Lab04: Support Vector Machines

**Handed out:** Wednesday, March 29, 2023

**Return date:** Saturday, April 8, 2023, at the eLearning link **Lab04Submit** in the **Lab04** folder.

**Objectives:** Work with different support vector machine algorithms and datasets

**Grades:** This lab counts 16 % towards your final grade

**Format of answer:** Your answers (statistical figures and verbal description) should be submitted electronically as Word document. Add a running title with the following information: Lab04, your name and page numbers. Use this document as template: add your answers for each subtask, i.e., 1 (a) etc., in a red color as well as any requested statistical figures. Trial and error answers will lead to a deduction of points. You are expected to hand in professionally formatted answers: use a fixed pitch font, like **Courier New**, for any Picture 8 code and output.

Support Vector Machines [16 points]

**Task 1:** You will answer an applied exercise 5 in James et al., 2021. *An Introduction to Statistical Learning with Application in R*. pages 399 and 400. Please follow the sequence of tasks/questions in the exercises*.* [5 points]

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

rm(list=ls()) # Clear environment

oldpar <- par() # save default graphical parameters

if (!is.null(dev.list()["RStudioGD"])) # Clear plot window

dev.off(dev.list()["RStudioGD"])

cat("\014") # Clear the Console

#a

set.seed(12345)

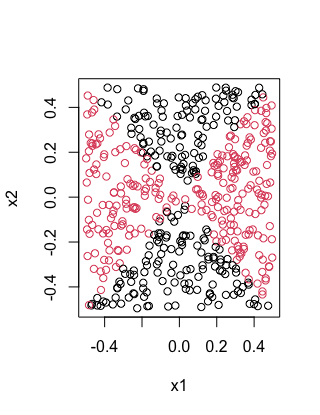
x1 <- runif(500) - 0.5

x2 <- runif(500) - 0.5

y <- 1 \* (x1^2 - x2^2 > 0)

#b

plot(x1, x2, col = y + 1)



#c

#fits a logistic regression model

logit <- glm(as.factor(y) ~ x1 + x2, family = "binomial")

#d

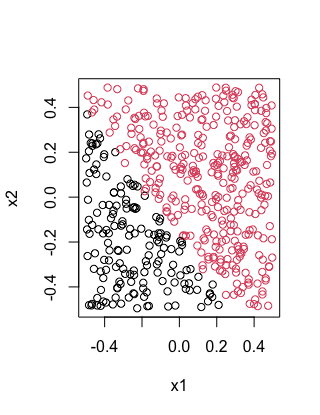
# Make predictions using the logistic regression model

y\_pred <- predict(logit, type = "response")

# Convert predicted probabilities to class labels

y\_pred\_class <- as.numeric(y\_pred > 0.5)

plot(x1, x2, col = y\_pred\_class + 1)



#e

# Create new variables based on x1 and x2

x3 <- x1^2

x4 <- x2^2

x5 <- x1\*x2

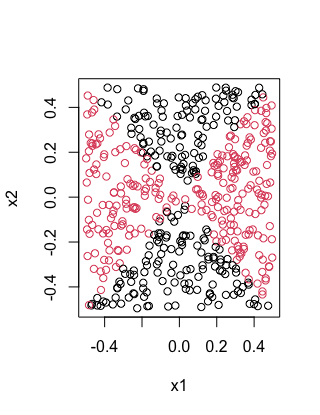
# Fit a logistic regression model with non-linear terms

logit\_nl <- glm(y ~ x1 + x2 + x3 + x4 + x5, family = "binomial")

#f

y\_pred\_nl <- predict(logit\_nl, type = "response")

y\_pred\_class\_nl <- as.numeric(y\_pred\_nl > 0.5)

plot(x1, x2, col = y\_pred\_class\_nl + 1)

#g

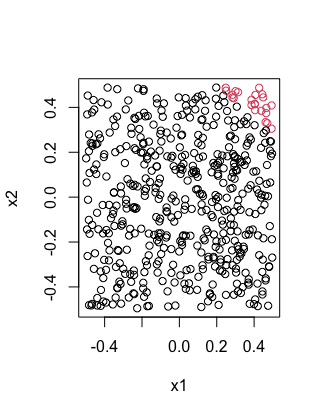
??e1071

library(e1071)

svm\_linear <- svm(y ~ x1 + x2, data = data.frame(x1, x2, y), kernel = "linear")

y\_pred\_svm\_linear <- predict(svm\_linear)

plot(x1, x2, col = y\_pred\_svm\_linear + 1)

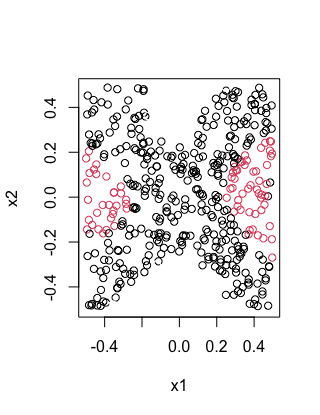


#h

svm\_nonlinear <- svm(y ~ x1 + x2, data = data.frame(x1, x2, y), kernel = "radial")

y\_pred\_svm\_nonlinear <- predict(svm\_nonlinear)

plot(x1, x2, col = y\_pred\_svm\_nonlinear + 1)



(i)

The logistic regression model with a nonlinear transformation of the predictor variables produces a nonlinear decision boundary. Support vector classifiers with linear kernels also produce linear decision boundaries, but support vector machines with nonlinear kernels are able to capture nonlinear patterns in the data and provide better classification boundaries.

**Task 2:**. For the following tasks continue working with the **credit.csv** data set to predict the default probabilities. .[5 points]

[a] Split the data into a stratified training data set with 70% of the observations and a test data set with the remaining 30% of the observations.

# Load the required packages

library(caret)

library(dplyr)

library(e1071)

# Read the data

credit <- read.csv("/Users/jimpan/Documents/EPPS 6326/week files/Weeks06and07/credit.csv")

# Split the data into training (70%) and test (30%) sets

set.seed(123)

trainIndex <- createDataPartition(credit$default, p = 0.7, list = FALSE, times = 1)

train <- credit[trainIndex, ]

test <- credit[-trainIndex, ]

[b] Use a radial kernel support vector classifier. Identify with cross-evaluation the “optimal” cost parameter.

# Set up the tuning grid

tuneGrid <- expand.grid(.sigma = c(0.01, 0.1, 1, 5, 10), .C = c(0.1, 1, 5, 10, 100))

# Set up the train control

control <- trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = TRUE, summaryFunction = twoClassSummary)

# Fit the SVM model with radial kernel using cross-validation

svmFit <- train(default ~ ., data = train, method = "svmRadial", tuneGrid = tuneGrid, trControl = control, preProcess = c("center", "scale"), metric = "ROC")

# Print the optimal cost parameter

svmFit$bestTune

# sigma C

#2 0.01 1

[c] Evaluate your optimal model with the confusion matrix for the test dataset and the ROC curve including the AUC.

library(pROC)

# Predict on the test data using the optimal model

svmPred <- predict(svmFit, newdata = test)

# Convert predicted class labels to factor with levels "No" and "Yes"

svmPred <- factor(svmPred, levels = c("No", "Yes"))

# Convert true class labels to factor with levels "No" and "Yes"

test$default <- factor(test$default, levels = c("No", "Yes"))

svmPred <- factor(ifelse(svmPred == "No", "No", "Yes"), levels = c("No", "Yes"))

# Confusion matrix

confusionMatrix(data = svmPred, reference = test$default)

# ROC curve and AUC

rocObj <- roc(test$default, as.numeric(svmPred))

plot(rocObj)