Arbeitsblatt 6

Aufgabe 1: Orangensaft

Wir wollen vorhersagen, welches Produkt ein Kunde kauft, basierend auf einigen Eigenschaften des Kunden und des Produkts. Dazu verwenden wir einen Datensatz, der 1070 Käufe enthält, bei denen der Kunde entweder Citrus Hill (CH) oder Minute Maid (MM) Orangensaft gekauft hat. Die Klassifizierungsaufgabe besteht darin, vorherzusagen, ob ein Kunde Orangensaft der Marke CH oder MM kauft (Zielvariable `Purchase`). Der Datensatz OJ ist im ISLR-Paket enthalten oder auf Moodle zu finden. (Die Aufgabe stammt aus dem Buch ISLR.)

a) Verwenden Sie die OJ-Daten und erstellen Sie ein Trainingsset mit einer Zufallsstichprobe von 800 Beobachtungen und ein Testset mit den restlichen Beobachtungen.

b) Passen Sie einen Support-Vektor-Klassifikator an die Trainingsdaten an, indem Sie in der `train`-Funktion `method='svmLinear2'` und Kosten `cost = 0.01` verwenden. Zielvariable `Purchase` mit allen anderen Variablen als Prädiktoren. Verwenden Sie `..$finalModel`, um das endgültige Modell zu sehen.

c) Was ist die Genauigkeit von Training- und Testdaten?

d) Optimieren Sie den Cost-Parameter. Berücksichtigen Sie für das Argument `tuneGrid` Werte im Bereich von 0.01 bis 10.

e) Was ist die Genauigkeit von Training- und Testdaten mit dem optimierten Cost-Parameter?

f) Zeichnen Sie die ROC-Kurve für den optimierten Klassifizierer auf den Testdaten. Damit sie in der `predict` den `typ='prob'` verwenden können, müssen Sie hier in `trainControl` `classProbs = TRUE` setzen, da SVM standardmässig keine Wahrscheinlichkeitsvorhersagen liefert.

Exercise 2: Nicht-lineare Kernel

Eine SVM kann auch mit einem nichtlinearen Kernel ausgestattet werden, um eine Klassifizierung anhand einer nichtlinearen Entscheidungsgrenze durchzuführen. (Idee aus ISLR.)

a) Erzeugen Sie einen Datensatz mit n = 500 und p = 2, so dass die Beobachtungen zu zwei Klassen mit einer quadratischen Entscheidungsgrenze gehören. Sie können dies zum Beispiel wie folgt tun:

```
set.seed(4)
x1 <- runif(500) - 0.5
x2 <- runif(500) - 0.5
y <- as.factor(1 * (x1^2 - x2^2 > 0))
```

b) Zeichnen Sie die Beobachtungen in einen Scatterplot und färben Sie die Beobachtungen entsprechend ihrer Klassenbezeichnungen ein.

c) Passen Sie einen linearen Support-Vektor-Klassifikator an die Daten an (`method='svmLinear2'` und `c =0.1`). Berechnen Sie mit `predict` die Klassenvorhersage für die Trainingsdaten. Zeichnen Sie die Beobachtungen mit den vorhergesagten Klassen (Farbe) und den tatsächlichen Klasse (Symbol z.B. `pch` in `plot` oder `shape` in `ggplot`) auf.

d) Passen Sie eine SVM an die Daten an, indem Sie einen radialen Kernel (`method='svmRadial'`) verwenden. Passt das Modell besser? Wenn Sie Lust haben, können Sie auch noch einen

polynomialen Kernel ausprobieren (`method='svmPoly'`), Sie können sich dabei auf Polynome zweiten Grads beschränken (`degree=2`).