Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	TDDE01 Machine Learning
Date and time	2022-03-18, 14.00-19.00
Assisting teacher	Oleg Sysoev
Allowed aids	PDF of the course book + your help file (if submitted to LISAM in due time)
Grades:	
	5=18-20 points
	4=14-17 points
	3=10-13 points
	U=0-9 points

Provide a detailed report that includes plots, conclusions and interpretations. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in the appendix.

Note: seed 12345 should be used in all codes that assumes randomness unless stated otherwise!

To start work in RStudio, type this in the Terminal application:

```
module add courses/TDDE01
rstudio
```

To submit your report:

1. Create one file (DOC, DOCX, ODT, PDF)
2. Use Exam Client to submit, and choose Assignment 1 in the drop box
3. Attach your report
4. Submit.
5. "Request Received" status implies that your report is successfully submitted.

Assignment 1 (10p)

The data file **geneexp.csv** contains information about gene expressions of various cells including column CellType showing the type of the cell which is either B-cell or T-cell

Start your work by importing data to R and removing columns containing zeros only. In this assignment, you need to work with the resulting cleaned data.

1. Perform Principal Component Analysis by using the gene expressions data. Report how much variation is explained by the first two principal components. Why is scaling important for these data? Plot the data scores in the coordinate system of the first principal components so that observations are colored according to the cell type. Based on this plot, report whether it should be easy or difficult to classify these data by a machine learning method. **(3p)**
2. Use gene expression data to fit a logistic regression model predicting the cell types from the gene expressions and another logistic regression model using only first two principal components as features to predict the cell types. Report the confusion matrix for the model based on gene expressions and the learned probabilistic model for the model based on principle components. Why are such gene expression data problematic for some machine learning models? **(3p)**
3. Implement a function that depends on parameter **theta** and returns the objective function (cost function) of the binary logistic regression. Report the code and a mathematical formula of this function. Split your data into training and test (50/50) and use this function to evaluate the training and test cost for different iterations of optim() function starting at initial point equal to vector of zeros, using parameter *control=list(maxit=20)* and BFGS method. Plot a dependence of the training and test cost values on the iteration number and answer whether early stopping would be reasonable to do. **(4p)**
    a. **Hint: when debugging your code use** *control=list(maxit=2)* **since optimization process takes some time.**

Assignment 2 (10p)

KERNEL MODELS – 4 POINTS

In the course, you have learned about kernel models for classification and regression. Kernel models can also be used for density estimation, i.e. to model a probability distribution or density function p(x_*). In particular,

$$p(x_*) = \frac{1}{n}\sum_{i=1}^{n} k(\frac{x_* - x_i}{h})$$

where the kernel function k() must integrate to 1. To ensure this, you will hereinafter consider k() to be the density function of a Gaussian distribution with mean equal to 0 and standard deviation equal to 1. You can get it by using the command `dnorm` in R.

Run the code below to produce some learning data, which consist of 1000 samples from class 1 and 1000 samples from class 2. These points are stored in the variables data_class1 and data_class2.

Implement the kernel model presented above to estimate the density function of the data sampled from class 1. Do the same for class 2. Use only 800 samples from class 1 and 800 samples from class 2.

```
set.seed(123456789)

N_class1 <- 1000

N_class2 <- 1000


data_class1 <- NULL

for(i in 1:N_class1){

  a <- rbinom(n = 1, size = 1, prob = 0.3)

  b <- rnorm(n = 1, mean = 15, sd = 3) * a + (1-a) * rnorm(n = 1, mean = 4, sd
= 2)

  data_class1 <- c(data_class1,b)

}


data_class2 <- NULL

for(i in 1:N_class2){

  a <- rbinom(n = 1, size = 1, prob = 0.4)

  b <- rnorm(n = 1, mean = 10, sd = 5) * a + (1-a) * rnorm(n = 1, mean = 15,
sd = 2)

  data_class2 <- c(data_class2,b)

}
```

Once you have kernel models for the class conditional density functions p(x_*|class=1) and p(x_*|class=2), you can use them to produce posterior class probabilities p(class|x_*) via Bayes theorem. Specifically,

p(class=1|x_*) = p(x_*|class=1) p(class=1) / [p(x_*|class=1) p(class=1) + p(x_*|class=2) p(class=2)]

Use these probabilities to compute the correct classification rate on 200 samples that you did not use before, 100 from class 1 and 100 from class 2. Use this classification rate to select the kernel width h from among the values 0.1, 0.2, …, 4.9, 5. Finally, use the 200 samples that you have not used so far to estimate the generalization error of the kernel model selected.

In summary, you should use 1600 samples to build kernel models of the class conditional density functions that you should convert into a probabilistic classifier via Bayes theorem. To select the kernel width, you should use 200 samples as validation set. Finally, you should use 200 samples to estimate the generalization error of the model selected. **Comment your code.**

NEURAL NETWORKS – 6 POINTS

In this exercise, you are essentially asked to repeat the previous kernel method exercise but using this time a neural network as probabilistic classifier. More specifically, you should use 1600 samples as training data, 200 samples as validation data for selecting among two neural network architectures of your choice (i.e., number of layers and units per layer), and 200 samples as test date to estimate the generalization error of the selected architecture. To see how to perform binary classification with the R package `neuralnet`, check the documentation or help file. **Comment your code.**