

Support Vector Machine Decision Boundary

JaimeGoB

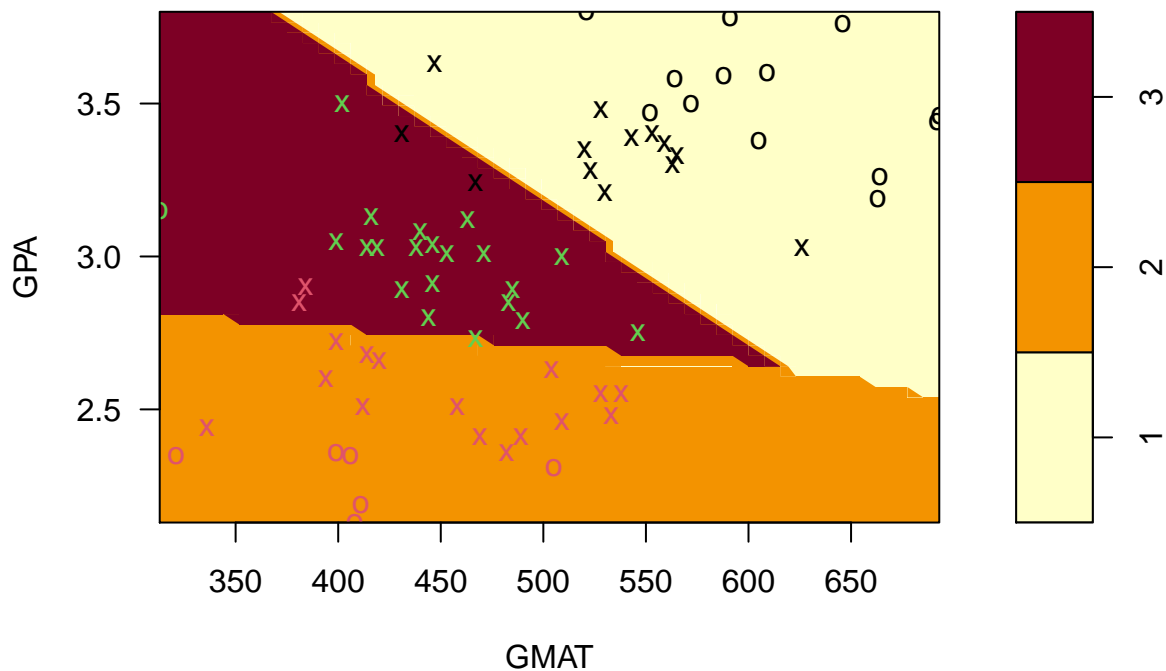
11/28/2020

2. Consider the business school admission data from Mini Project 2. Split the data into training and test sets just as in that project.

(a) i) Fit a support vector classifier to the training data with cost parameter chosen optimally using 10-fold cross-validation. Summarize key features of the fit. ii) Evaluate its performance on the test data. iii) Summarize your results.

i) Fit a support vector classifier to the training data with cost parameter chosen optimally using 10-fold cross-validation After performing a 10-Fold CV. I stored the optimal cost parameter and used it to fit a Support Vector Classifier. The decision boundaries from different Groups can be seen in the

SVM classification plot



plot.

```
##  
## Call:  
## svm(formula = Group ~ ., data = train, kernel = "linear", cost = optimal_cost_svc,  
##       scale = TRUE)  
##  
##  
## Parameters:  
##   SVM-Type:  C-classification
```

```
## SVM-Kernel: linear
## cost: 0.1
##
## Number of Support Vectors: 50
##
## ( 13 17 20 )
##
##
## Number of Classes: 3
##
## Levels:
## 1 2 3
```

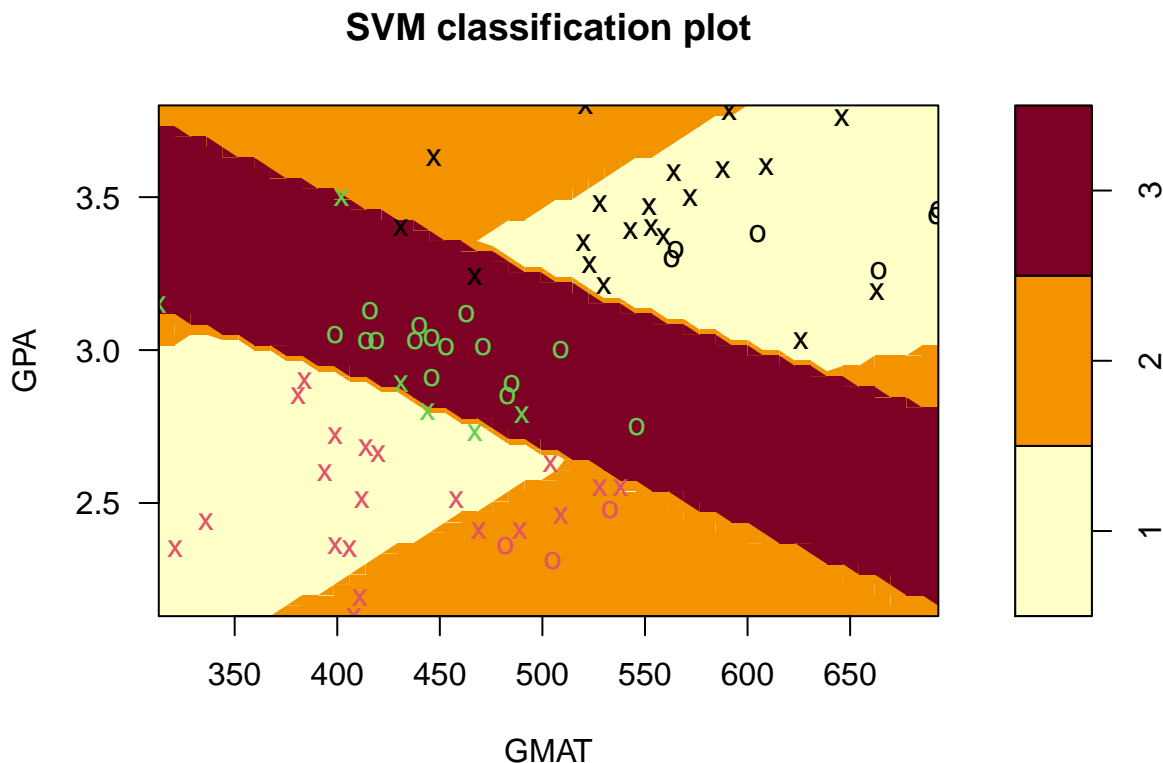
ii) Evaluate its performance on the test data

```
## true
## pred 1 2 3
## 1 3 0 0
## 2 0 5 0
## 3 2 0 5
```

iii) **Summarize your results.** From the confusion matrix it is evident to see that there are only two observations from Group 1 are missclassified as Group 3.

(b) Repeat (a) using a support vector machine with a polynomial kernel of degree two.

i) **Fit a support vector machine to the training data with cost parameter chosen optimally using 10-fold cross-validation** After performing a 10-Fold CV. I stored the optimal cost parameter and used it to fit a Support Vector Machine with polynomial kernel degree 2. The decision boundaries from different Groups can be seen in the plot.



```
##
## Call:
## svm(formula = Group ~ ., data = train, kernel = "polynomial", cost = optimal_cost_svm_polynomial,
##      degree = 2, scale = TRUE)
##
##
## Parameters:
##   SVM-Type:  C-classification
## SVM-Kernel:  polynomial
##      cost:   100
##    degree:   2
##   coef.0:   0
##
## Number of Support Vectors:  46
##
## ( 20 20 6 )
##
##
## Number of Classes:  3
##
## Levels:
##  1 2 3
```

ii) Evaluate its performance on the test data

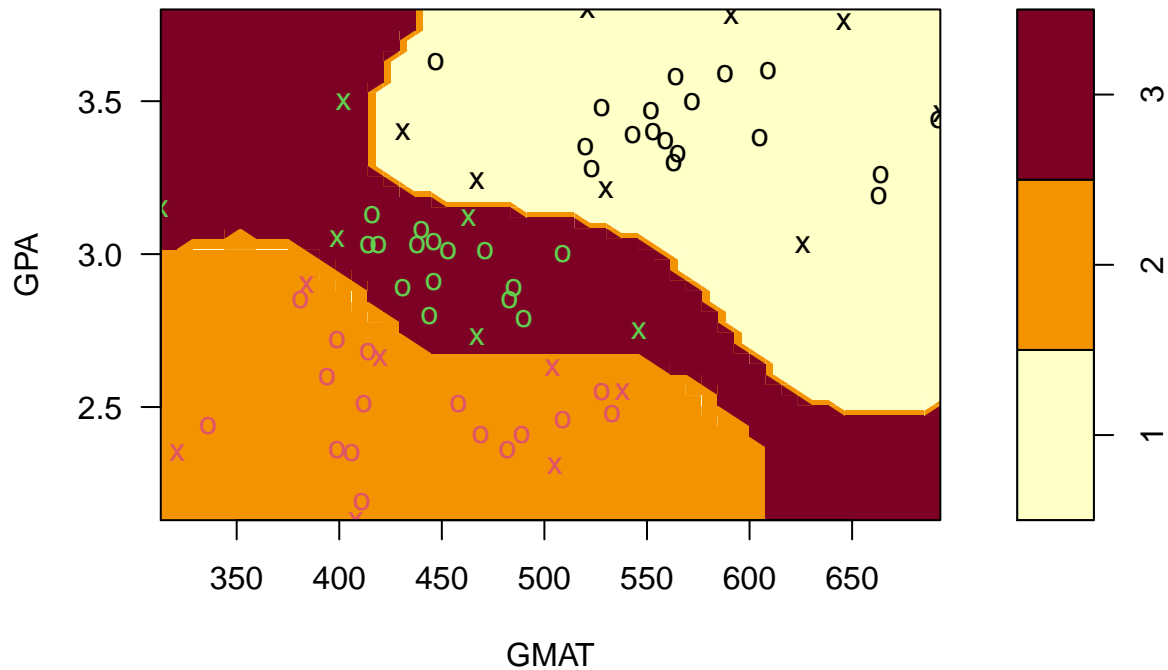
```
##      true
## pred 1 2 3
##      1 1 2 0
##      2 1 2 0
##      3 3 1 5
```

iii) **Summarize your results.** From the confusion matrix it is evident to see that there are many observations that are misclassified. There are only 8 observations from test correctly classified and 7 are misclassified. The SVM with kernel degree 2 does slightly better than random guessing.

(c) Repeat (a) using a support vector machine with a radial kernel with both Gamma and cost parameter chosen optimally.

i) **Fit a support vector machine to the training data with cost parameter chosen optimally using 10-fold cross-validation** After performing a 10-Fold CV. I stored the optimal cost parameter and gamma parameter and used it to fit a Support Vector Machine. The decision boundaries from different Groups can be seen in the plot.

SVM classification plot



```
##
## Call:
## svm(formula = Group ~ ., data = train, kernel = "radial", gamma = optimal_gamma_svm_radial,
##      cost = optimal_cost_svm_radial, scale = TRUE)
##
##
## Parameters:
##   SVM-Type:  C-classification
## SVM-Kernel:  radial
##      cost:   10
##
## Number of Support Vectors:  21
##
## ( 8 7 6 )
##
##
## Number of Classes:  3
##
## Levels:
##  1 2 3
```

ii) Evaluate its performance on the test data

```
##      true
## pred 1 2 3
##      1 4 0 0
##      2 0 5 0
##      3 1 0 5
```

iii) **Summarize your results.** From the confusion matrix it is evident to see that there is only one misclassified observation from Group 1 misclassified as Group 3.

(d) **Compare results from the above three methods and also the method you recommended for these data in Mini Project 2. Which method would you recommend now? For this method, display the data with decision boundary superimposed.** From Mini project 2. Using QDA the observations that are misclassified are only 1 similar to SVM - Radial Decision boundary. Since the both methods provide the exact results I would recommend both SVM - Radial Decision boundary and Quadratic Discriminant Analysis.

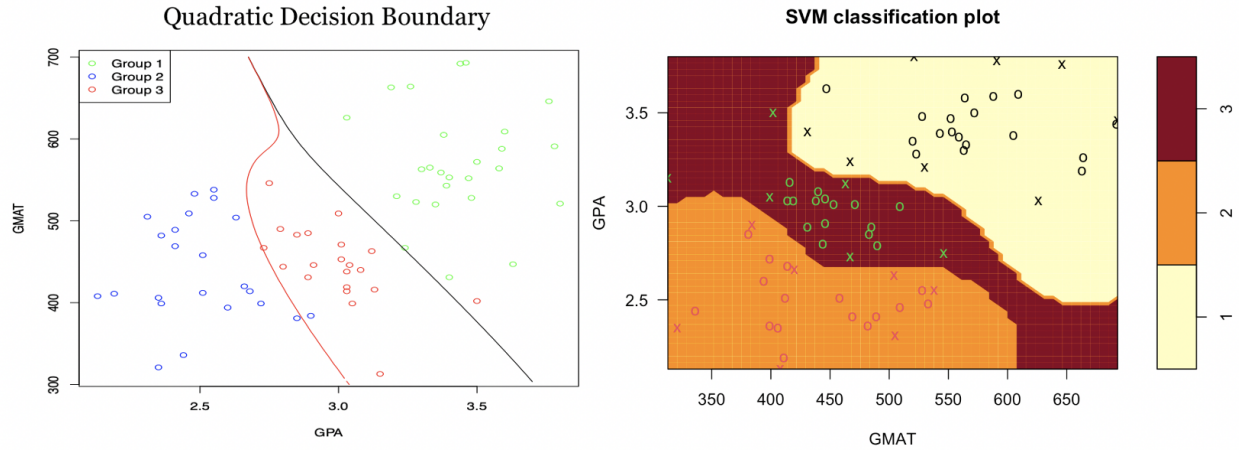


Figure 1: QDA vs SVM - Radial Kernel