**The Elements of Statistical Learning, WS 2017/18**
Prof. Dr. Dr. Thomas Lengauer
Michael Scherer
D3: Computational Biology & Applied Algorithmics
Max Planck Institute for Informatics

2018-01-03

# Problem Set 6

**Deadline:** Wednesday, January 17. 2018, 10:00 a.m.

**Please read and follow the following requirements to generate a valid submission.**
This problem set is worth 50 points. You may submit your solutions in groups of two students. The solutions to the theoretical problems should be submitted either digitally (in .pdf format) to mscherer@mpi-inf.mpg.de or as a hard copy before the lecture. **Label your hard copy submissions with your name(s).**
Solutions to programming problems and resulting plots need to be submitted in digital format (.pdf). For the programming problems you have to submit an executable version of your code (R script).

For digital submissions the subject line of your email should have the following format:

`[SL][problem set 6] lastname1,firstname1;lastname2,firstname2`

Please include the numbers of the problems you submitted solutions to (both digitally and analogously) in the email's body. **Please make sure that all the files are attached to the email.** The attached files should only include an executable version of your code as .R file and **one** .pdf file with all the other solutions.

## Problem 1 (T, 10 Points)

**MARS: Multivariate Adaptive Regression Splines**
MARS is an adaptive procedure for regression, which uses pairs of piecewise linear basis functions (a sort of very simple splines that is also called *reflected pairs*) of the form $(x-t)_+$ and $(t-x)_+$ with "+" denoting the positive part, e.g.,

$$(x-t)_+ = \begin{cases} x-t & \text{if } x > t, \\ 0, & \text{otherwise.} \end{cases}$$

At the value $t$, the function has a knot and for each predictor $X_j$, we generate basis function pairs with knots at every observed value $x_{ij}$, such that we have a collection of basis function pairs

$$C = \{(X_j - t)_+, (t - X_j)_+\}, \qquad t \in \{x_{1j}, x_{2j}, ..., x_{Nj}\}, \quad j = 1, 2, ..., p.$$

The trained model uses products of the basis functions from $C$ instead of the original inputs and has the form

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X),$$

where $h_m(X)$ is one function from $C$, or a product of two or more such functions. All $h_m$ form the model set $\mathcal{M}$. Model training is done similarly to forward stepwise linear regression, such that iteratively new terms are added to the model set $\mathcal{M}$. Initially, we start with $\mathcal{M}$ containing only the constant function $h_0(X) = 1$ and add in each step a product of a function $h_l \in \mathcal{M}$ with one reflected pair from $C$. This product is chosen such that adding the term

$$\hat{\beta}_{M+1} h_l(X) \cdot (X_j - t)_+ + \hat{\beta}_{M+2} h_l(X) \cdot (t - X_j)_+$$

to our model decreases the training error most. All coefficients $\hat{\beta}_0, .., \hat{\beta}_{M+2}$ are estimated together using least squares. This process ends when some preset maximum number of terms are contained in $\mathcal{M}$. In order to avoid overfitting, the final model is cropped, i.e., one iteratively removes that term whose removal leads to a minimum increase in training error. This gives us for each possible number of terms an estimated best model. Among all these models, the best one is chosen using generalized cross-validation.

(a) How do you have to change the procedure of generating a MARS model to make a decision tree?

(b) Can you argue on the basis of the relationship between MARS and decision trees revealed in (a) what is an advantage of MARS over decision trees and what is an advantage of decision trees over MARS?

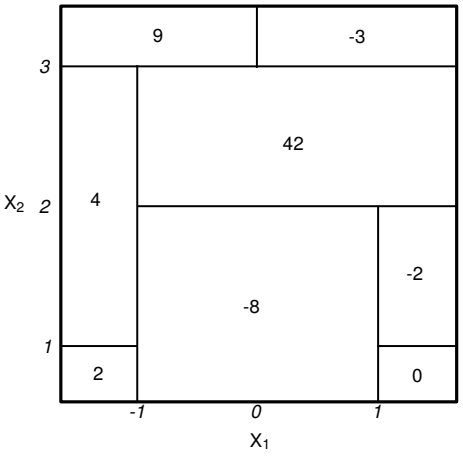
## Problem 2 (T, 5 Points)

**Loss functions**
In the regression setting, two different loss functions are quite popular: squared error loss $L(y, f(x)) = (y - f(x))^2$ and absolute loss $L(y, f(x)) = |y - f(x)|$. Squared error loss is differentiable at zero, but the values increase very strongly in regions far away from the origin, which makes the loss function quite sensitive to outliers. Absolute loss does not have the latter drawback but it is not differentiable at zero. Define a loss function which combines the two advantages, i.e., which is differentiable everywhere, mimics squared error loss for values $y$ such that $|y - f(x)| < \delta$ for some threshold $\delta$ and is linear for all other $x$.

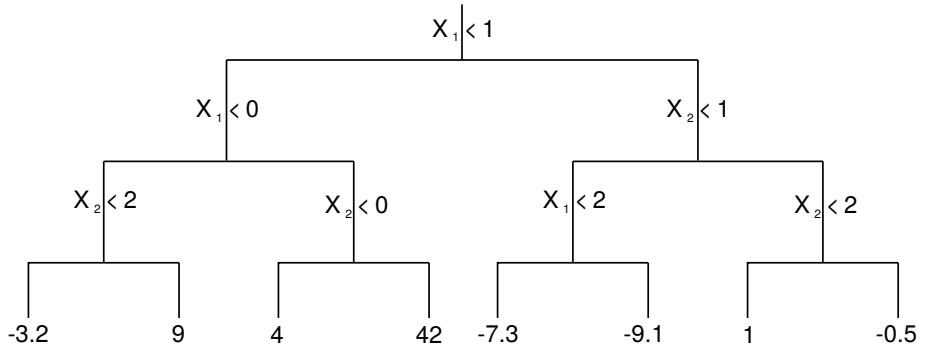
## Problem 3 (T, 15 Points)

(modified version of Exercise 8.4.4 in ISLR)

(a) (5 P) Sketch the tree corresponding to the partition of the predictor space indicated in the figure below. The numbers inside the boxes indicate the mean of $Y$ within each region.




(b) (5 P) Create a diagram similar to the one provided in a), using the tree illustrated below. You should divide up the predictor space into the correct regions, and indicate the mean for each region.




(c) (5 P) Create another tree representing the same partition of the predictor space as the one discussed in b), but with a different split at the root node.

## Problem 4 (P, 20 Points)

Go through **7.8 Lab: Non-Linear Modeling** (ISLR p.287–297) and **8.3 Lab: Decision Trees** (ISLR p.324–331).

(Adapted from Exercise 8.4.10 in ISLR) We use boosting to predict **Salary** in the **Hitters** data set. Load the **ISLR** library, which contains the data.

(a) (2P) Remove the observations for whom the salary information is unknown, and then log-transform the salaries. Create a training set consisting of the first 200 observations and a test set consisting of the remaining observations.

(b) (5P) Perform boosting on the training set with 1000 trees for a range of values of the shrinkage parameter $\lambda$. Produce a plot showing training and test set MSE for the different shrinkage values. Comment on your observations, choose a value for $\lambda$ and justify your choice.

(c) (4P) Compare the test MSE of boosting to the MSE that results from applying least squares regression and ridge regression.

(d) (3P) Which variables appear to be the most important predictors in the boosted model? Compare these to the ones that appear to be important in the models you generated in (c).

(e) (4P) Now apply bagging to the training set. Use decision stumps (A decision stump is a tree classifier which only makes one Guillotine cut parallel to one of the axes.) and vary the number of trees. Plot train and test MSE as a funtion of the number of trees. What can you observe? Compare them to the previously obtained train and test errors.

(f) (2P) Apply random forests to the training set. Report train and test set MSE and comment on the importance of the individual predictors. Compare them to the previously obtained train and test errors.