Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	732A99/732A68 Machine Learning
Date and time	2023-01-09, 8.00-13.00
Assisting teacher	Oleg Sysoev
Allowed aids	PDF of the course book + your help file (if submitted to LISAM in due time)
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.

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/732A99
rstudio
```

Assignment 1 (10p)

File **Rice.csv** contains a total of 3810 rice grain's images taken for the two species (Cammeo and Osmancik), which were processed and feature inferences were made. 7 morphological features were obtained for each grain of rice which are the variables in the data set.

1. Scale the data set and compute the optimal LASSO classification model that uses Class as target variable and remaining variables as features, by means of cross-validation. Report the optimal

penalty value, which features are selected by the model, and the plot showing how the cross-validation error depends on the penalty factor. Explain by using the plot whether the model corresponding to the optimal penalty factor is statistically significantly better than LASSO models having one feature less. Finally, by using the loss matrix

$$\begin{matrix} & Pred.\ Cammeo & Predicted\ Osmancik \\ True\ Cammeo & \begin{pmatrix} 0 & 37 \\ 3 & 0 \end{pmatrix} \\ True\ Osmancik & \end{matrix}$$

compute the confusion matrix for the entire data set and comment on the patterns of the confusion matrix and why such patterns are observed. **(4p)**

2. Scale the dataset and compute principal components by using all numerical variables of the dataset. Use the transformed data set with the new features (all principal components) and the target variable and divide it into training, validation, and test sets (40/30/30). After this, compute 7 logistic regression models with target Class and features $PC_1, \dots PC_m$, where $m = 1, \dots 7$ (so each model is using first m principal components). Compute training and validation misclassification errors for each model and plot a dependence of these errors on m. Explain this plot in terms of bias-variance tradeoff. Which value of m is the optimal one? Compute also the test error for the optimal model and compare it with the training and validation error and make necessary conclusions. **(4p)**
3. Assume that PCA is done in the same way as in step 2 and two principal components are finally kept. Report the equation showing how the first original (unscaled) variable (Area) can be expressed as a function of these two principal components. **(2p)**

Assignment 2 (10p)

ENSEMBLE MODELS

You are asked to implement the AdaBoost algorithm for binary (-1/+1) classification as it appears in the course textbook and slides. You can find the pseudocode below. **Comment your code**. **(5 p)**

Learn an AdaBoost classifier

Data: Training data $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$
Result: B weak classifiers

Assign weights $w_i^{(1)} = 1/n$ to all data points.
for $b = 1, \dots, B$ **do**
  Train a weak classifier $\widehat{y}^{(b)}(\mathbf{x})$ on the weighted training data $\{(\mathbf{x}_i, y_i, w_i^{(b)})\}_{i=1}^n$.
  Compute $E_{\text{train}}^{(b)} = \sum_{i=1}^n w_i^{(b)} \mathbb{I}\{y_i \neq \widehat{y}^{(b)}(\mathbf{x}_i)\}$.
  Compute $\alpha^{(b)} = 0.5 \ln((1 - E_{\text{train}}^{(b)})/E_{\text{train}}^{(b)})$.
  Compute $w_i^{(b+1)} = w_i^{(b)} \exp(-\alpha^{(b)} y_i \widehat{y}^{(b)}(\mathbf{x}_i)), i = 1, \dots, n$.
  Set $w_i^{(b+1)} \leftarrow w_i^{(b+1)} / \sum_{j=1}^n w_j^{(b+1)}$, for $i = 1, \dots, n$.
end

Predict with the AdaBoost classifier

Data: B weak classifiers with confidence values $\{\widehat{y}^{(b)}(\mathbf{x}), \alpha^{(b)}\}_{b=1}^B$ and test input $\mathbf{x}_\star$
Result: Prediction $\widehat{y}_{\text{boost}}^{(B)}(\mathbf{x}_\star)$

Output $\widehat{y}_{\text{boost}}^{(B)}(\mathbf{x}_\star) = \text{sign}\left\{\sum_{b=1}^B \alpha^{(b)} \widehat{y}^{(b)}(\mathbf{x}_\star)\right\}$.

Method 7.2: AdaBoost

As weak classifier, you should use a decision stump. This is essentially a one-level decision tree. To learn the decision stump, we recommend you to consider (i) sorting the data in ascending order according to one of the predictive attributes, (ii) sorting the instance weights and true labels according to the same ordering, (iii) computing the cumulative sum cumsum(w * y) where w and y are the sorted instance weights and true labels, and (iv) identifying the threshold value that maximizes or minimizes the cumulative sum (if maximized then return +1, and if minimized then return -1). Repeat the previous step for every predictive attribute and choose the best one. **Comment your code**. **(2.5 p)**

Finally, you are asked to run your code on the following data

```
x1 <- runif(10000,0,10)

x2 <- runif(10000,-2,2)

y <- ifelse(x2 > sin(x1),1,-1)

plot(x1,x2,col = y + 2)

allData <- cbind(x1,x2,y)

D <- 2
```

The data consists of two predictive attributes and the class label. The data contains 10000 instances. Use 500 for training and the rest for testing. Report the results when using B=1, 2, …, 50 weak classifiers in AdaBoost. It may be a good idea to split the data at random into training and test a number of times for each value of B and report the average performance. **Comment your results**. Feel free to use the following template. **(2.5 p)**

```
stumpLearn <- function(…){}

stumpPredict <- function(…){}

AdaBoostLearn <- function(...){}

AdaBoostPredict <- function(...){}

res <- NULL

for(B in 1:50){

  res2 <- NULL

  for(i in 1:10){

    foo <- sample(1:10000,500)

    dat <- allData[foo,]

    tes <- allData[-foo,]

    foo <- AdaBoostLearn(dat,…)
```

```
    res2 <- c(res2,mean(AdaBoostPredict(tes[,1:D],...) != tes[,D+1]))
  }
  res <- c(res,mean(res2))
}
plot(res, type = "l")
```