Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	732A95 Introduction to Machine Learning
Date and time	2016-01-09, 08.00-13.00
Assisting teacher	Oleg Sysoev
Allowed aids	"Pattern recognition and Machine Learning" by Bishop and "The Elements of Statistical learning" by Hastie
Grades:	A=19-20 points B=16-18 points C=11-15 points D=9-10 points E=7-8 points F=0-6 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. Use seed 12345 when randomness is present unless specified otherwise.

Assignment 1 (10p)

Dataset **crx.csv** contains encrypted information about the customers of a bank and whether each individual has paid back the loan or not:

- Class: 1=paid back, 0=not paid back

1. Divide the dataset into training and test sets (80/20), use seed 12345. Fit a decision tree with default settings to the training data and plot the resulting tree. Finally, remove the second observation from the training data, fit the tree model again and plot the tree. Compare the trees and comment why the tree structure changed so much although only one observation is deleted. **(2p)**
2. Prune the tree fitted to the training data by using the cross-validation. Provide a cross-validation plot and comment how many leaves the optimal tree should have. Which variables were selected by the tree? **(3p)**

3. Fit a GAM model with features A3 and A9, comment on the choice of *family* parameter. Why is it pointless to include a spline component of A9? Provide an equation of the fitted model. Comment which components are significant. Plot the spline component of A3 and interpret the plot. **(3p)**
4. Use the following error function to compute the test error for the GAM and tree models: $E = \sum_i Y_i \log \hat{p}_i + (1 - Y_i) \log(1 - \hat{p}_i)$, where Y_i are target values and $\hat{p}_i$ are predicted probabilities of $Y_i = 1$. Which model is the best according to this criterion? Why is this criterion sometimes more reasonable to use than the misclassification rate? **(2p)**

Assignment 2 (10p)

ENSEMBLE METHODS

1. Interpret the plot resulting from the code below. **(1p)**

```
library(mboost)
bf <- read.csv2("bodyfatregression.csv")
set.seed(1234567890)
m <- blackboost(Bodyfat_percent~Waist_cm+Weight_kg, data=bf)
mstop(m)
cvf <- cv(model.weights(m), type="kfold")
cvm <- cvrisk(m, folds=cvf, grid=1:100)
plot(cvm)
mstop(cvm)
```

SUPPORT VECTOR MACHINES

In the following steps, you are asked to use the R package *kernlab* to learn a SVM for classifying the *spam* dataset that is included with the package. For the *C* parameter, consider values 1 and 5. Consider the radial basis function kernel (also known as Gaussian) and the linear kernel. For the former, consider a width of 0.01 and 0.05. This implies that you have to select among six models.

2. Use nested cross-validation to estimate the error of the model selection task described above. Use two folds for inner and outer cross-validation. Note that you only have to implement the outer cross-validation: The inner cross-validation can be performed by using the argument cross=2 when calling the function ksvm. **Hint**: Recall that inner cross-validation estimates the error of the different models and selects the best, which is then evaluated by the outer cross-validation. So, the outer cross-validation evaluates the model selection performed by the inner cross-validation **(3p**)
3. Produce the code to select the model that will be returned to the user. **(1p)**

NEURAL NETWORKS

4. Implement the backpropagation algorithm for fitting the parameters of a NN for regression. The NN has one input unit, 10 hidden units, and one output unit. Use the *tanh* activation function. Recall that you have an example on Bishop's book as well as on the course slides. Feel free to

use stochastic or batch gradient descent. Please use only basic R functions in your solution, e.g. *sum*, *tanh*. **(4p)**

5. Run your code for 5000 iterations (if time permits) on the training data *tr* below. A learning rate in the interval $\left[\frac{1}{25^2}, \frac{1}{25}\right]$ should work fine. Plot the error on *tr* as well as on the validation data *va*, as a function of the number of iterations. **(1p)**

```
set.seed(1234567890)
Var <- runif(50, 0, 10)
trva <- data.frame(Var, Sin=sin(Var))
tr <- trva[1:25,] # Training
va <- trva[26:50,] # Validation
```