Aufgabe 3.2

Tobias Janßen

08.06.2021

Support Vector Machines (SVM)

a)

Bestimmung einer wesentlichen Teilmenge einer Stichprobe -> Support Vektoren
Lernphase -> konvexe quadratische Optimierung
Klassentrennung durch Hyperebene mit neutralem Rand (Margin)
Hyperebene: <w,x>+b = 0 | w: Gewicht
Margin: <w,x>+b = 1|-1
Breite das Rands maximieren unter Bedingung der Hyperebene als Trennebene

Problem:
Rauschen -> Soft Margin-Prinzip (Schlupfvariable|Strafterm)
Nicht Linear -> Projektion in höhere Dimensionen (Kernel-Prinzip|Funktion)

```
library(e1071)
library(datasets)
data(iris)
attach(iris)
```

b)

```
SVM = svm(Species ~., data = iris, kernel = 'linear')
summary(SVM)
```

```
##
## Call:
## svm(formula = Species ~ ., data = iris, kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  29
##
##  ( 2 15 12 )
##
##
```

```
## Number of Classes:  3
##
## Levels:
##  setosa versicolor virginica
```

3 Klassen
29 Stützvektoren
- 2 in der ersen Klasse, 15 in der Zweiten und 12 in der Dritten

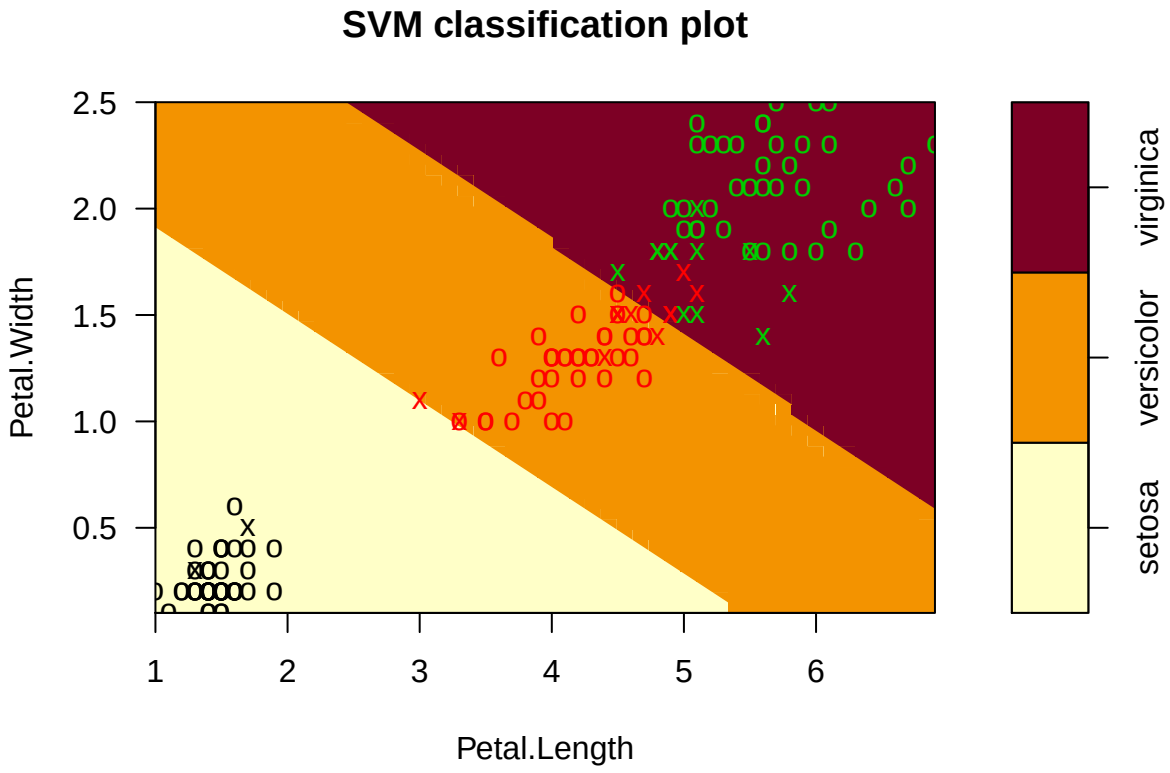
c)

```
pred = predict(SVM, iris)
print(table(pred,Truth = iris$Species))
```

```
##             Truth
## pred         setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         46         1
##   virginica       0          4        49
```

d)

```
plot(SVM,data = iris,Petal.Width~Petal.Length,slice = list(Sepal.Width=3, Sepal.Length=4))
```



e)

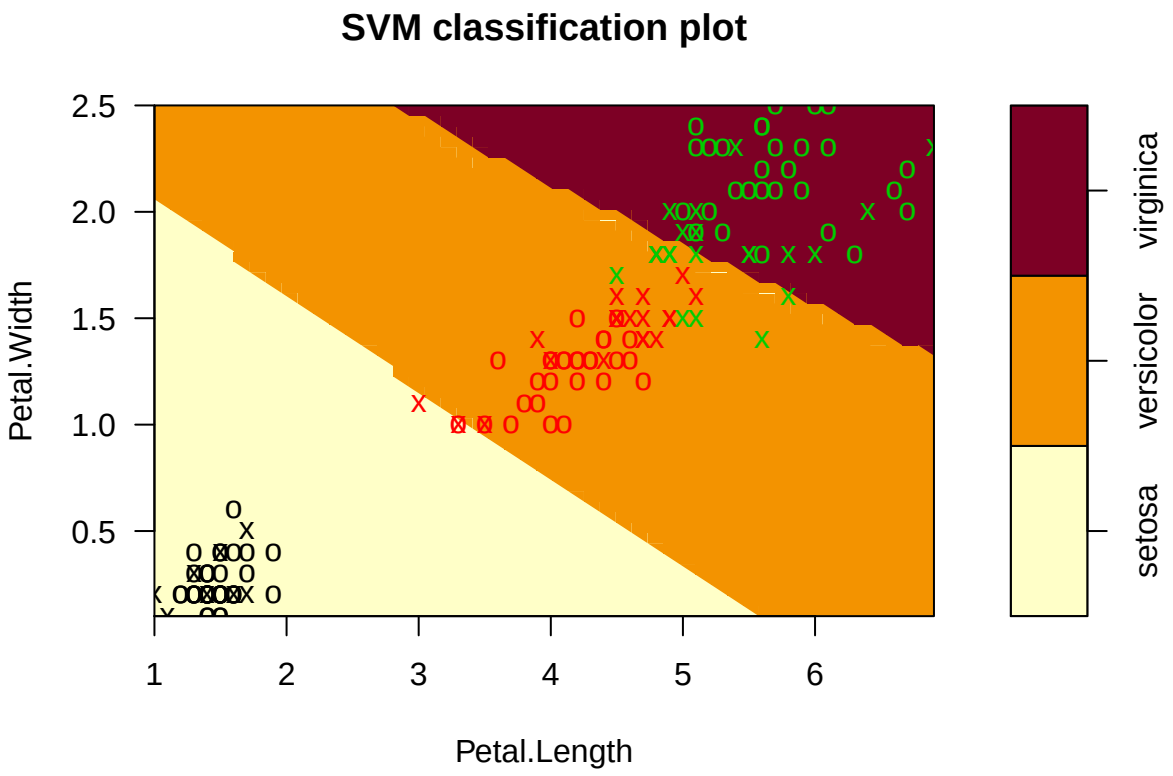
```
SVMr = svm(Species ~., data = iris, kernel = 'radial')
summary(SVMr)
```

```
##
## Call:
## svm(formula = Species ~ ., data = iris, kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  51
##
##  ( 8 22 21 )
##
##
## Number of Classes:  3
##
## Levels:
##  setosa versicolor virginica
```

```
pred = predict(SVMr, iris)
print(table(pred,Truth = iris$Species))
```

```
##             Truth
## pred         setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          2        48
```

```
plot(SVMr,data = iris,Petal.Width~Petal.Length,slice = list(Sepal.Width=3, Sepal.Length=4))
```



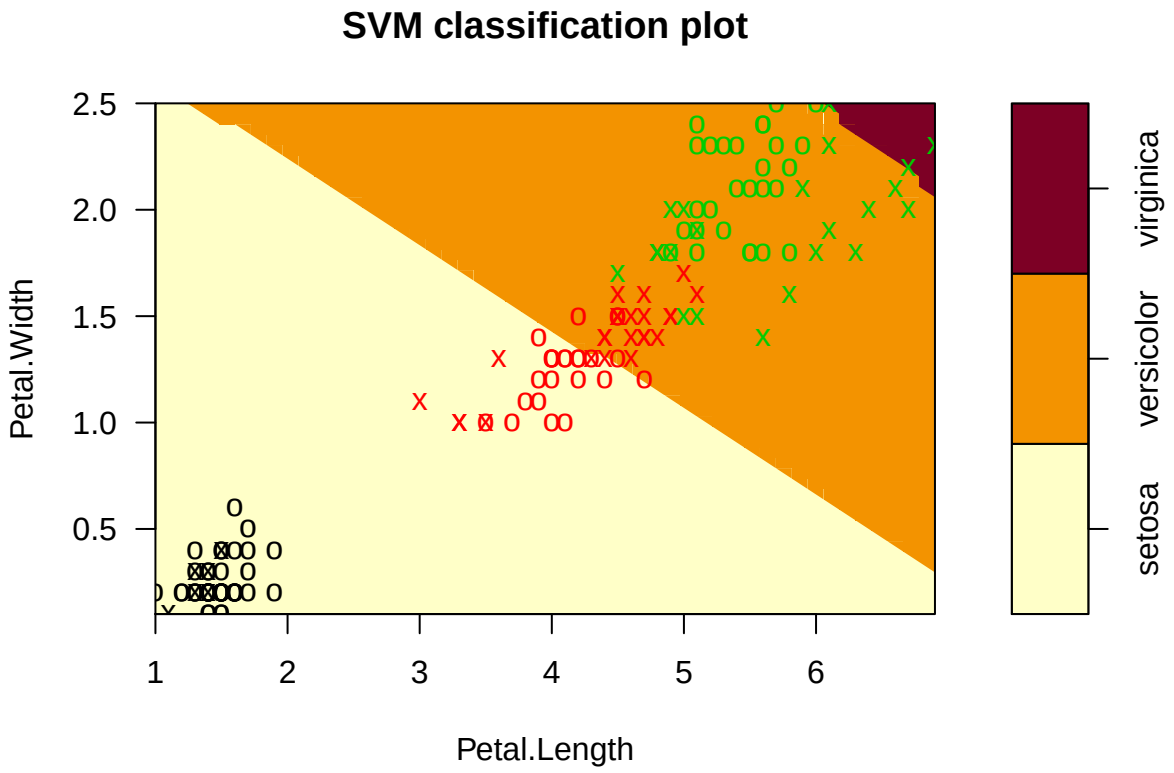

```
SVMs = svm(Species ~., data = iris, kernel = 'sigmoid')
summary(SVMs)
```

```
##
## Call:
## svm(formula = Species ~ ., data = iris, kernel = "sigmoid")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  sigmoid
##        cost:  1
##      coef.0:  0
##
## Number of Support Vectors:  54
##
##  ( 6 26 22 )
##
##
## Number of Classes:  3
##
## Levels:
##  setosa versicolor virginica
```

```
pred = predict(SVMs, iris)
print(table(pred,Truth = iris$Species))
```

```
##              Truth
## pred          setosa versicolor virginica
##   setosa          49          0         0
##   versicolor       1         41         7
##   virginica        0          9        43
```

```
plot(SVMs,data = iris,Petal.Width~Petal.Length,slice = list(Sepal.Width=3, Sepal.Length=4))
```




```
SVMp = svm(Species ~., data = iris, kernel = 'polynomial')
summary(SVMp)
```

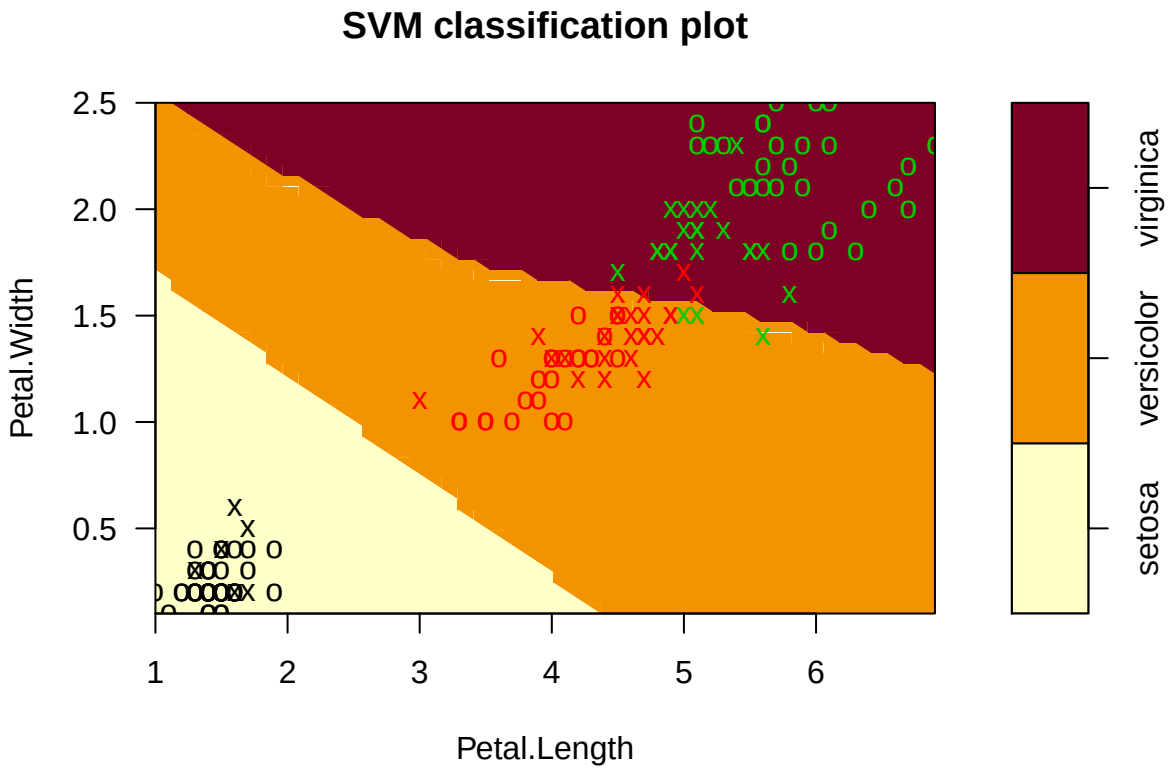
```
##
## Call:
## svm(formula = Species ~ ., data = iris, kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  54
##
##  ( 6 26 22 )
##
```

```
##
## Number of Classes:  3
##
## Levels:
##  setosa versicolor virginica
```

```
pred = predict(SVMp, iris)
print(table(pred,Truth = iris$Species))
```

```
##             Truth
## pred         setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         50         7
##   virginica       0          0        43
```

```
plot(SVMp,data = iris,Petal.Width~Petal.Length,slice = list(Sepal.Width=3, Sepal.Length=4))
```



Die klassifikation mit dem radiale Kernel hat die geringsten falsch klassifizierten Ereignisse (4) dafür aber 51 Stützvektoren.
Der lineare Kernel hat nur 29 Stützvektoren und nur ein falsch klassiviziertes ereigniss mehr.