

SKETCH of the SOLUTIONS

STK-IN4300/STK-IN9300 - 2021

Problem 1

- a Elastic Net is a penalised regression approach, in which the penalty term is a mixture of L_1 and L_2 penalties. For this reason, it can be seen as a compromise between lasso and ridge regression. The amount of L_1 penalty is controlled by the hyperparameter λ_1 , so that Elastic Net is equivalent to lasso if $\lambda_1 = 0$. Similarly, the amount of L_2 penalty is also controlled by an hyperparameter, in this case λ_2 . When λ_2 is equal to 0, Elastic Net is equivalent to ridge regression.

As a combination between L_1 and L_2 penalties, Elastic Net has the advantage to have both the variable selection feature of lasso and the ability of handling correlation of ridge regression. The main disadvantage is related to the presence of two hyperparameters: it is computationally costly to tune two hyperparameters and it is not so easy to find the best combination of the two.

- b Like Elastic Net, Bridge regression is a penalize regression approach that relies on two hyperparameters. When $1 \leq q \leq 2$, in particular, it can also be seen as a compromise between the lasso ($q = 1$) and the ridge regression($q = 2$).

The difference, in the case of $1 \leq q \leq 2$, is related to the form of the constraints: while similar in shape, that of elastic net has non-differentiable points that lead to sparser models, as some regression coefficients are forced to be exactly 0. When $1 \leq q \leq 2$, this is not the case for bridge regression. The two approaches are equivalent when $\lambda_1 = 0$, $q = 1$ and $\lambda_2 = \lambda$ and when $\lambda_2 = 0$, $q = 2$ and $\lambda_2 = \lambda$. These case are those corresponding to lasso and ridge regression, respectively.

Problem 2

- a The difference lies in the selection of the variables. Ho's approach uses the same m variables for the entire construction of the tree on each bootstrap sample, Breiman's version samples the m variables at each tree split.

We expect that Ho's algorithm performs worse than Breiman's one in this specific simulation example: since the selection of the m variables happens only once per tree, there are more chances that all the splits are done only considering noisy variables.

- b The random forest regression estimator is defined as the expected value of the single tree, such that

$$\hat{f}_{RF}(x) = E_{\Theta|Z}[T(x; \Theta(Z))],$$

where Θ includes all the characteristics of the individual sample tree (split variables, cut-points, leaves' values).

Then the bias is

$$f(x) - E_Z[E_{\Theta|Z}[T(x; \Theta(Z))]] = f(x) - E_{\Theta,Z}[T(x; \Theta(Z))]$$

i.e., the bias of the individual sample tree.

About the unpruned tree grown to Z , it has smaller bias because it allows more complex estimated functions: the individual trees that compose the random forest are allowed to only use $m < p$ variables at each split. Indeed, in general, increasing m decreases the bias.

Problem 3

- a The effect of the solar radiation on the ozone increases almost linearly until around 230 Langleys, after which it seems to have a negative effect, i.e. a larger amount of radiations exceeding 230 Langleys will mean a smaller concentration of ozone. If the model had only considered a linear effect, this change would have not been modelled, and a slightly increase in the concentration of ozone would have been modelled for each increase of 1 Langsley in solar radiation.
- b See [solution to exercise 5.7](#).

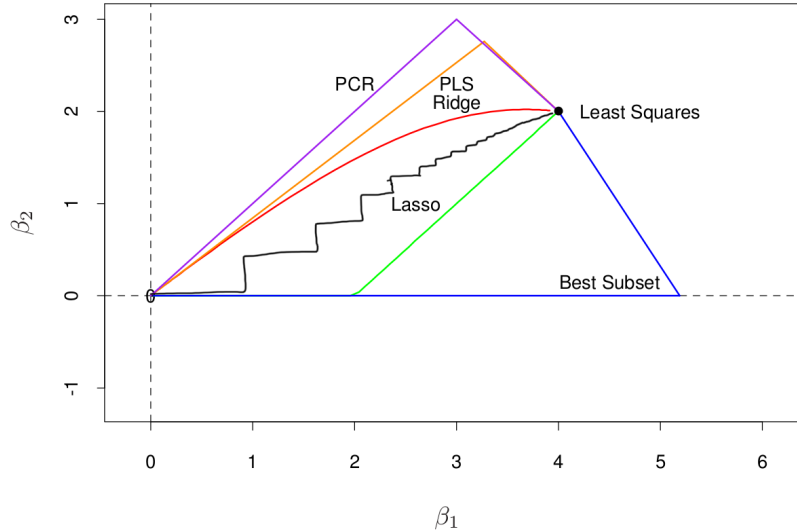
Problem 4

- a The figure describe the learning path of several algorithms in a low-dimensional situation. In particular:

- PCR and ridge regression have paths not so far from each other because they both work on the directions of the principal components. While the former only allows the principal components to be fully in or fully out of the model (in particular the point (3, 3) is reached when the model only includes the first principal component), the latter shrinks the estimates smoothly in both directions, penalising more in the direction of the second principal component. In contrast to PCR, all points of the ridge regression paths are possible, depending on the amount of penalty applied;
- the Lasso path is also a continuous path, where each point corresponds to a value of the penalty parameter, ranging from 0 (the point denoted by “Least Squares” in the figure) to ∞ (the origin). In (2,0) the path changes direction because the penalty term becomes weak enough to allow both variables to enter into the model;
- the Best subset estimate of $\hat{\beta}_1$ is larger when $\hat{\beta}_2 = 0$ than the corresponding value in the Least Squares point due to the positive correlation between X_1 and X_2 : while X_2 is not included, the model tries to capture the common effect of X_1 and X_2 by increasing the value of $\hat{\beta}_1$.

b The path is stepwise, with decreasing step size, like:

$$\rho = 0.5$$



The component-wise boosting approach with a linear base learner updates at each iteration only one component. Therefore the path has a “step” behaviour. Based on the value of the hyperparameter “boosting step size”, these steps can be larger or smaller, but their size decreases

while approaching the Least Squares point, which is only reached in an infinite number of steps.

Problem 5

- a The bias-variance trade-off concept describes the situation for which the minimization of the prediction error must take into account to two terms, bias and variance, that act in opposite way: increasing the model complexity the bias reduces but the variance increases, and the other way around when the model complexity decreases. Finding the best model reduces to find the best point in which the two terms balance.

In gradient boosting, this is mainly associated to the number of boosting steps m_{stop} : given a boosting step size ν , increasing the number of boosting steps leads to a smaller bias but a larger variance, and the other way around if m_{stop} is reduced. The parameter ν , instead, mainly control the speed of this process: larger values of ν mean a faster bias decrease (and variance increase) for each boosting step, while lower values has the opposite effect (slower bias decrease/variable increase for each boosting step).

- b Since we want to use the negative log-likelihood from a Poisson distribution and $\lambda(x) = \exp\{f(x)\}$,

$$\begin{aligned} L(y, f(x)) &= -\log \left(\frac{\exp\{-\exp\{f(x)\}\} \exp\{f(x)\}^y}{y!} \right) \\ &= \exp\{f(x)\} - y \log(\exp\{f(x)\}) + \log(y!) \\ &= e^{f(x)} - y \cdot f(x) + \log(y!). \end{aligned}$$

The negative gradient is then

$$u(y, f(x)) = -\frac{\partial L(y, f(x))}{\partial f(x)} = -e^{f(x)} + y.$$

Problem 6

- a The suggestion of using a large number of hidden units goes together with a proper regularization approach: if the number of hidden units is too small, the Neural Network is not able to fully capture the function complexity, even without regularization. It is therefore suggested to have a large number of hidden units, that can potentially capture the complexity, and have a regularization approach to “put down” the unnecessary units. In the figure, the weight decay parameter is too

small, for this reason the Neural Networks with larger amount of hidden units overfit. the solution is to increase the value of the weight decay parameter.

- b Smaller the weights, closer the Neural Network to the linear model. Increasing values of weights allow the Neural Network to capture non-linearities, increasing the model complexity.