SKETCH of the SOLUTIONS
STK-IN4300/STK-IN9300 - 2019

Problem 1

a Both ridge regression and principal component regression act on the principal components of the X matrix: the former by shrinking their regression coefficients, the latter by setting them to 0. In particular, ridge regression shrinks more the components with smaller eigenvalues, while the only "shrinkage effect" of the principal component regression is given by the elimination of the information related to the principal components whose coefficients are set to 0.

b
- Because the intercept is excluded from the penalization, as it makes no sense to shrink it toward 0;
- one obtains the group lasso and the ordinary lasso, respectively;
- "bet on sparsity" is that principle for which it is preferable to use a procedure which assumes a sparse truth over one that does not, because the former performs better if the problem is actually sparse, while both procedures tend to perform badly in a dense problem.

c Bridge regression is a penalize regression approach which uses a penalty of the form $\sum_{i=1}^{p} |\beta_i|^q$, where p is the number of explanatory variables and β_i the regression coefficients. When $1 \leqslant q \leqslant 2$, the penalty can be seen as a compromise between the lasso ($q = 1$) and the ridge ($q = 2$) penalties.

Similarly, elastic net is also a compromise between lasso and ridge regression: in this case, a mixture of L_1 and L_2 penalties is used, with a hyperparameter controlling the ratio between the two penalties. The resulting methods is supposed to enclose the advantages of both penalties (mainly, variable selection and a better handling of correlation, respectively).

The difference in the models that one obtains by applying bridge regression and elastic net is related to the form of the constraints: while similar, that of elastic net has non-differentiable corners that lead to sparser models, as some regression coefficients are forced to be exactly 0.

Problem 2

a Consider three binary classification trees for a new point x_0. Two of them classifies it as 0 with probability 0.55 (and 1 with probability 0.45), the third with probability 0.10 (so 1 with probability 0.90). When aggregating based on the estimated class, x_0 will be classified as 0 (two votes versus one). In contrast, by considering the probabilities, 1 (averaging the probability, one obtains 0.40 for 0, 0.60 for 1).

b Since random forests only consider a subset of variables for the tree splitting, an increasing number of noisy variables increases the probability of having subsets which do not include any variable correlated with the outcome.

c The left-hand side of the equation is the signal-to-noise ratio (larger the value, larger the ratio), while the right-hand side is a quantity related to the shrinkage of the base learner: the stronger the shrinkage (i.e., the smaller λ_k), the smaller the quantity. Therefore, in order for boosting to bring improvement, the signal must be large with respect to the noise, or the shrinkage sufficiently strong.

d
- The main tuning parameters are the number of boosting steps m_{stop} and the boosting step size ν;
- usually one fixes the boosting step size (e.g., $\nu = 0.1$) and compute the best value of m_{stop} via cross-validation;
- As long as the boosting step size has the right magnitude, it does not really influence the prediction performance. If it is too small, the procedure results to be too slow, while there is the risk to overfit if its value is set too large. In contrast, the number of boosting steps is extremely important to obtain a good prediction: too many boosting steps lead to overfitting (large variance), not enough boosting steps lead to underfitting (large bias).
- if the goal is to obtain the best prediction model, a smaller value of the boosting step size requires a larger number of boosting steps, and vice versa.

Exercise 3

a (see formula (7.9) in the book)

b (see the proof of Proposition 3 of Bülmann & Yu, JASA 2003)

c Model complexity is strictly connected with the concept of bias-variance trade-off: a more complex model tends to have smaller bias and larger variance, while it is the other way around for less complex models. One can see this in kNN. The most complicated model is obtained for $k = 1$, in which there is one average (the predicted outcome) for each observation: in this case, the bias in the training set is 0, but the variance (and, consequently, the expected predicted error) is very large. At the opposite, $k = n$, with n the number of observations, only one average is computed (so the model is the less complex that can be fitted), causing a very large bias and a large expected prediction error (despite the small variance). To obtain a good bias-variance trade-off, the right model complexity must be found.