Introduction to Machine Learning - Lab 5

Martina, Araya, Andrea, Gustav Sternelöv

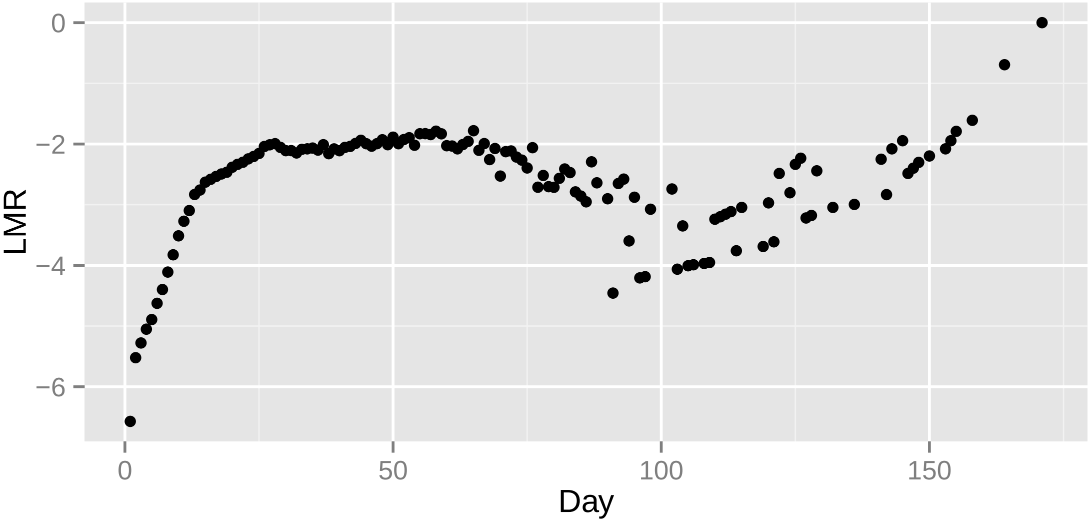
Thursday, November 19, 2015

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

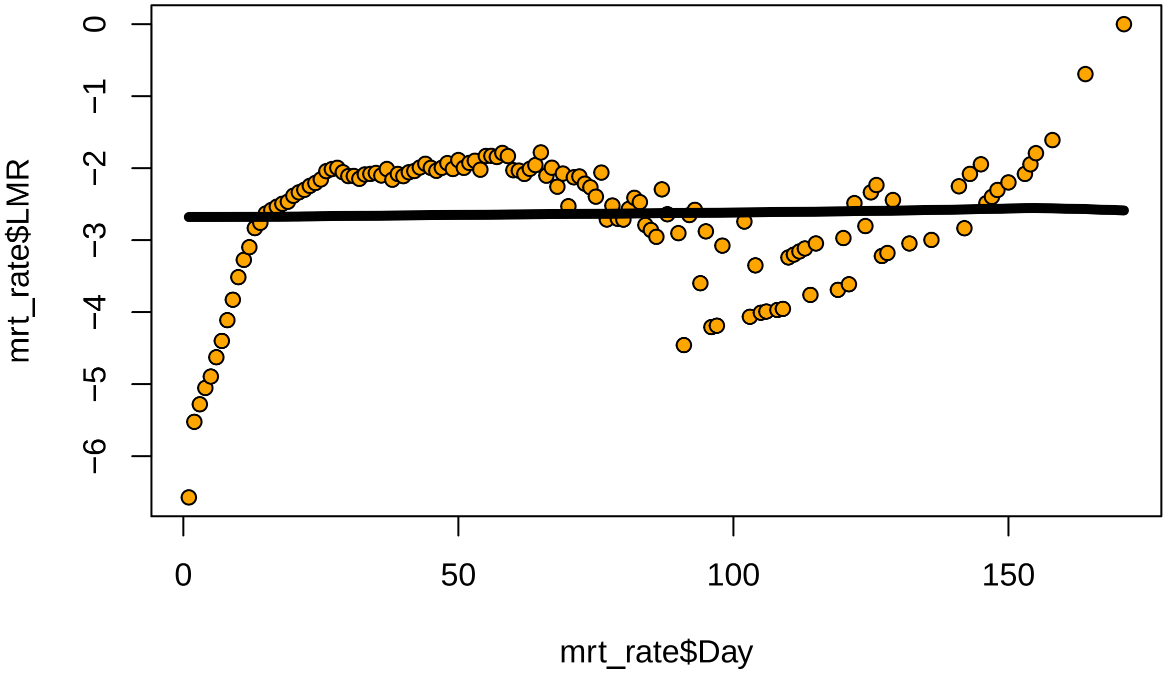
$$K_\lambda(x, x_0) = \frac{3}{4}\left(1 - \left(\frac{|x - x_0|}{\lambda}\right)^2\right) I\left(\frac{|x - x_0|}{\lambda} < 1\right) \tag{1}$$

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_i, x_0) y_i}{\sum_{i=1}^{N} K_\lambda(x_i, x_0)} \tag{2}$$

The Nadaraya-Watson kernel smoothing with Epanechnikov kernel is computed according to (1). Here x is the values in the X vector, x_0 is the values in the Xtest vector and y the values in the Y vector. The predicted target for x_0 is computed as (2). The implemented function can be seen in *Appendix*.

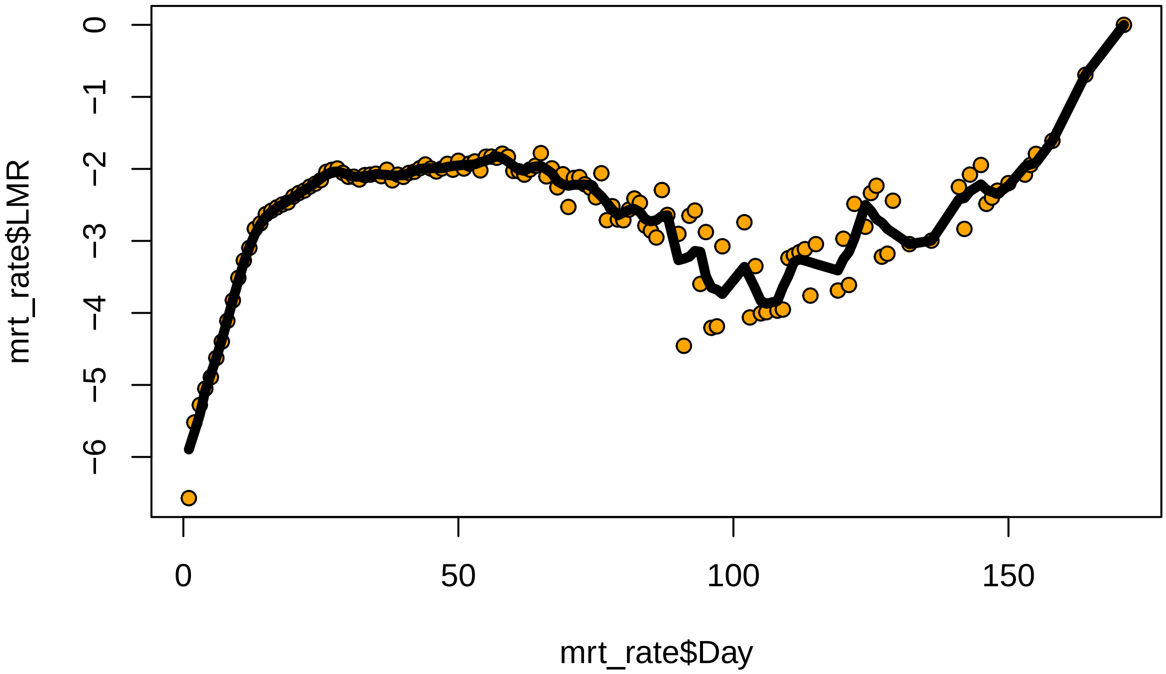
a)

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



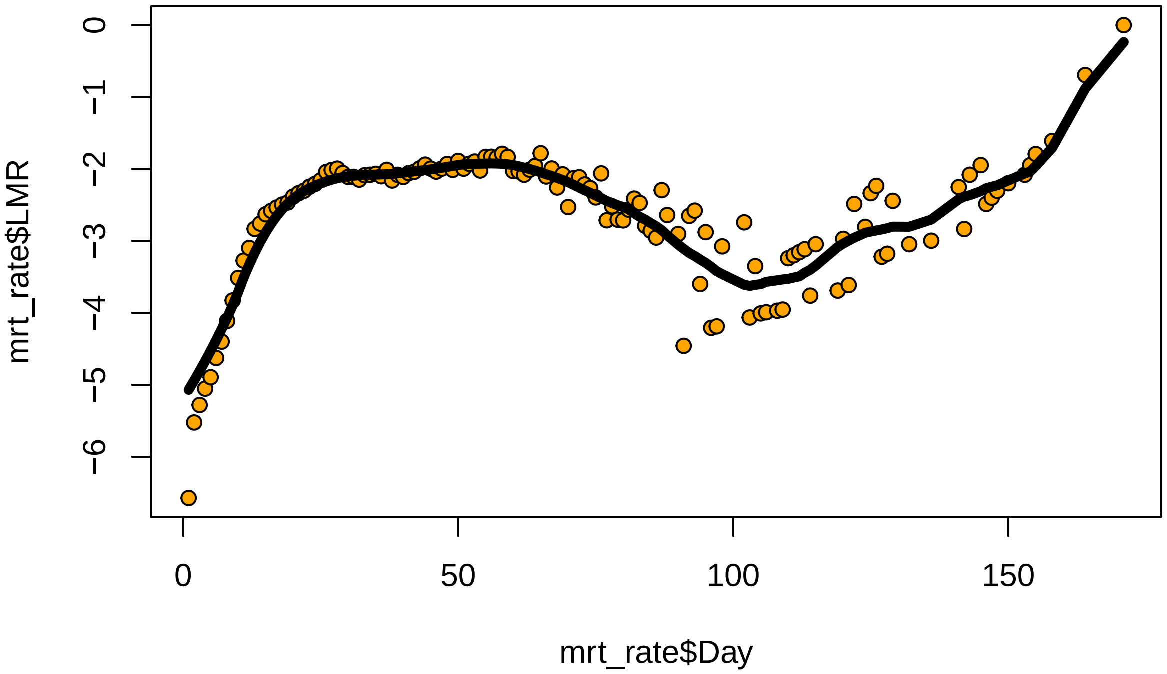
b)

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.

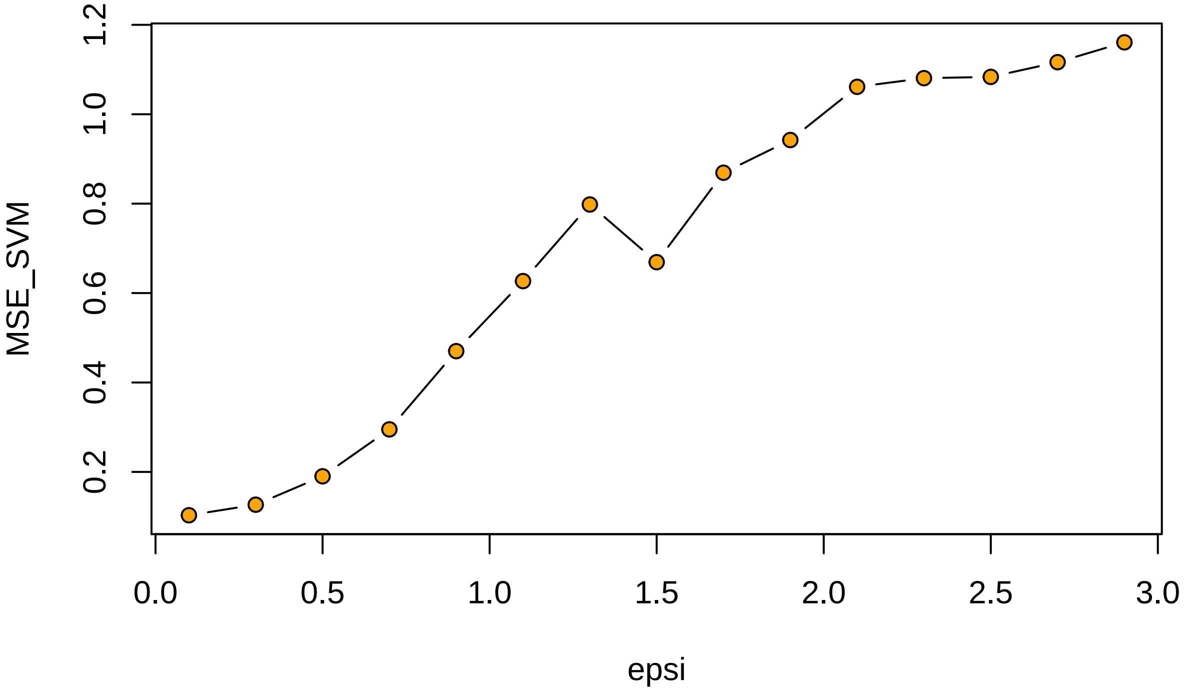


Comparison of the predicted curves

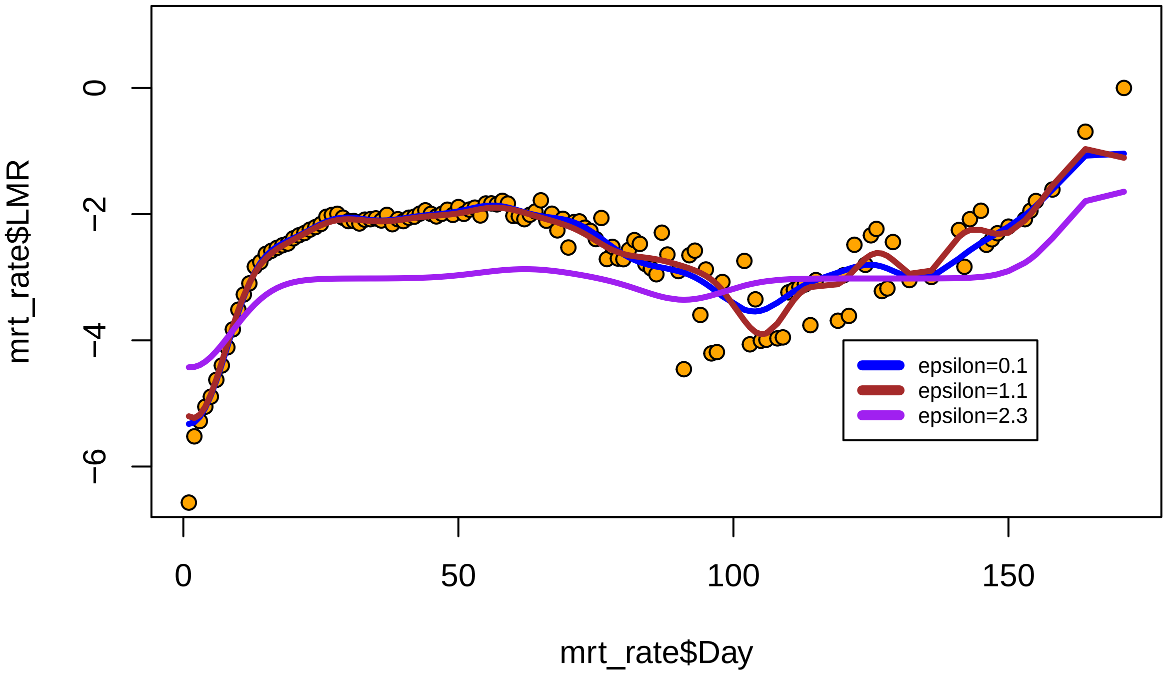
The connection between the bandwidth parameter λ and the form of the predicted curves is straightforward. High values of λ results in more smooth curves and low values results in curves with a more wiggly form. The best model is thought to be the model presented in *3.c* and the estimated MSE value for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



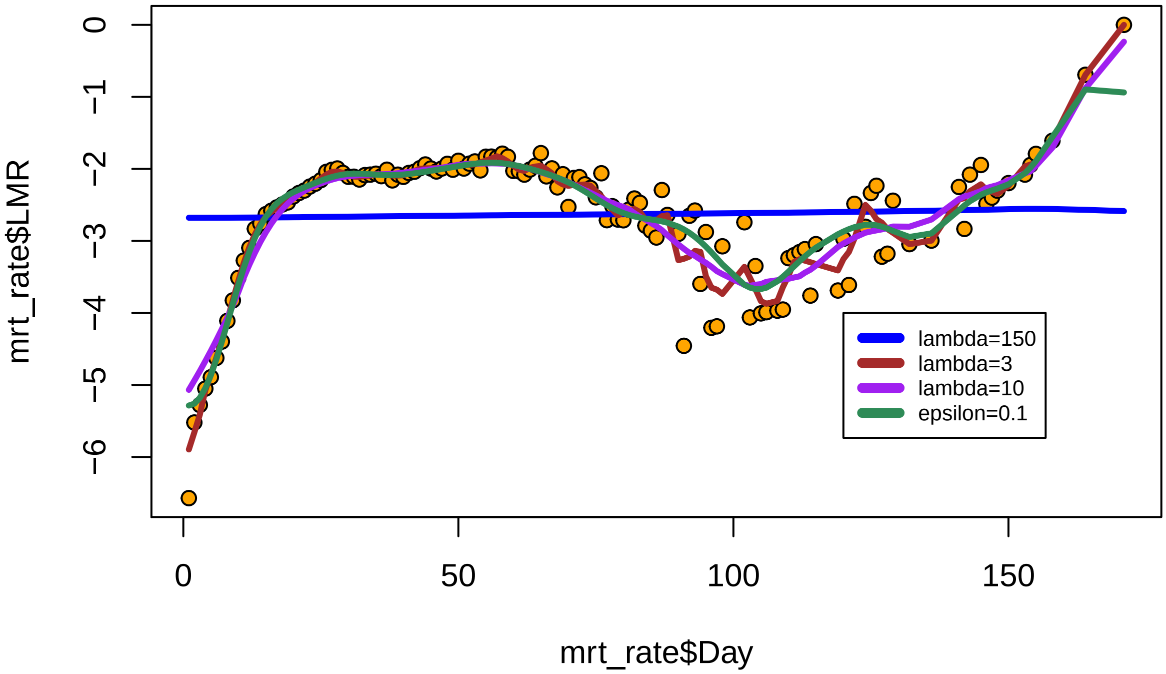
The lowest MSE value is obtained for an ϵ value equal to 0.1, so a reasonable fit is assumed to be given by that model.
To investigate how the fit changes when ϵ increases the predicted curves for a set of different values of ϵ are compared.



The value of ϵ affects the number of support vectors that used to construct the model. The bigger ϵ, the fewer support vectors are selected which also results in more flat estimates. Moreover, an increase in ϵ means a reduction for the model accuracy.

For higher and higher values for ϵ the predicted curves becomes more and more smooth. This is because ϵ functions as 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.

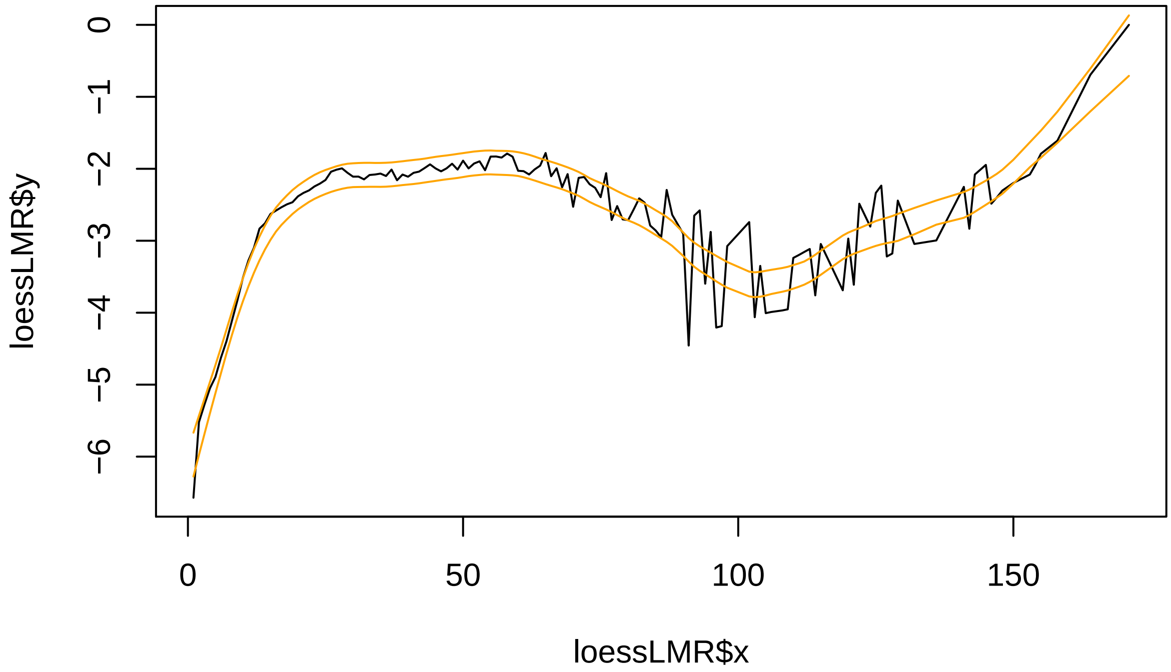


The predicted curve for the SVM model seem to be a bad fit, especially when it is compared to predicted curve for the model from 1.3c)(the model with λ equal to 10). The MSE for the SVM model is 0.1030965 and for the best model in 1.3 the MSE was 0.1051538. In the light of both better looking prediction curves and lower MSE the best model from 1.3 undoubtedly seem to be a better fit than the SVM model.

The predicted curves for the the best fit in 1.3 and the SVM model is rather similar. The latter 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.



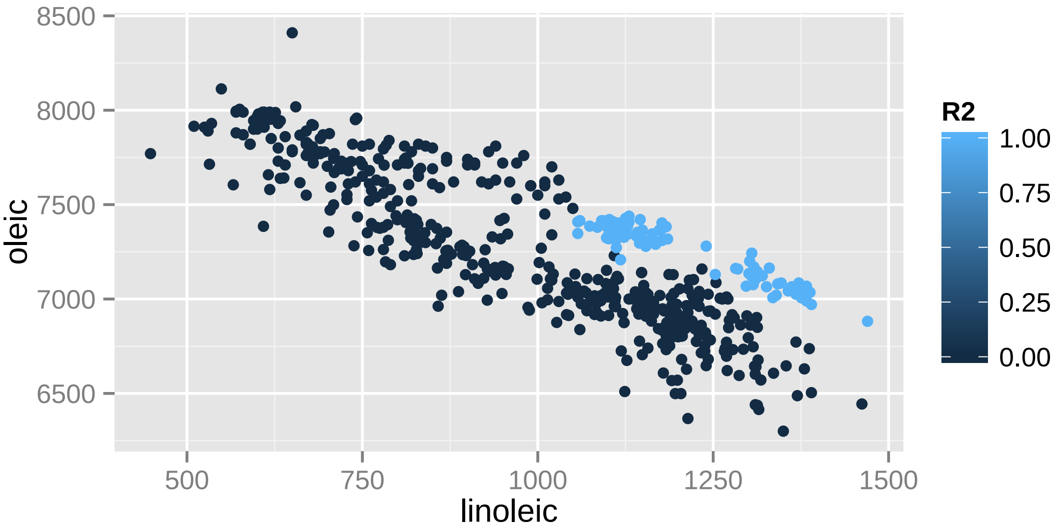
By looking at the plot it is not suggested that Rate is an exponential function of Day. In order for the Gompertz hypothesis to be satisfied the fitted values should be increasing for all values for the variale Day.

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

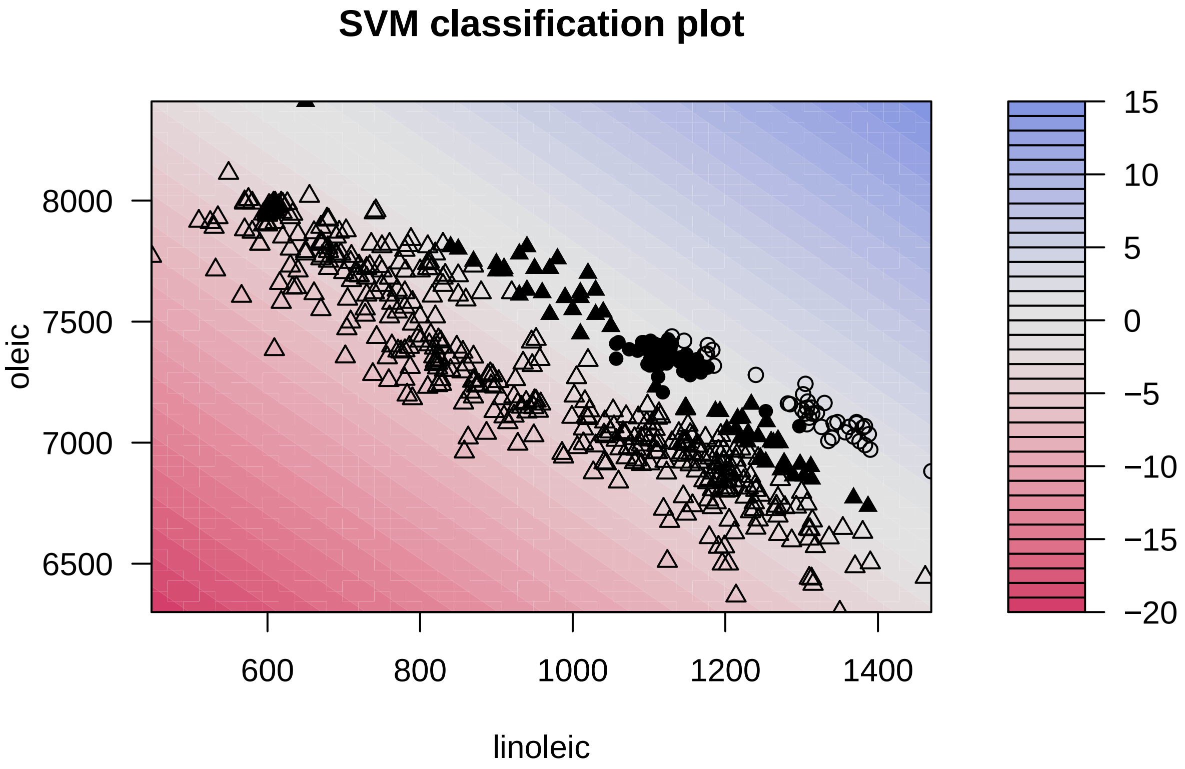
Two of the acids in the data set are *Oleic* and *Linoleic*. In the following graph these acids are plotted against each other and coloured after region where oils from region two are light blue and the others are dark blue.



The oils from region two are quite easy to identify since they lies 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 lies 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.

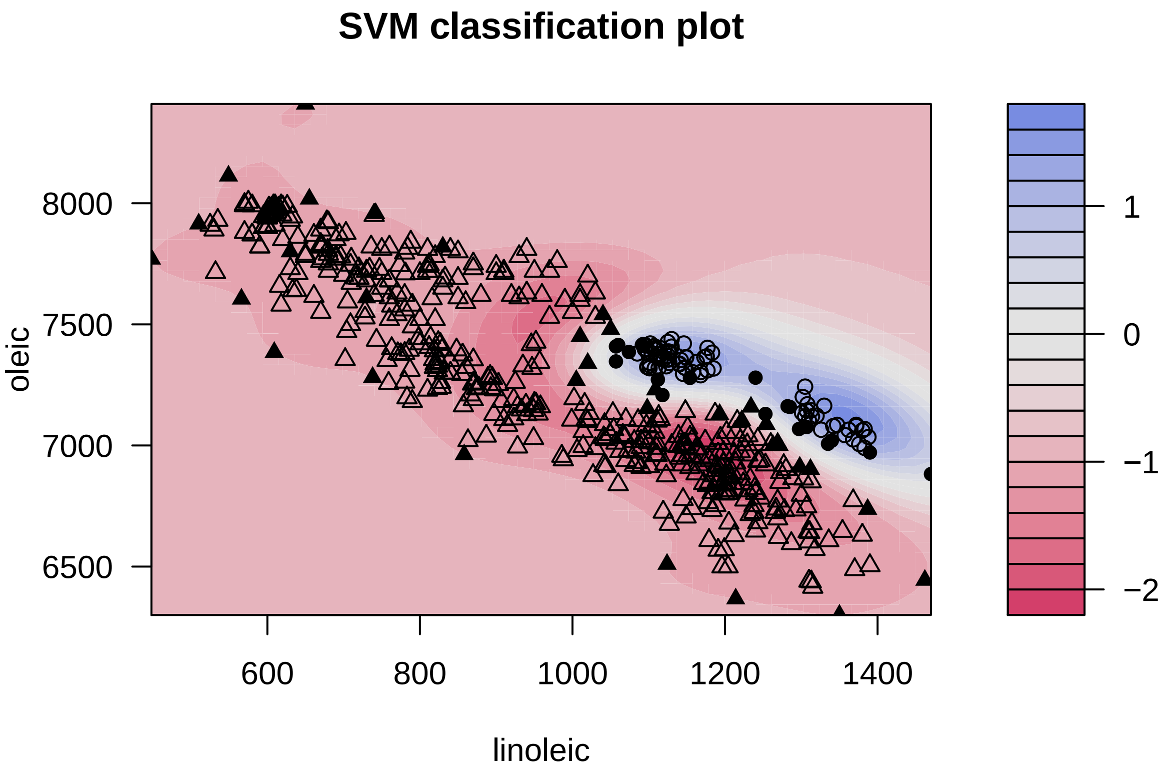
2.2

a)



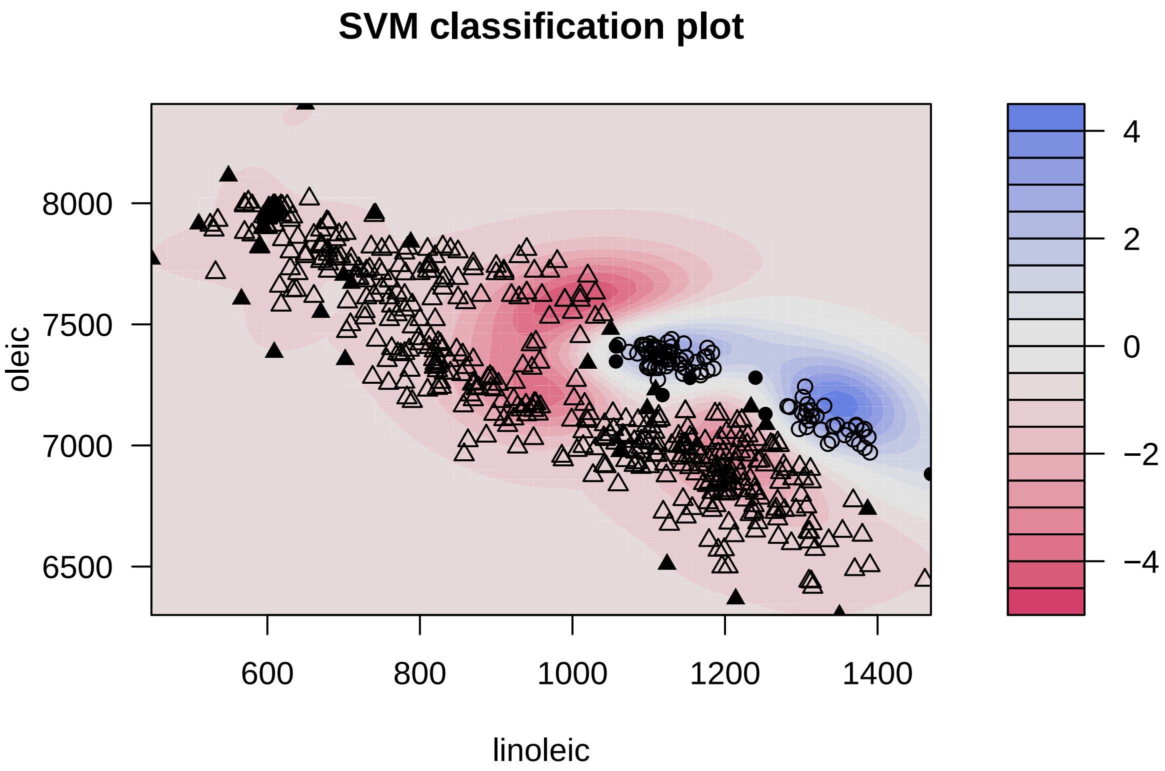
The misclassification rate: 0.0524476
The amount of support vectors: 119

b)



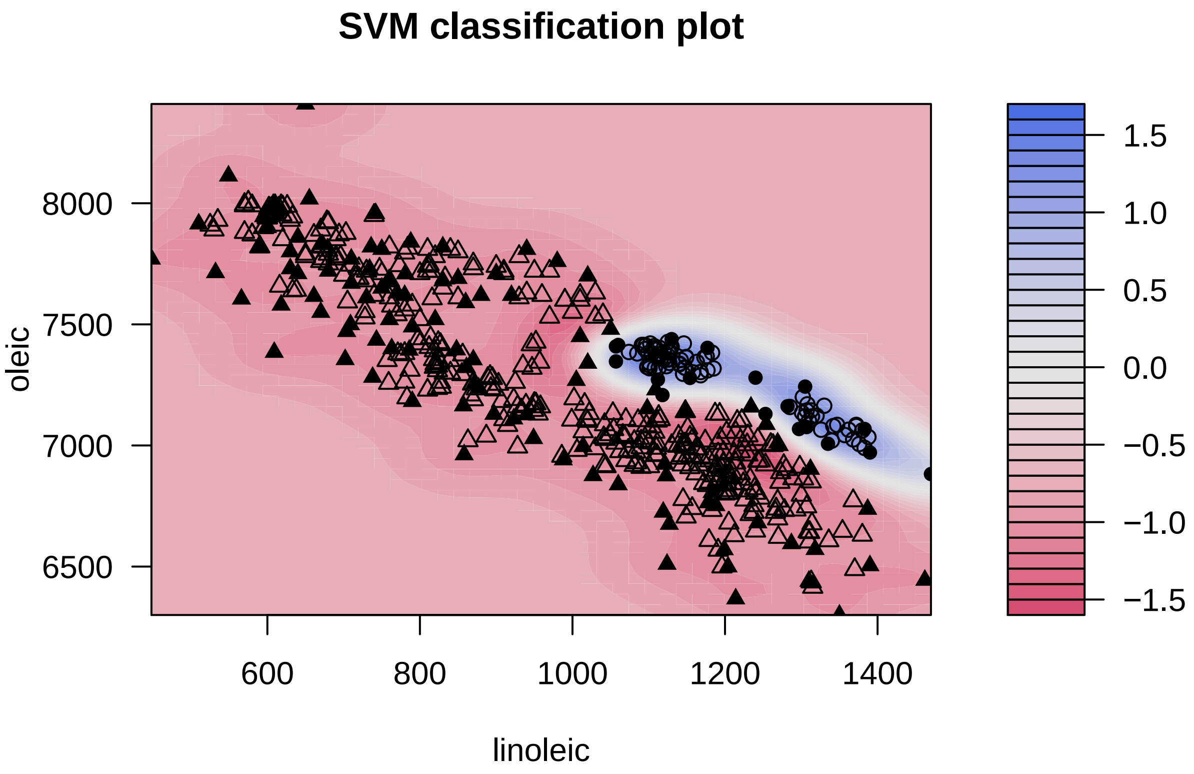
The misclassification rate: 0.0052448
The amount of support vectors: 52

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

```
##                 rate nSV
## Model a 0.052447552 119
## Model b 0.005244755  52
## Model c 0.005244755  32
## Model d 0.005244755 119
```

Four different types of SVM models are fitted with R2 as response and Oleic and Linoleic as predictors. In the first figure we can see the SVM misclassification plot for model a which has a linear kernel, misclassification rate 0.0524 and 119 support vectors. The white area corresponds to the classification boundary and in this case it does not look like it fits well. The blue area corresponds to high probability that the observation is in region 2 and red to low probability. The filled black symbols corresponds to the support vectors. We can conclude that this model does not fit well. In the second figure we can see the SVM misclassification plot for model b which has a RBF kernel, misclassification rate 0.0052 and 52 support vectors. Here the classification boundary looks much better than for model a. In the third figure we can see the SVM misclassification plot for model c which has a RBF kernel, penalty C=100, misclassification rate 0.0052 and 32 support vectors. In the fourth figure we can see the SVM misclassification plot for model d which has a RBF kernel, kernel bandwith σ =10, misclassification rate 0.0052 and 119 support vectors. From these information the model c seems to be the best since it has a low misclassification rate and quite few support vectors.

	1	2	3
1	313	0	0
2	5	98	0
3	5	0	151

The penalty C (cost of constraints violation) used in model c makes it more expensive for a observation to be placed near the decision boundary, and thus have an impact on how the decision boundary is formed. In the fourth figure where we chose kernel bandwith to 10 we see that the red classification area is bigger, and this is due to that more observations is included in the classification

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 a relatively large sigma, like in this case, gives a larger margin width and all observations that lies inside the margin lines becmoes support vectors.(?)

2.3

Next, a SVM classification model with region as response and all acids in data as predictors is created.

$$\text{Misclassification rate} = \frac{5+5}{5+5+313+98+151} = 0.0175$$

From this classification model the confusion matrix as seen in the table above is produced. The misclassification rate in this classification is less than 2 %. 52 support vector are chosen and the cross validation error (score) is 0.0298. It seems that SVM is a good classification model for these data.

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$LMR - tryC)^2 / 136)
library(kernlab)
set.seed(12345)
epsi <- seq(0.1, 3.5, 0.2)
j <- 1
MSE_SVM <- 0
for(i in epsi){
  LMR_SVM <- ksvm(LMR ~ Day, mrt_rate, type="eps-svr",
                  kernel="rbfdot", epsilon=i)
  MSE_SVM[j] <- sum(((LMR_SVM@fitted - mrt_rate$LMR)^2)/ 136)
  j <- j+1
}

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


eps_mod <- seq(0.1, 3.5, 0.2)
eps_mod <- eps_mod[c(1,6,12,18)]
eps_SVM <- matrix(ncol=4, 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])
  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")
```

```r
points(eps_SVM[,3], x=mrt_rate$Day, type="l", lwd=3, col="purple")
points(eps_SVM[,4], x=mrt_rate$Day, type="l", lwd=3, col="seagreen")
legend(120,-4,c("epsilon=0.1","epsilon=1.1", "epsilon=2.3", "epsilon=3.5"), lty=c(1,1),
       lwd=c(5,5),col=c("blue","brown", "purple", "seagreen"),  cex=0.65)
# Compares original and fitted values, epsilion =2.3
set.seed(12345)
LMR_SVM <- ksvm(LMR ~ Day, mrt_rate, type="eps-svr",
                kernel="rbfdot", epsilon=2.3)

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$Day, type="l", lwd=3, col="seagreen")
legend(120,-4,c("lambda=150","lambda=3", "lambda=10", "epsilon=2.3"), lty=c(1,1),
       lwd=c(5,5),col=c("blue","brown", "purple", "seagreen"),  cex=0.65)
library(fANCOVA)
loessLMR <- loess.as(mrt_rate$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_bwitdhSVM <- ksvm(R2 ~ oleic+linoleic, olive, type="C-svc",
                   kernel="rbfdot", kpar=list(sigma=10))
plot(rbf_bwitdhSVM, 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)
##
```