Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	TDDE01 Machine Learning
Date and time	2020-03-19, 8.00-13.00
Assisting teacher	Oleg Sysoev
Allowed aids	"Pattern recognition and Machine Learning" by Bishop and "The Elements of Statistical learning" by Hastie
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.

- FIRST READ THE FILE **732A99_TDDE01_EXAM_ REGULATIONS.PDF** UNLESS YOU HAVE ALREADY DONE THAT IN ADVANCE
- **Use seed 12345 when randomness is present unless specified otherwise.**
- **Specify RNGversion("3.5.2") in the code**

Assignment 1 (4p)

The data file **Dailytemperature.csv** contains information about the daily temperatures in Stockholm for some time period

1. Create new features ϕ_{1i} and ϕ_{2i} by using variable Day denoted here as x as follows:

$$\phi_{1i} = \sin\left(0.5^i x\right), i = -50, -49, \dots - 1,0,1, \dots 50$$
$$\phi_{2i} = \cos\left(0.5^i x\right), i = -50, -49, \dots - 1,0,1, \dots 50$$

   Use all these 202 features to fit a Lasso regression with target Temperature and plot a dependence of the degrees of freedom on the value of the penalty factor. Explain the trend you observe in this dependence. By using cross-validation, present the dependence of the predicted

MSE on the log-penalty factor and state whether the optimal penalty factor is statistically significantly better than log-penalty factor equal to -4. Present also the number of non-zero features corresponding to the optimal penalty factor. Finally, present the time series plot of the original and the fitted data corresponding to the optimal penalty factor and comment on the quality of fit.

Assignment 2 (6p)

In this assignment, you will work with dataset "mtcars" present in R base library. Type *mtcars* in R console to inspect the data or *?mtcars* to see the meaning of different variables.

1. Use a reduced data set with only variables *mpg* and *hp* in order to extract the direction of the first principle component: do this by using function *eigen()* and other basic R functions and report the components of the first principle component. Finally, plot the reduced data and the line that shows the first principle component direction of the data. State whether this direction looks reasonable according to what PCA is supposed to do. **(3p)**
2. Divide the full data set into training and validation sets (50/50) and use the holdout principle to compute a classification tree of the optimal size in which *am* variable is target and all remaining variables are features. Provide a plot showing dependence of the training and validation error on the number of leaves and select the optimal tree size. Plot this tree and interpret the decisions provided by it. Finally, remove observation number 7 (seventh row) from your training data, fit the tree again (without holdout) and compare the obtained unpruned tree with the unpruned tree computed for the full training data. What is the main theoretical reason behind the difference found? **(3p)**

Assignment 3 (10p)

KERNEL METHODS – 4 POINTS

(2.5 p) You are asked to produce a kernel model to predict the sine function in the interval [0, 20]. You have at your disposal 100 training points drawn from this interval. See the code below. You **must** use the Gaussian kernel. Run your code to predict the target value for each training point from the rest of the training points, i.e. **discard** the training point you are trying to predict. Do this for kernel width values 0.1, 1, and 10. Report the mean square error in each case. Plot also the predictions and the ground truth.

(1.5 p) Finally, **answer** the following question: Say that the kernel width 0.1 got the smallest mean squared error. Would you use this value to make future predictions or do you think that there is a risk of overfitting ? In other words, do you think that it is wise to select the kernel width according to the scheme described ?

```
# Code for sampling the training data.

n <- 100
```

```
Var <- runif(n, 0, 20)

tr <- data.frame(Var, Sin=sin(Var))


# Code for plotting the predictions alongside the ground truth.

plot(tr[,1],predictions, col="blue", cex=3, ylim = c(-1.1,1.1))

points(tr, col = "red", cex=3)
```

SUPPORT VECTOR MACHINES – 3 POINTS

Run the code below and interpret the resulting plots. In particular, explain why they differ.

```
library(kernlab)

Var <- runif(50, 0, 10)

tr <- data.frame(Var, Sin=sin(Var))


svm1 <- ksvm(Sin~.,data=tr,kernel="rbfdot",kpar=list(sigma=1),C=.1)

plot(tr[,1],predict(svm1,tr), col="blue", cex=3, ylim = c(-1.1,1.1))

points(tr, col = "red", cex=3)


svm2 <- ksvm(Sin~.,data=tr,kernel="rbfdot",kpar=list(sigma=1),C=1)

plot(tr[,1],predict(svm2,tr), col="blue", cex=3, ylim = c(-1.1,1.1))

points(tr, col = "red", cex=3)
```

NEURAL NETWORKS – 3 POINTS

The code below trains a neural network to predict the sine function in the interval [0, 10]. In the code, there are two variables: Var and Sin. The code trains a NN to predict Sin from Var. You are now asked to try to predict Var from Sin. You **must** use the same settings and training data as in the code below, i.e.

you are only allowed to swap the roles of Var and Sin. **Answer** the following question: Why do you think that the new results are better/worse than the original ones ?

```
library(neuralnet)

set.seed(1234567890)

Var <- runif(50, 0, 10)

tr <- data.frame(Var, Sin=sin(Var))

winit <- runif(31, -1, 1)

nn <- neuralnet(formula = Sin ~ Var, data = tr, hidden = 10, startweights = winit, threshold = 0.02, lifesign = "full")

plot(tr[,1],predict(nn,tr), col="blue", cex=3)

points(tr, col = "red", cex=3)
```