

Introduction to Machine Learning - Lab 5

Gustav Sternelöv

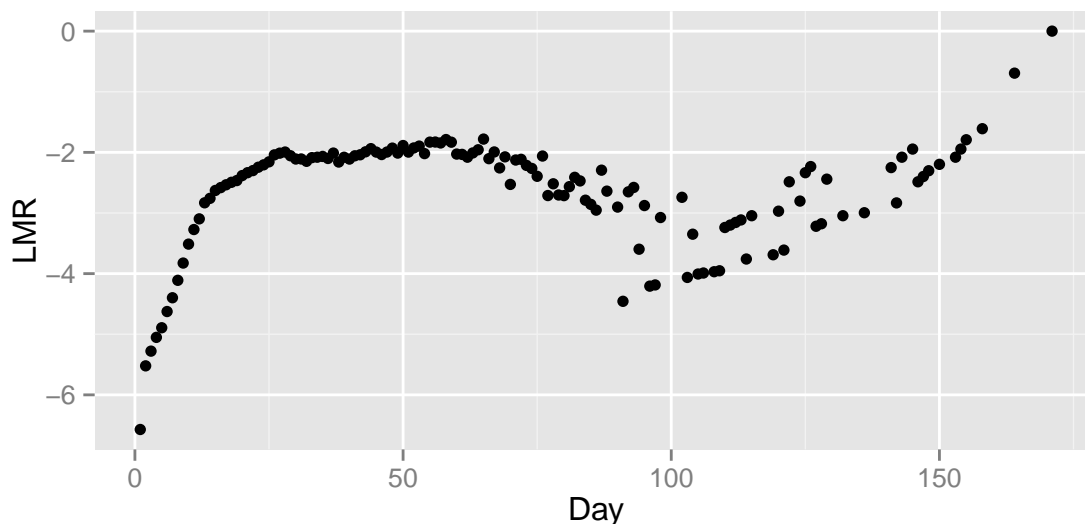
Thursday, November 19, 2015

Assignment 1

The studied data set contains information about the mortality rate for fruit flies for each day. The data comes from a study where the theory that the mortality rates (probability of dying per unit time) of many organisms increase at an exponential rate was tested.

1.1

The variable LMR, that is the logarithm of the variable Rate, is created and plotted against the variable Day.

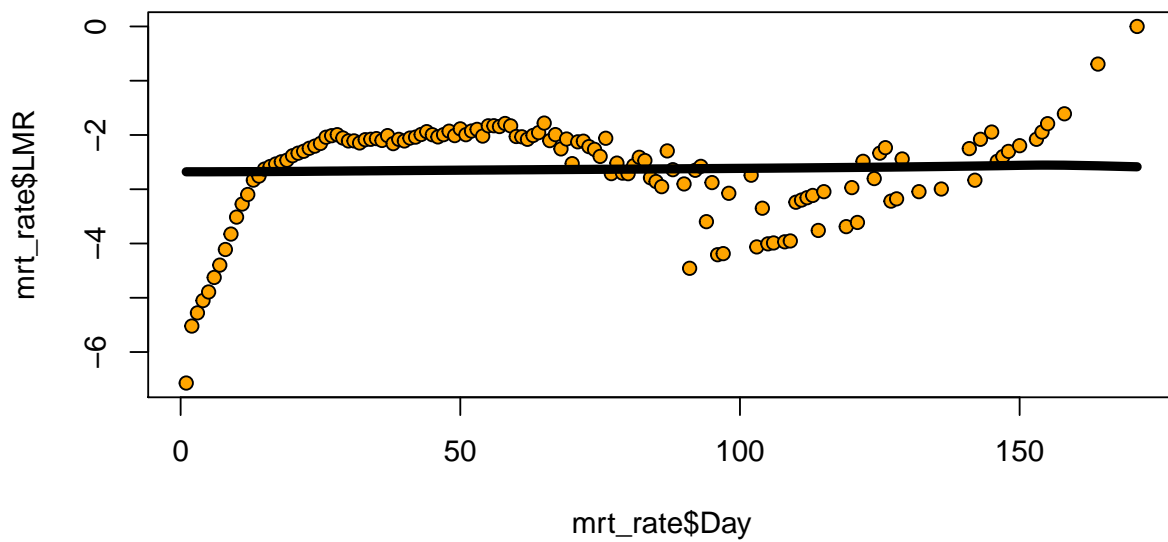


1.2 - 1.3

A function that performs Nadarya-Watson kernel smoothing with Epanecnikov kernel is implemented and tested in this step. The function depends on an x and y vector, a vector Xtest that contains the grid and the value λ that represents the bandwidth.

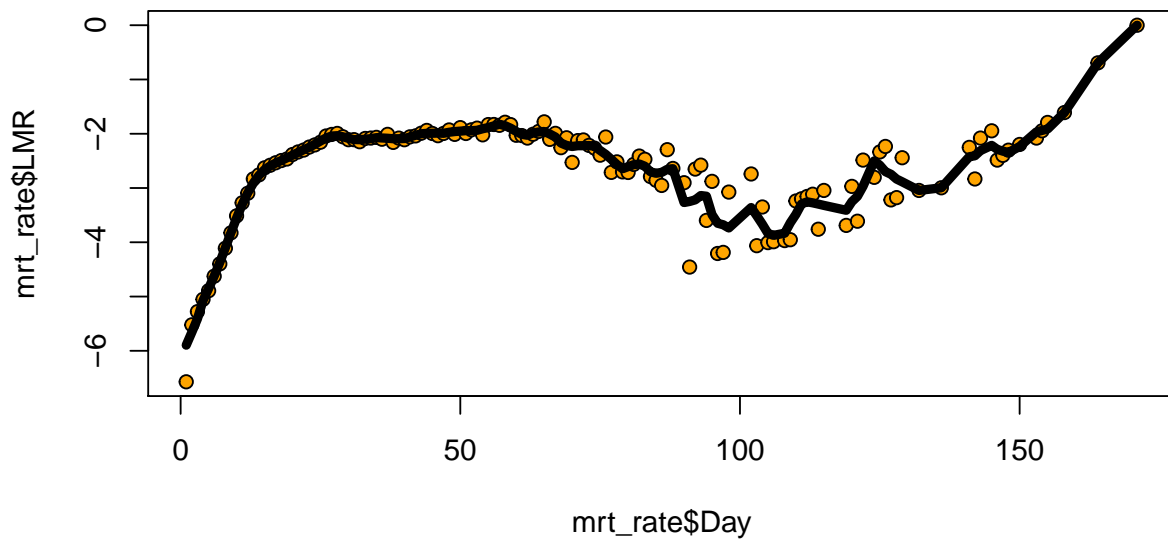
a)

The aim here is to get a very smooth curve and this is achieved by setting λ to 150.



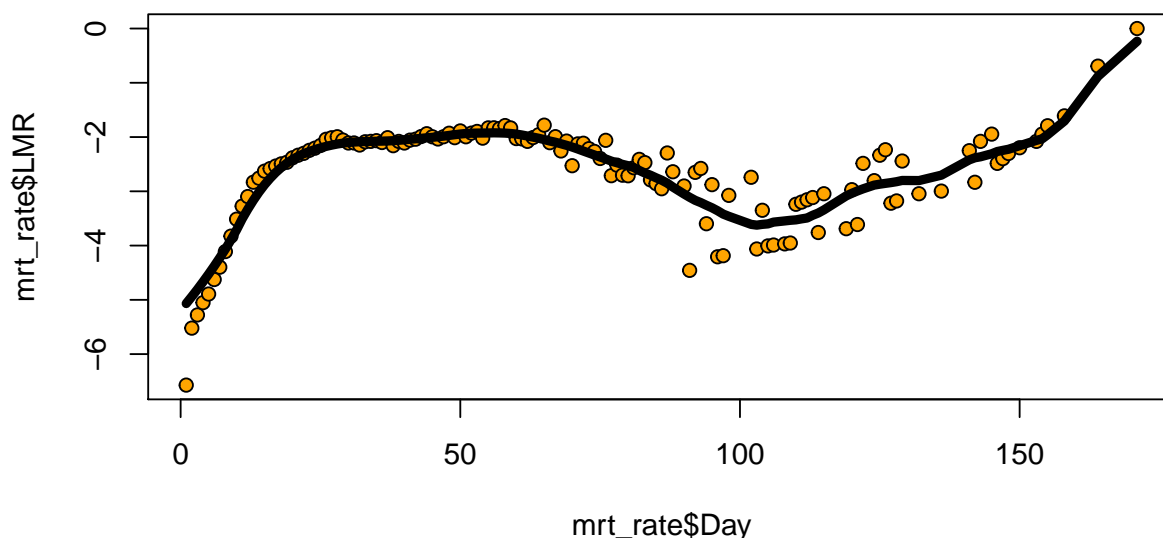
b)

The opposite to a very smooth curve is a very wiggly curve. An example of this type of predicted curve is given for a λ value equal to 3.



c)

In the last example the aim is to get a predicted curve that seems to be a reasonably good fit. This is achieved by choosing a λ value that neither overfit nor underfit data. A value for λ equal to 10 then seem to be an appropriate choice.

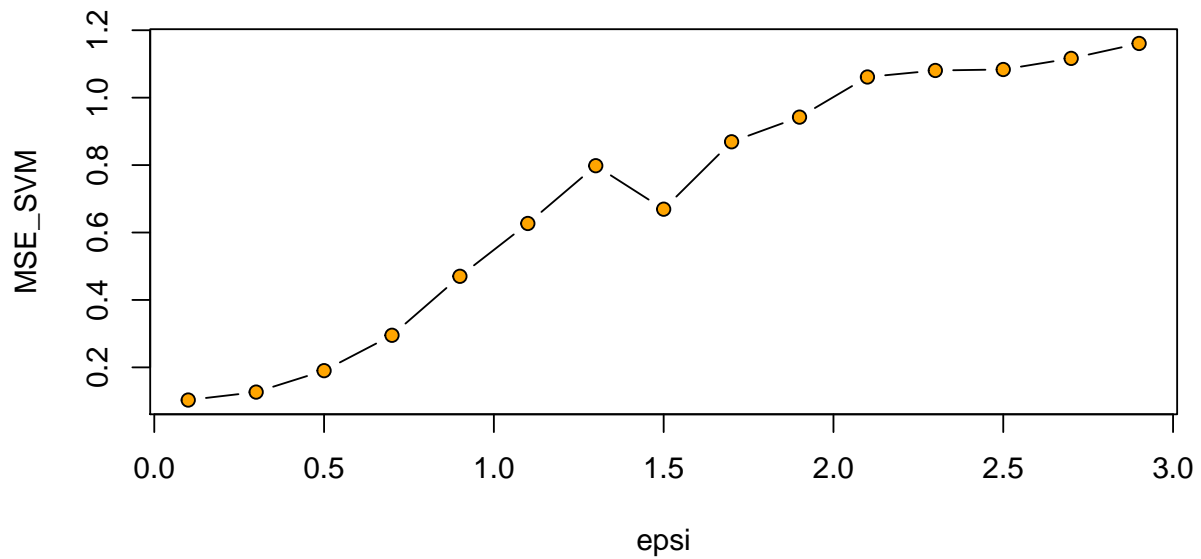


Comparison of the predicted curves

The connection between the bandwidth parameter λ and the form of the predicted curves is straightforward. High values of λ results in more smooth curves and low values results in curves with a more wiggly form. The best model is thought to be the model presented in 3.c and the estimated MSE value for this model is 0.1051538.

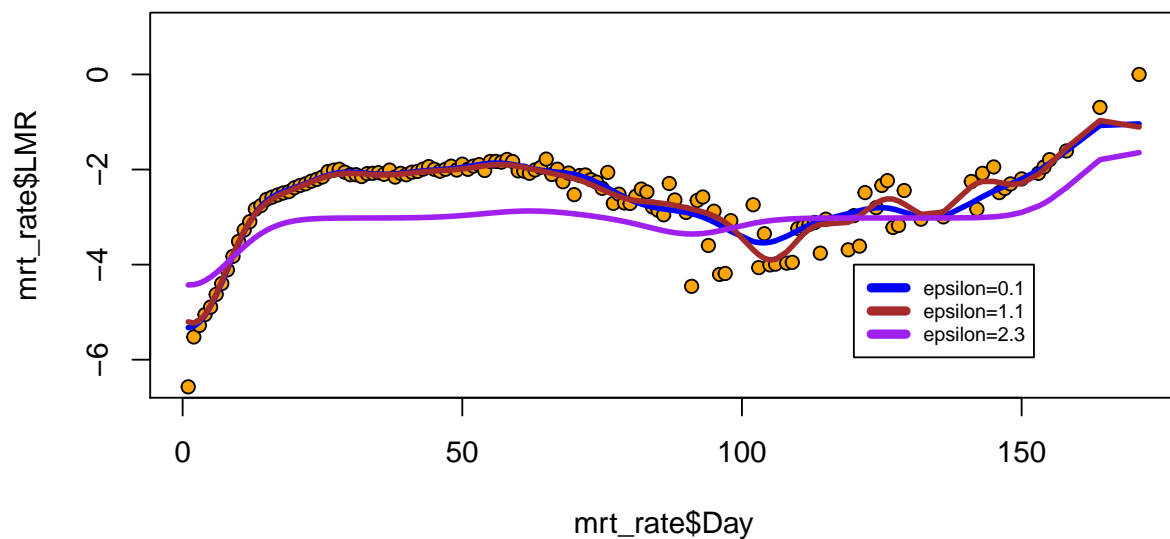
1.4

Another way to fit the logarithm of rate as a function of the variable day is by using SVM for regression. An RBF kernel is used and the task is to find a value for the parameter *epsilon*, ϵ , that results in a good fit. The MSE for values of ϵ from 0.1 to 3 by steps of 0.2 is calculated to investigate what value of ϵ that can be reasonable to select.



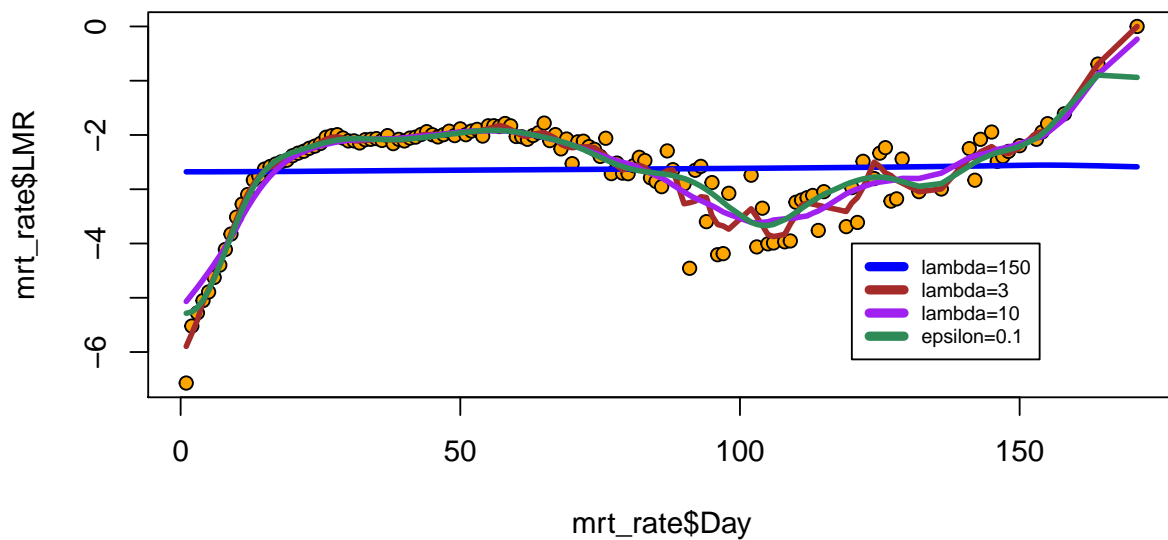
The lowest MSE value is obtained for an ϵ value equal to 0.1, so a reasonable fit is assumed to be given by that model.

To investigate how the fit changes when ϵ increases the predicted curves for a set of different values of ϵ are compared.



For higher and higher values for ϵ the predicted curves becomes more and more smooth. This is because ϵ functions a bit like a smoothness parameter, like λ for the function implemented in 1.3.

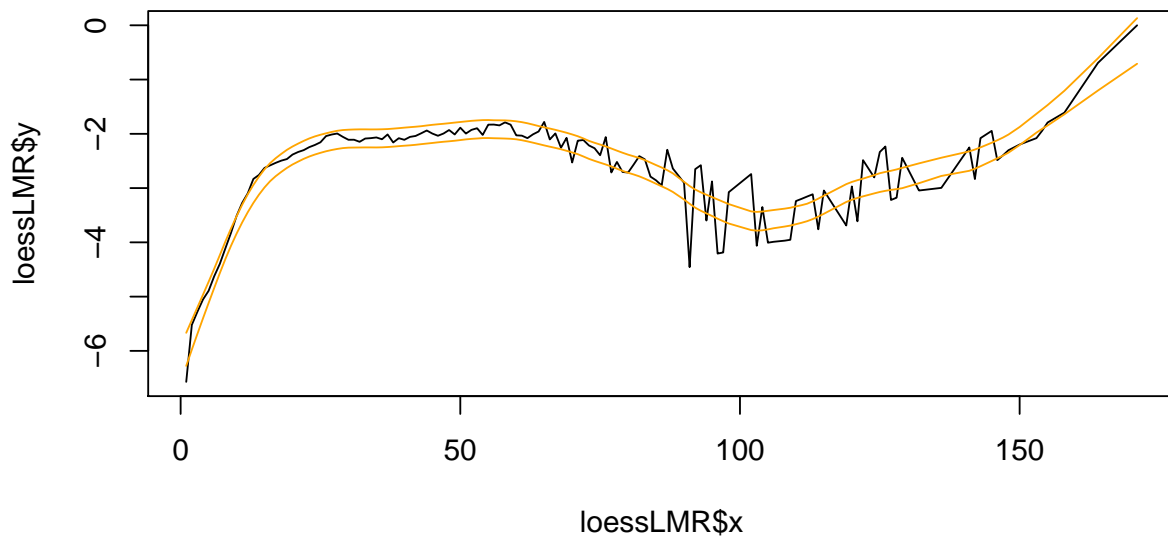
A comparsion between the predicted curves from the models in 1.3 and 1.4 is given with the following graph.



The predicted curves for the the best fit in 1.3 and the SVM model is rather similar. The MSE for the SVM model is 0.1030965 and for the best model in 1.3 the MSE was 0.1051538. The SVM model is concluded to be a slightly better fit since it has a lower MSE.

1.5

A third method used to model the logarithm of Rate against Day is *loess*. A plot of the fitted values and 95 % confidence bands can be seen below.



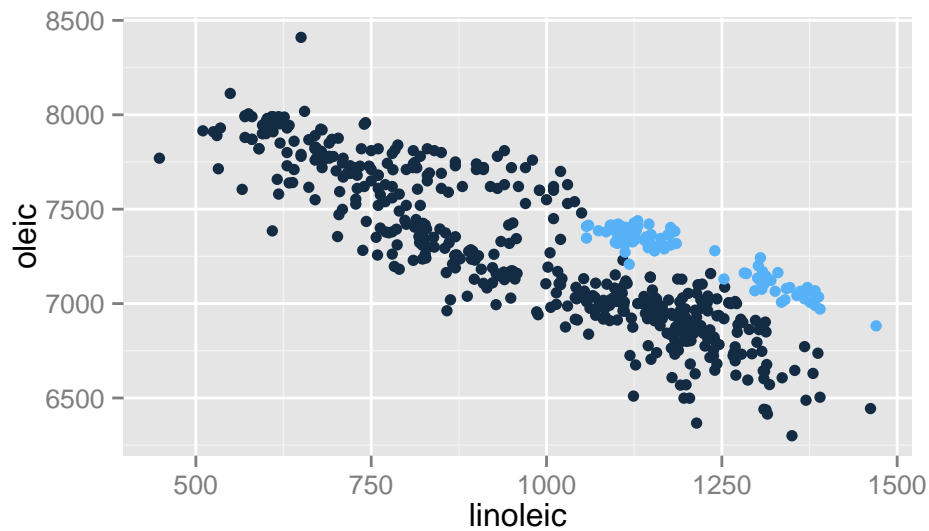
By looking at the plot it is not suggested that Rate is an exponential function of Day. In order for the Gompertz hypothesis to be satisfied the fitted values should be increasing for all values for the variable Day.

Assignment 2

The data set analysed in this assignment consists of information about 572 Italian olive oils coming from different regions of the country. How much of different acids each olive oil contains and from which region and area the olive oil comes from is the information given.

2.1

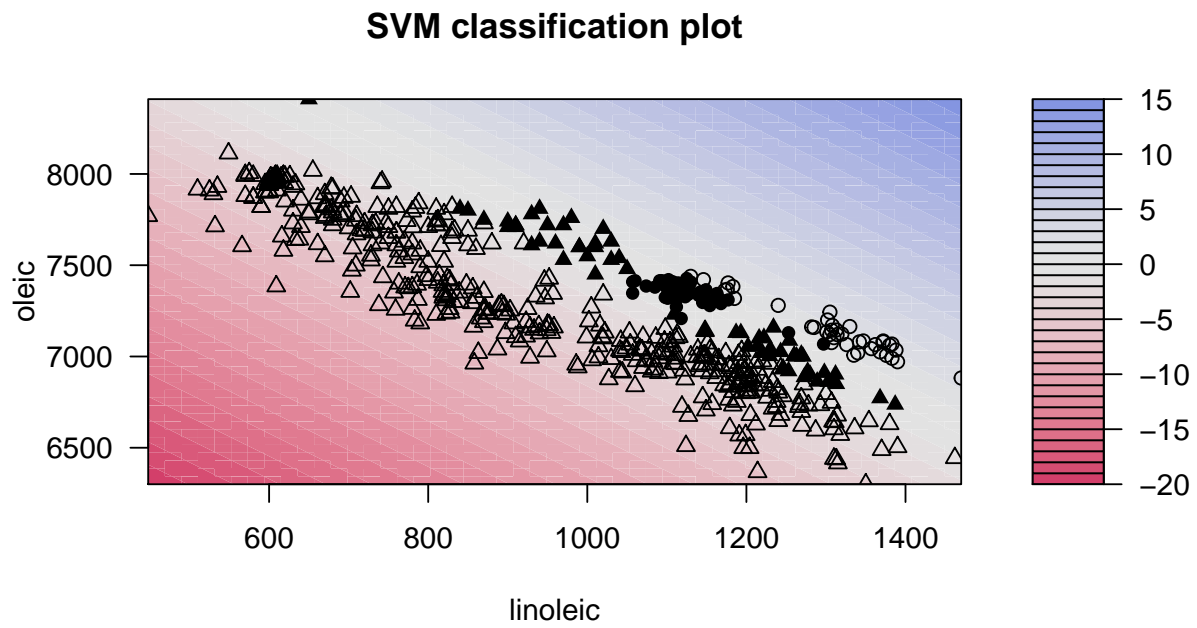
Two of the acids in the data set are *Oleic* and *Linoleic*. In the following graph these acids are plotted against each other and coloured after region where oils from region two are light blue and the others are dark blue.



The oils from region two are quite easy to identify since they lie rather separately from oils from the other regions. At least that is true for the majority of the observations from region two. Some of the dark blue points lie very close to the outer edges of the group of light blue points. For these observations it may be hard for a model to correctly classify an olive oil as coming from region two or from one of the other regions.

2.2

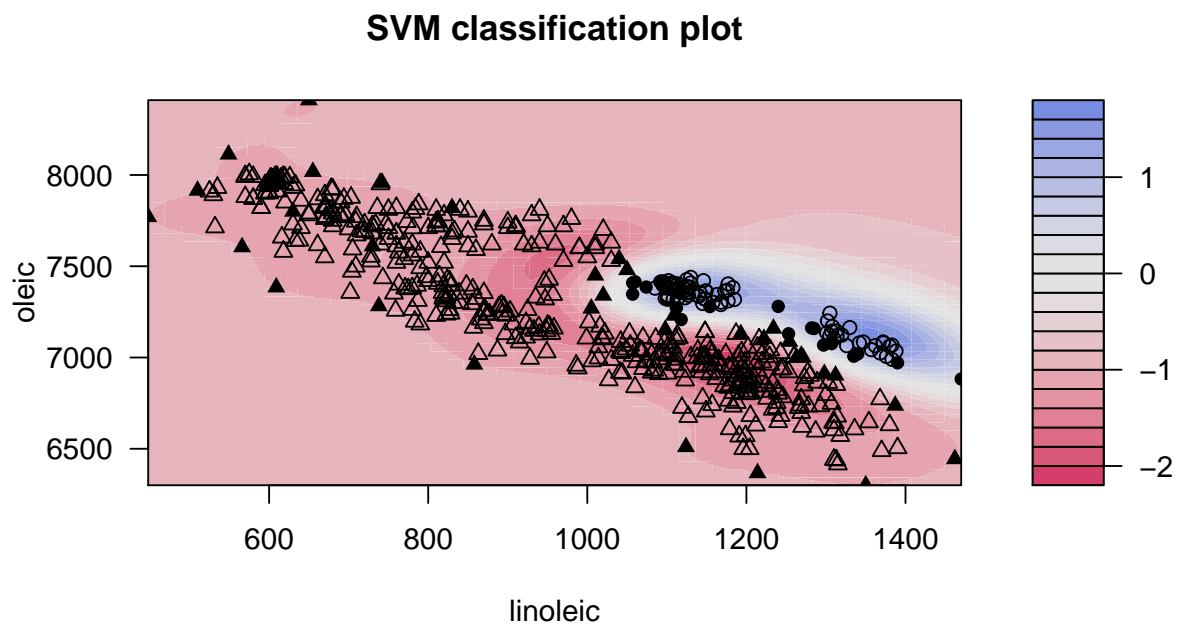
a)



The misclassification rate: 0.0524476

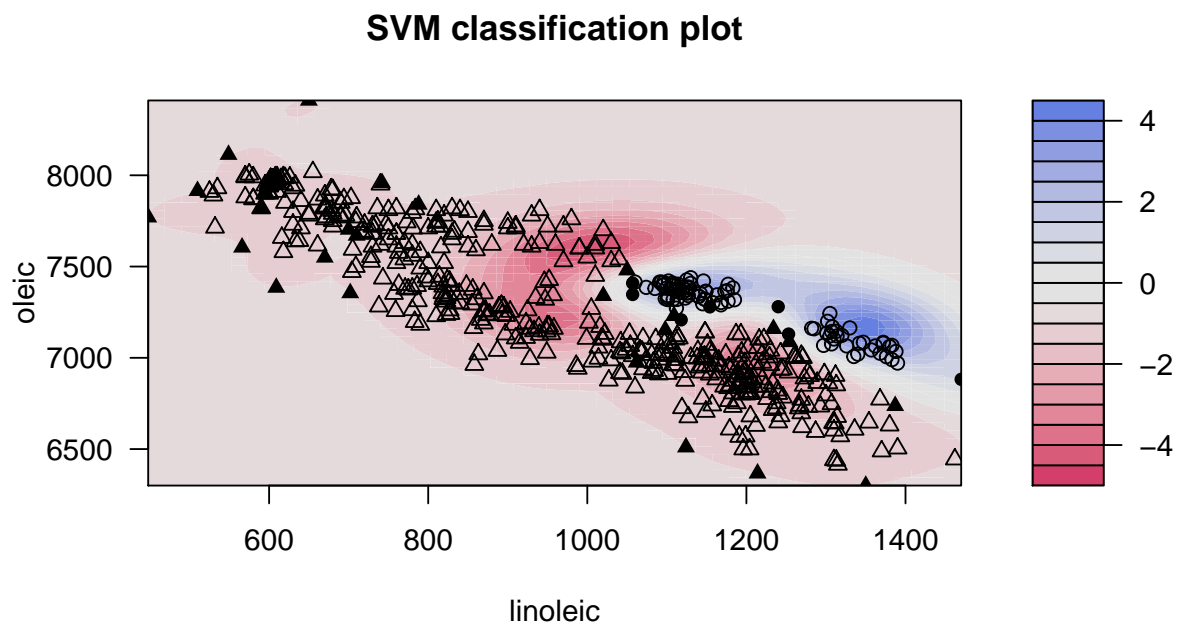
The amount of support vectors: 119

b)



The misclassification rate: 0.0052448
The amount of support vectors: 52

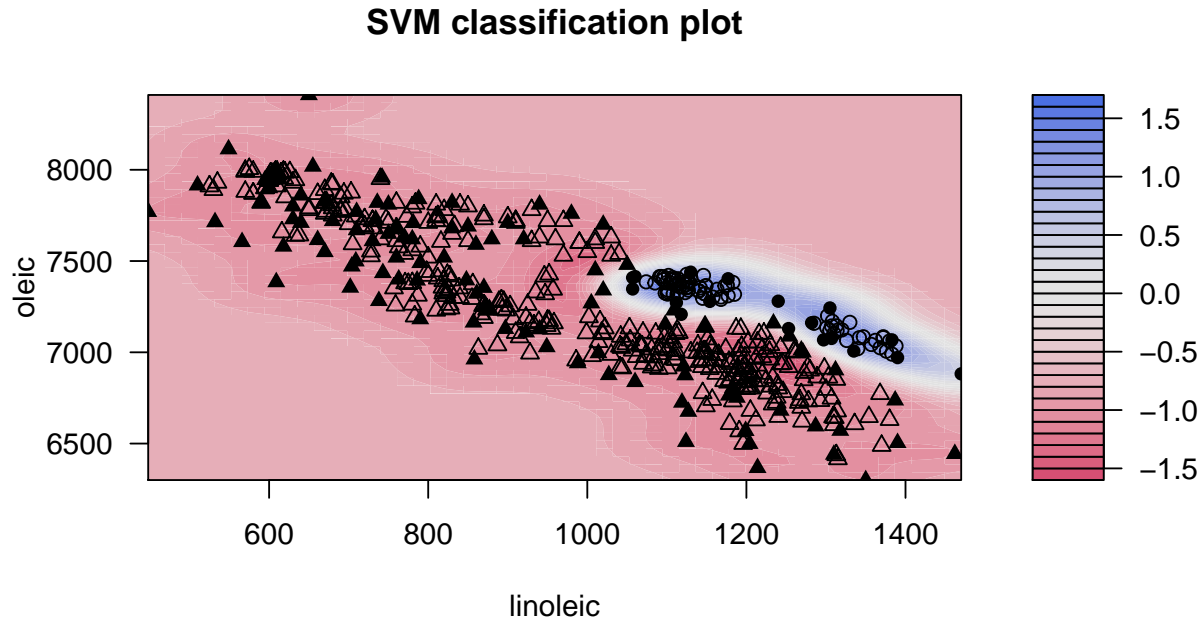
c)



The misclassification rate: 0.0052448

The amount of support vectors: 32

d)



The misclassification rate: 0.0052448

The amount of support vectors: 119

Comparison of models

In terms of misclassification it is very hard to pick the best model since the misclassification rates for model *b*, *c* and *d* is exactly the same. Model *a* has a misclassification rate ten times as high and is therefore suggested to be a worse fit than the others.

The parameter chosen in step c, that penalty *C* is equal to 100, defines the cost of constraints violation. The effect of this is that fewer support vectors are used to make classifications.

In step d the value of sigma has the concrete effect that the amount of support vectors increases. That is because the used sigma value results in a larger margin width and all observations that lies inside the margin width becomes support vectors.

The amount of support vectors in the models differs significantly. In model *a* and in model *d* 119 of the 572 observed values are used as support vectors. Less than a half of this amount of support vectors are used in model *b*, 52, and in model *c* 32 values are used as support vectors.

The best model is suggested to be model *c* because it has the lowest amount of support vectors. It uses the lowest number of observed values as support vectors and predicted equally well as model *b* and *d* that has a much larger amount of support vectors.

2.3

Next, a SVM classification model with region as response and all acids in data as predictors is created. The appropriateness for the model is evaluated with help of the following values.

The misclassification rate: 0.0174825
The amount of support vectors: 53
The cross-validation score: 0.0297641

The low misclassification rate implies that SVM is a good classification model for these data. This suggestion is also supported by the relatively small amount of support vectors and the low cross-validation score.

Appendix

R-code

```
mrt_rate <- read.csv("C:/Users/Gustav/Documents/Machine-Learning/Lab 5/mortality_rate.csv", sep=";")

library(ggplot2)
mrt_rate$LMR <- log(mrt_rate$Rate)
ggplot(mrt_rate, aes(y=LMR, x=Day)) + geom_point()
# 1.2
NadWat <- function(X, Y, Xtest, lambda){
  # Wants to go through all x for every Xtest
  # Compute the value for every run and sum the 136 values
  # Do this for every value in xtest
  NdaWat <- 0
  K <- 0
  h <- 0
  for (i in 1:length(Xtest)){
    for (j in 1:length(X)){
      if(abs(X[j]-Xtest[i]) < lambda ){
        K[j] <- 3/4 * (1 - (abs(X[j] - Xtest[i]) / lambda)^2)
      }else{
        K[j] <- 0
      }
    }
    h <- h+1
    NdaWat[h] <- sum(K*Y) / sum(K)
  }
  return(NdaWat)
}

# a) A very smooth curve
tryA <- NadWat(mrt_rate$Day, mrt_rate$LMR, mrt_rate$Day, 150)
plot(mrt_rate$Day, mrt_rate$LMR, pch=21, bg="orange")
points(tryA, x=mrt_rate$Day, type="l", lwd=5)
MSEA <- sum((mrt_rate$LMR - tryA)^2 / 136)

# b) A wiggly curve
tryB <- NadWat(mrt_rate$Day, mrt_rate$LMR, mrt_rate$Day, 3)
plot(mrt_rate$Day, mrt_rate$LMR, pch=21, bg="orange")
points(tryB, x=mrt_rate$Day, type="l", lwd=5)
MSEB <- sum((mrt_rate$LMR - tryB)^2 / 136)

# c) A good fit
tryC <- NadWat(mrt_rate$Day, mrt_rate$LMR, mrt_rate$Day, 10)
plot(mrt_rate$Day, mrt_rate$LMR, pch=21, bg="orange")
points(tryC, x=mrt_rate$Day, type="l", lwd=5)
MSEC <- sum((mrt_rate$LMR - tryC)^2 / 136)
```

```

library(kernlab)
set.seed(12345)
epsi <- seq(0.1, 3, 0.2)
j <- 1
MSE_SVM <- 0
for(i in epsi){
  LMR_SVM <- ksvm(LMR ~ Day, mrt_rate, type="eps-svr",
                  kernel="rbfdot", epsilon=i, scaled=FALSE)
  MSE_SVM[j] <- sum(((LMR_SVM@fitted - mrt_rate$LMR)^2)/ 136)
  j <- j+1
}

plot(epsi, MSE_SVM, type="b", pch=21, bg="orange")

epsi_mod <- seq(0.1, 3, 0.2)
eps_mod <- eps_mod[c(1,6,12)]
eps_SVM <- matrix(ncol=3, nrow=136)
j <- 0
set.seed(12345)
for (i in eps_mod){
  j <- j+1
  eps_SVMR <- ksvm(LMR ~ Day, mrt_rate, type="eps-svr",
                  kernel="rbfdot", epsilon=eps_mod[i], scaled=FALSE)
  eps_SVM[,j] <- eps_SVMR@fitted[,1]
}
plot(mrt_rate$Day, mrt_rate$LMR, pch=21, bg="orange", ylim=c(-6.5, 1))
points(eps_SVM[,1], x=mrt_rate$Day, type="l", lwd=3, col="blue")
points(eps_SVM[,2], x=mrt_rate$Day, type="l", lwd=3, col="brown")
points(eps_SVM[,3], x=mrt_rate$Day, type="l", lwd=3, col="purple")
legend(120,-4,c("epsilon=0.1", "epsilon=1.1", "epsilon=2.3"), lty=c(1,1),
      lwd=c(5,5),col=c("blue", "brown", "purple"), cex=0.65)
# Compares original and fitted values, epsilon=0.1
set.seed(12345)
LMR_SVM <- ksvm(LMR ~ Day, mrt_rate, type="eps-svr",
               kernel="rbfdot", epsilon=0.1, scaled=FALSE)

MSE_SVM <- sum(((LMR_SVM@fitted - mrt_rate$LMR)^2)/ 136)
plot(mrt_rate$Day, mrt_rate$LMR, pch=21, bg="orange")
points(tryA, x=mrt_rate$Day, type="l", lwd=3, col="blue")
points(tryB, x=mrt_rate$Day, type="l", lwd=3, col="brown")
points(tryC, x=mrt_rate$Day, type="l", lwd=3, col="purple")
points(LMR_SVM@fitted[,1], x=mrt_rate$Day, type="l", lwd=3, col="seagreen")
legend(120,-4,c("lambda=150", "lambda=3", "lambda=10", "epsilon=0.1"), lty=c(1,1),
      lwd=c(5,5),col=c("blue", "brown", "purple", "seagreen"), cex=0.65)
library(fANCOVA)
loessLMR <- loess.as(mrt_rate$Day, mrt_rate$LMR, 1, family = "gaussian", plot=FALSE,
                    criterion="gcv")
predLoess <- predict(loessLMR, se=TRUE)
upper <- predLoess$fit + predLoess$se.fit * 2
lower <- predLoess$fit - predLoess$se.fit * 2
plot(loessLMR, type="l")
points(loessLMR$x, upper, type="l", col="orange")

```

```

points(loessLMR$x, lower, type="l" ,col="orange")
olive <- read.csv("C:/Users/Gustav/Documents/Machine-Learning/Lab 5/olive.csv", sep=",")
library(kernlab)
olive$R2 <- 0
for (i in 1:572){
  if(olive$Region[i] == 2){
    olive$R2[i] = 1
  }else{
    olive$R2[i] = 0
  }
}
ggplot(olive, aes(x=linoleic, y=oleic)) + geom_point(aes(col=R2))

set.seed(12345)
linearSVM <- ksvm(R2 ~ oleic+linoleic, olive, type="C-svc", kernel="vanilladot")
plot(linearSVM, data=olive)
set.seed(12345)
rbfSVM <- ksvm(R2 ~ oleic+linoleic, olive, type="C-svc",
               kernel="rbfdot")
plot(rbfSVM, data=olive)
set.seed(12345)
rbf_penSVM <- ksvm(R2 ~ oleic+linoleic, olive, type="C-svc",
                  kernel="rbfdot", C=100)
plot(rbf_penSVM, data=olive)
set.seed(12345)
rbf_bwidthSVM <- ksvm(R2 ~ oleic+linoleic, olive, type="C-svc",
                     kernel="rbfdot", kpar=list(sigma=10))
plot(rbf_bwidthSVM, data=olive)
set.seed(12345)
olive_acid <- olive[, c(2, 4:11)]
rbf_spocSVM <- ksvm(Region ~. , olive_acid, type="spoc-svc",
                   kernel="vanilladot", cross=10)
##

```