SKETCH of the SOLUTIONS
STK-IN4300/STK-IN9300 - 2018

Exercise 1

a When using the squared-error loss, the expected prediction error can be decomposed into three parts:

- an irreducible error, which cannot be avoided;
- the squared bias, where bias is the difference between the average of the estimate and the true mean;
- a variance term, the expected square deviation of our estimate from its mean.

When minimizing the expected prediction error, we are in a situation in which if we reduce the variance component, for example by adding constraints to the model space, we increase the squared bias, and vice versa. This is called bias-variance trade-off. In the picture this is represented by lines and circles: when adding constraints, o.e. when we move from the model space to the restricted model space, we increase the bias (represented by the line named “Estimation Bias") but we reduce the variance (the radius of the circle around the “Closest fit in population” point is smaller than that around the corresponding point on the restricted model space).

b Bias:

$$E[\hat{\beta}_{\text{LS}}] = \beta$$

$$\begin{aligned} E[\hat{\beta}_{\text{ridge}}(\lambda)] &= E[(X^TX + \lambda I_p)^{-1}X^Ty] \\ &= E[(I_p + \lambda(X^TX)^{-1})^{-1} \underbrace{(X^TX)^{-1}X^Ty}_{\hat{\beta}_{\text{LS}}}] \\ &= \underbrace{(I_p + \lambda(X^TX)^{-1})^{-1}}_{w_\lambda} E[\hat{\beta}_{\text{LS}}] \\ &= w_\lambda\beta \implies E[\hat{\beta}_{\text{ridge}}(\lambda)] \neq \beta \text{ for } \lambda > 0. \end{aligned}$$

Variance:

$$\text{Var}[\hat{\beta}_{\text{LS}}] = \sigma^2(X^TX)^{-1}$$

$$\begin{aligned} \text{Var}[\hat{\beta}_{\text{ridge}}(\lambda)] &= \text{Var}[w_\lambda\hat{\beta}_{\text{LS}}] \\ &= w_\lambda \text{Var}[\hat{\beta}_{\text{LS}}]w_\lambda^T \\ &= \sigma^2 w_\lambda(X^TX)^{-1}w_\lambda^T. \end{aligned}$$

Then,

$$\begin{aligned}
&\mathrm{Var}[\hat{\beta}_{\mathrm{LS}}] - \mathrm{Var}[\hat{\beta}_{\mathrm{ridge}}(\lambda)] = \sigma^2 \left[(X^T X)^{-1} - w_\lambda (X^T X)^{-1} w_\lambda^T\right] \\
&\quad = \sigma^2 w_\lambda \left[(I_p + \lambda(X^T X)^{-1})(X^T X)^{-1}(I_p + \lambda(X^T X)^{-1})^T - (X^T X)^{-1}\right] w_\lambda^T \\
&\quad = \sigma^2 w_\lambda \left[((X^T X)^{-1} + 2\lambda(X^T X)^{-2} + \lambda^2 (X^T X)^{-3}) - (X^T X)^{-1}\right] w_\lambda^T \\
&\quad = \sigma^2 w_\lambda \left[2\lambda(X^T X)^{-2} + \lambda^2 (X^T X)^{-3})\right] w_\lambda^T > 0
\end{aligned}$$

(since all terms are quadratic and therefore positive)

$$\implies \mathrm{Var}[\hat{\beta}_{\mathrm{ridge}}(\lambda)] \preceq \mathrm{Var}[\hat{\beta}_{\mathrm{LS}}]$$

Exercise 2

a The procedure is incorrect because the prediction error is computed on observations already used to train the prediction rule. This lead to underestimation of the error (writing "too optimistic" was acceptable).

A possible solution is to compute the prediction error only on those observations (in average 36.8% of the original sample) not included in the bootstrap sample. Since this approach leads to overestimating the prediction error, solutions like those described in the point (b) has been implemented.

b The `0.632 bootstrap` procedure addresses the problem of overestimation of the correct procedure described in point (a) by averaging it (with weight 0.632 and 0.368, respectively) with the training error (underestimated error). The result is a sort of compromise between overestimation and underestimation.

In formula:

$$\widehat{\mathrm{Err}}^{(0.632)} = 0.632\ \widehat{\mathrm{Err}}^{(1)} + 0.368\ \overline{\mathrm{err}},$$

where $\overline{\mathrm{err}}$ is the training error and $\widehat{\mathrm{Err}}^{(1)}$ the corrected procedure described at point (a).

Since the 0.632 and 0.382 weights may not be the best choice (e.g., in case of complete overfitting in the training set), the `0.632+ bootstrap` has been developed. In the latter procedure,

$$\widehat{\mathrm{Err}}^{(0.632+)} = \hat{w}\ \widehat{\mathrm{Err}}^{(1)} + (1 - \hat{w})\ \overline{\mathrm{err}},$$

the weights depend on the relative overfitting rate, so the `0.632+ bootstrap` can be seen as a better compromise between the overestimation and underestimation of the prediction error done by the corrected procedure described at point (a) and the training error, respectively.

Exercise 3

a The penalization term penalizes curves too "wiggly", reducing the model complexity by penalizing curves with high curvature. The amount of penalty is controlled by the tuning parameter λ:

- when $\lambda = 0$ there is no penalization, and it leads to a curve which passes though all the points;
- when $\lambda = \infty$ no curvature is allowed, and it leads to a straight line.

The choice of λ is also a case of bias-variance trade-off: smaller λ, smaller the bias (and higher the variance); larger λ, larger the bias (and smaller the variance).

b Rewriting equation (1) in terms of θ, i.e. by plugging in $f(x) = \sum_{i=1} N_j(x)\theta_j$, one obtain the form

$$RSS(\theta, \lambda) = (y - N\theta)^T(y - N\theta) + \lambda\theta^T\Omega_N\theta,$$

where $\{N\}_{ij} = N_j(x_i)$ and $\{Omega_N\}_{jk} = \int N''(t)N''(t)dt$.

Either deriving (and setting the first derivative equal to 0) or recognizing the solution of a regularized ridge regression, one obtains

$$\theta = (N^TN + \lambda\Omega_N)^{-1}N^Ty.$$

Therefore

$$\hat{f} = N(N^TN + \lambda\Omega_N)N^Ty,$$

which is linear in y. Knowing that the degrees of freedom of a linear smoother correspond to the trace of the smoothing matrix,

$$\text{dof}(\hat{f}) = \text{trace}(N(N^TN + \lambda\Omega_N)N^T).$$

Exercise 4

a Bagging (Bootstrap AGGregatING) is a procedure which consists in aggregating (by averaging, by voting, etc.) the results of a prediction rule applied to a number of bootstrap samples generated from the original data. The prediction rule is typically (but not necessarily) a tree.

An advantage of bagging with respect to a single tree is its stability (reduced variance), while in contrast to a boosted tree model bagging is not able to take advantage of the results of the previous iterations to improve the later predictions (e.g., in the context of classification, AdaBoost is able to focus on misclassified observations by weighting them more in the later iterations).

b Consider a simple case of binary classification in which we aggregate the results of three trees: two trees say that an observation x_i is of class A with probability 0.55, of class B with probability 0.45; the third tree, instead, says A with probability 0.1, B with probability 0.9.

When aggregating by "majority of votes", x_i is classified as A (two votes against one).

When aggregating by "class probabilities", x_i is classified as B (average probabilities being 0.4 for A, 0.6 for B).

Exercise 5

a The solution was seen in class during the 11th lecture (solution of exercise 10.2 of the text book), see also here.

b The algorithm is called "gradient boosting" and the omitted expression is

$$u_m = - \left. \frac{\partial L(y, f(x))}{\partial f(x)} \right|_{f(x)=f_{m-1}(x)}$$