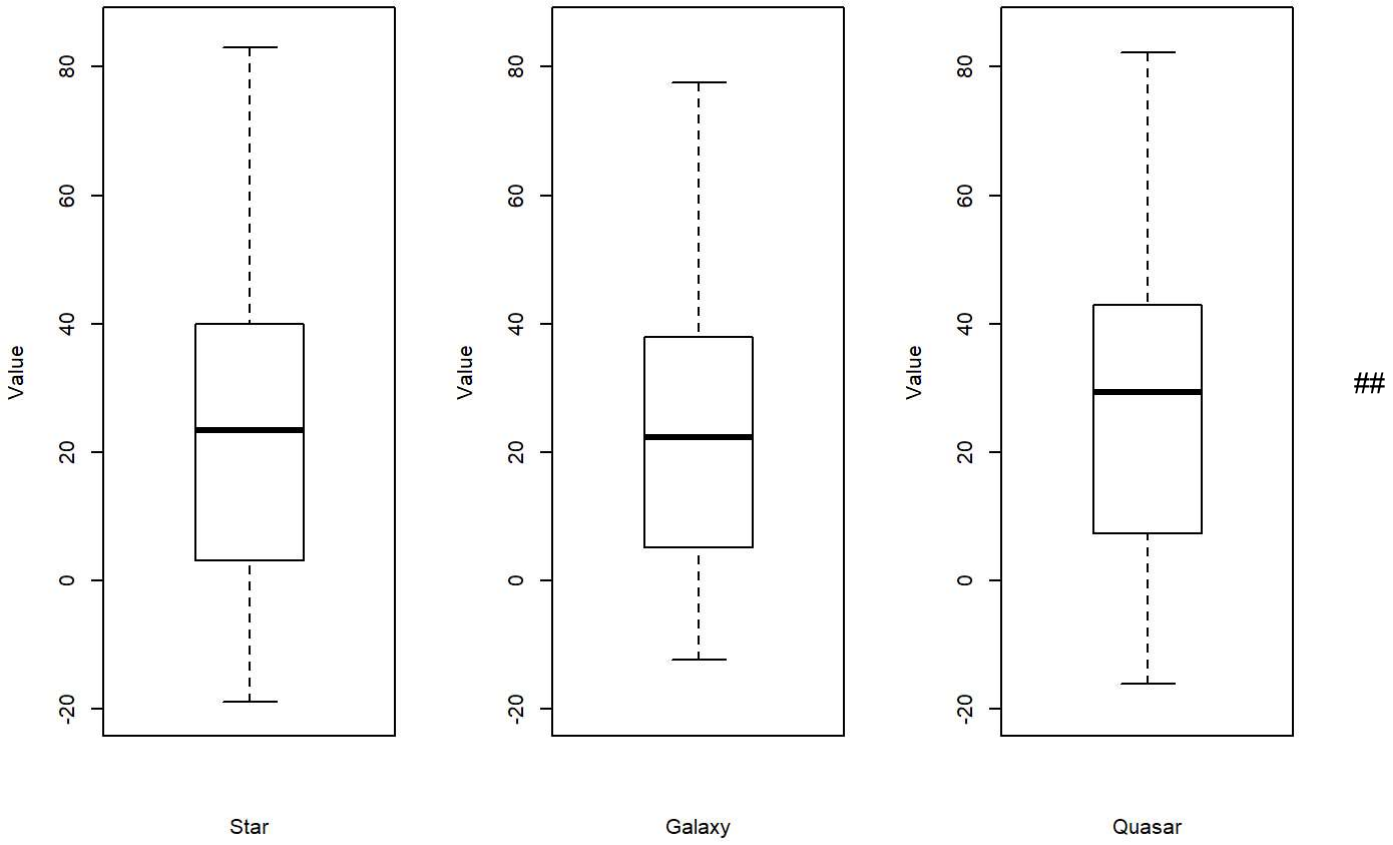
Graphing Alpha

```
par(mfrow=c(1,3))
boxplot(star$alpha[star$class == 0], names=c("Alpha"), ylim= c(0, 400), ylab=("Value"), xlab=("S
tar"), col=("lightblue"))
boxplot(star$alpha[star$class == 1], names=c("Alpha"), ylim= c(0, 400), ylab=("Value"), xlab=("G
alaxy"), col=("lightblue"))
boxplot(star$alpha[star$class == 2], names=c("Alpha"), ylim= c(0, 400), ylab=("Value"), xlab=("Q
uasar"), col=("lightblue"))
```



Graphing Delta

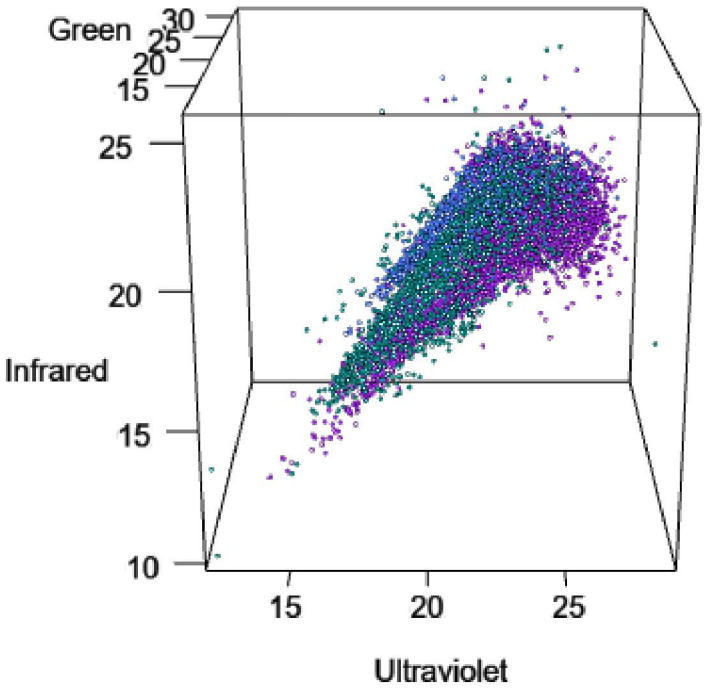
```
par(mfrow=c(1,3))
boxplot(star$delta[star$class == 0], names=c("Delta"), ylim= c(-20, 85), ylab=("Value"), xlab=(
"Star"), col=("white"))
boxplot(star$delta[star$class == 1], names=c("Delta"), ylim= c(-20, 85), ylab=("Value"), xlab=(
"Galaxy"), col=("white"))
boxplot(star$delta[star$class == 2], names=c("Delta"), ylim= c(-20, 85), ylab=("Value"), xlab=(
"Quasar"), col=("white"))
```



##

3D Graph of Red, Green, IR of Each Subject

```
library(rgl)
options(rgl.useNULL=TRUE)
colors <- c('darkcyan', 'purple', 'royalblue1')

plot3d(
  x=train$'u', y=train$'g', z=train$'i',
  col = colors[as.numeric(train$class)],
  type='s',
  radius=0.1,
  xlab="Ultraviolet", ylab="Green", zlab="Infrared"
)
rglwidget()
```



SVM Training

```
library(e1071)
set.seed(1234)

starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="linear", cost=1, s
cale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.783
```

Tuning SVM

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="linear", cost=1, s
cale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.783
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="polynomial", cost=
1, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.7593333
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=1, s
cale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8346667
```

Radial had the best results at 83.5% accuracy, so we use use it in our final model.

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=0.00
1, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.5993333
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=0.01
, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.7066667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=0.1,
scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.7966667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=1, s
cale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8346667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8616667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=100,
scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8693333
```

Tuning shows us that the performance keeps on significantly improving but for the sake of not having insane training times we are going to stick with 10 despite the 0.8% gain from going 10 to 100.

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=0.001, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.7846667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=0.01, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8116667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=0.1, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8556667
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=1, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.871
```

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=5, scale = TRUE)
pred <- predict(starSVM, newdata=test)
mean(pred==test$class)
```

```
## [1] 0.8436667
```

Gamma of 1 appears to have the best results.

Final Model

Calculating the correlation, mean squared error and the table of our svm model.

```
starSVM <- svm(class ~ alpha + delta + u + g + r + i + z, data=train, kernel="radial", cost=10,
 gamma=1, scale = TRUE)
pred <- predict(starSVM, newdata=test)
table(pred, test$class)
```

```
##
## pred    0    1    2
##    0  465   44   42
##    1   95 1701   73
##    2   80   53  447
```

```
mean(pred==test$class)
```

```
## [1] 0.871
```

Conclusion

From these results we see that a radial kernel fits our data the strongest in addition with some fine tuning of the cost and the gamma. This tells us that the categories of our data, galaxy, star, and quasar are grouped in spheres with respect to their alpha, delta channels as well as their electromagnetic spectrogram.