UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in:	STK-IN4300/9300 — Statistical Learning Methods in Data Science
Day of examination:	Tuesday, 28 November 2023
Examination hours:	15.00 – 19.00
This solution proposal consists of 5 pages.	
Appendices:	None.
Permitted aids:	None.

Solution Proposal

Problem 1

(a) For the purpose of estimating the model parameters θ:

(i) The least squares method selects the parameter values that minimizes the residual sum-of-squares (RSS):

$$\theta_{\text{ls}} = \underset{\theta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \big(y_i - f_\theta(x_i)\big)^2 \right\}$$

(ii) The maximum likelihood method selects the parameter values that maximizes the (log-)likelihood:

$$\theta_{\text{mle}} = \underset{\theta}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} \log[p(y_i \,|\, x_i; \theta, \sigma^2)] \right\},$$

where $p(Y \,|\, x; \theta, \sigma^2)$ is the conditional density function of $Y \,|\, x$.

(b) Under the given assumptions of an additive error term $N(0, \sigma^2)$, we have that

$$(Y \,|\, x) \sim N(f_\theta(x), \sigma^2),$$

and the log-likelihood can thus be written as

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{N} \left(-\frac{1}{2}\log[2\pi] - \log[\sigma] - \frac{1}{2\sigma^2}(y_i - f_\theta(x_i))^2 \right) \\ &= -\frac{N}{2}\log[2\pi] - N\log[\sigma] - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f_\theta(x_i))^2, \end{aligned}$$

where the only term involving θ is the last one, which is RSS up to a scalar negative factor. Thus, maximizing the log-likelihood w.r.t. θ is equivalent to minimizing the RSS.

(Continued on page 2.)

Problem 2

(a) Under the given assumptions, we can decompose the squared-error loss as:

$$\begin{aligned}\mathbb{E}\big[(Y-\hat{f}(x_0))^2 \,|\, x_0\big] =& \mathbb{E}\big[(f(x_0)+\epsilon-\hat{f}(x_0))^2\big] \\ =& \mathbb{E}\big[f(x_0)^2\big] + \mathbb{E}\big[f(x_0)\epsilon\big] - \mathbb{E}\big[f(x_0)\hat{f}(x_0)\big] \\ &+ \mathbb{E}\big[\epsilon f(x_0)\big] + \mathbb{E}\big[\epsilon^2\big] - \mathbb{E}\big[\epsilon\hat{f}(x_0)\big] \\ &- \mathbb{E}\big[\hat{f}(x_0)f(x_0)\big] - \mathbb{E}\big[\hat{f}(x_0)\epsilon\big] + \mathbb{E}\big[\hat{f}(x_0)^2\big].\end{aligned}$$

Now, since $f(x_0)$ is fixed (i.e. a constant) and $\mathbb{E}[\epsilon]=0$, we have that

$$\begin{aligned}&\mathbb{E}\big[f(x_0)\epsilon\big] = f(x_0)\mathbb{E}\big[\epsilon\big] = 0, \\ &\mathbb{E}\big[\epsilon^2\big] = \mathrm{Var}\big[\epsilon\big] = \sigma_\epsilon^2, \\ &\mathbb{E}\big[\hat{f}(x_0)\epsilon\big] = \mathbb{E}\big[\hat{f}(x_0)\big]\mathbb{E}\big[\epsilon\big] = 0,\end{aligned}$$

where the last equality use the fact that $\hat{f}(x_0)$ and ϵ are independent. We thus have that

$$\begin{aligned}\mathbb{E}\big[(Y-\hat{f}(x_0))^2 \,|\, x_0\big] &= f(x_0)^2 - 2f(x_0)\mathbb{E}\big[\hat{f}(x_0)\big] + \sigma_\epsilon^2 + \mathbb{E}\big[\hat{f}(x_0)^2\big] \\ &= \sigma_\epsilon^2 + \big(\mathbb{E}[\hat{f}(x_0)] - f(x_0)\big)^2 - \mathbb{E}\big[\hat{f}(x_0)\big]^2 + \mathbb{E}\big[\hat{f}(x_0)^2\big] \\ &= \sigma_\epsilon^2 + \big(\mathbb{E}[\hat{f}(x_0)] - f(x_0)\big)^2 + \mathbb{E}\big[(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)])^2\big],\end{aligned}$$

where the last step makes use of the variance formula:

$$\mathrm{Var}\big[\hat{f}(x_0)\big] = \mathbb{E}\big[\hat{f}(x_0)^2\big] - \mathbb{E}\big[\hat{f}(x_0)\big]^2.$$

The first term is the variance of target around its true mean $f(x_0)$ (irreducible error), the second term represents the systematic error, that is, how much the average of our estimate differs from the true mean (squared bias), and the third term is the variance of our estimator, that is, how much it varies around its mean.

(b) Here, k denotes the number of neighbors and it is inversely related to model complexity. From small values of k, the model is more flexible/complex and can better adapt to the true function $f(x)$. For larger values of k, the model will become less flexible/complex, since we will compute the average based on training points further away from x_0. In terms of the bias-variance tradeoff, small values of k will give a model with low bias but high variance and large values will give a model with high bias but low variance.

Problem 3

(a) The general idea behind the considered shrinkage methods is to add a regularization term to the least squares objective that will shrink the estimates towards zero in relation to the least-squares solution. They key idea is that by adding bias to the model through shrinkage, one is able to reduce the variance and thus (possibly) decrease the

(Continued on page 3.)

expected prediction error. The amount of shrinkage is controlled by a regularization parameter, often denoted by λ, which in this case is directly (inversely) related to t in the given optimization problem.

(b) (i) The key difference between ridge regression and lasso, in terms of the model they produce, is that lasso encourage sparse solutions by shrinking some parameters all they way to zero (built-in feature selection). This property is reflected in the constraint region of lasso in Figure 1 (on the left), in the sense that one will typically hit one the "sharp corners" of the region, which represent solutions where one of the parameters is set to zero. This is in contrast to the constraint region of ridge regression (on the right) where there is no "sharp corners" and one will typically not have any zero parameter estimates.

(ii) By increasing t (corresponds to decreasing λ), the constraint regions in the figure will increase in size and the parameters will thus be shrunken less (and vice versa). If we increase t enough, the constraint region will cover the least-squares solution and we will not get any shrinkage.

Problem 4

(a) The problem with the described procedure is that initial screening in Step 1 is done prior to cross-validation, that is, it is using all the data. In other words, when training a model in Step 2 it has already used information from all data (including the fold that is used for testing). As a result, the averaged test error will be overly optimistic as an estimate of the expected test error. One way to improve the proposed procedure would be to move the screening step inside the cross-validation loop, such that the screening would only be based on the training data w.r.t. the current fold.

(b) **(i)** This is the generalization error or expected test error given the training set. In terms of model evaluation, this is typically what we are interested in for a given application where we have some data on which to train the model, that is, how well will the model trained on the available data perform in predicting new observations.

(ii) This is the expected test error when we do not fix the training data, but instead compute the expectation also w.r.t. the training data. This error is typically more convenient to work with from a statistical analysis point of view.

(iii) This is the in-sample error where we keep the input values and compute the expected error assuming that only the output values are redrawn. The in-sample error is overly optimistic in terms of estimating the expected test error in the sense that the test and training inputs coincide. Estimates of the in-sample error can still be very useful for model selection. For example, the AIC and BIC are popular model selection criteria that can be viewed as estimates of the in-sample error.

(Continued on page 4.)

Problem 5

(a) Bootstrap aggregating (or bagging) is a technique where: (i) B bootstrap samples are drawn by resampling the training data with replacement; (ii) a tree is fitted on each bootstrap sample; (iii) the predictions of the bootstrap trees are averaged to form the final prediction. Bagging is a variance reduction technique and can thus be particularly useful for improving the results of high-variance (and low-bias) methods, such as trees.

(b) Since the trees are fitted on bootstrap samples from the same distribution, the predictions of the individual trees are typically highly correlated. Thus, while the second term on the right-hand side of the equation will disappear as B is increased, the first term will not be affected. This implies that the correlation between bootstrap trees will limit the benefits of bagging. The random forest improves variance reduction by reducing the correlation between trees. This is achieved by, at each splitting step during the tree growing process, randomly selecting a subset of candidate features to split on.

(c) The total variance is the variance of the prediction of a tree at point x, which includes the sampling variability of the training data $\boldsymbol{Z}$ as well as the variability due to the random selection of split candidates. The first term on the right-hand side is the sampling variability of the random forest ensemble. The second term is the within-$\boldsymbol{Z}$ variance, which is the expected variance due to the randomization step when growing the trees.

Problem 6

(a) We compute the derivative of the objective function w.r.t. $f(x)$ and set it to zero:

$$\begin{aligned}\frac{d}{d\,f(x)}\mathbb{E}_{Y\,|\,x}[e^{Yf(x)}] &= \frac{d}{d\,f(x)}\left(e^{-f(x)}p(Y=1\,|\,x)+e^{f(x)}p(Y=-1\,|\,x)\right)\\ &= -e^{-f(x)}p(Y=1\,|\,x)+e^{f(x)}p(Y=-1\,|\,x)\\ &\overset{\text{set}}{=} 0.\end{aligned}$$

Then, we solve for $f(x)$:

$$e^{2f(x)} = \frac{p(Y=1\,|\,x)}{p(Y=-1\,|\,x)} \Rightarrow f(x) = \frac{1}{2}\log\left[\frac{p(Y=1\,|\,x)}{p(Y=-1\,|\,x)}\right],$$

and we know that solution is a minimum since $\frac{d^2}{d[f(x)]^2}\mathbb{E}_{Y\,|\,x}[e^{Yf(x)}] > 0$.

Adaboost classifies based on the sign of the additive expansion it produces (Step 3 in Alg. 10.1). This makes sense given the above population minimizer, since it corresponds to predicting the class label with the highest conditional probability (Bayes classifier).

(b) Under the considered setting, the classification rule implies that observations with a positive margin, $y_i f(x_i) > 0$, are classified

(Continued on page 5.)

correctly and observations with a negative margin, $y_i f(x_i) < 0$, are misclassified (and the decision boundary is defined by $f(x_i) = 0$). Thus, a reasonable loss criterion should penalize negative margins more heavily than positive margins. This is not case for the squared error loss, making it a bad choice in the considered setting. Both the exponential loss and binomial deviance are strictly decreasing functions of the margin, penalizing increasingly negative margins more than they reward increasingly positive ones. The exponential loss increases very quickly (exponentially) as a negative margin value is further decreased, putting a lot of focus on observations with large negative margins. In comparison, the binomial deviance increases more moderately (almost linearly) as a negative margin value is further decreased, and it is thus less influenced by observations with large negative margins. As a result, the binomial deviance is more robust in noisy settings (with high Bayes error rate), and especially when there are misspecification of class labels in the training data.