

UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in: STK2100 — Machine Learning and Statistical
Methods for Prediction and Classification
SKETCH OF SOLUTION

Day of examination: Friday June 12th, 2026

This problem set consists of 7 pages.

Appendices: None

Permitted aids: Any

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Problem 1

a

- The P-values from the Wald tests on the individual covariates indicate that all the covariates are important, except `fineagg`, that does not have a significant effect.
- The estimated coefficient for `cement` is 0.095, which means that the expected compression strength of concrete increases by 0.095MPa when the cement content of the concrete increases by 1kg/m³ and all other covariates are the same.
- The categorical covariate `age.gr` has three levels, and age group 1 is used as a reference. The estimated coefficient for age group 2 is 16.9, which means that the expected compression strength of concrete increases by 16.9MPa when the age of the concrete specimen belongs to age group 2, compared to age group 1, and all other covariates are the same. Similarly, the expected compression strength of concrete increases by 18.5MPa when the age of the concrete specimen belongs to age group 3, compared to age group 1, and all other covariates are the same.

b

- Let $l(\hat{\theta})$ be the log-likelihood of the model that is considered, with $\hat{\theta}$ the maximum likelihood estimate of the model parameters θ . Then

$$AIC = -2l(\hat{\theta}) + 2|\hat{\theta}| \quad \text{and} \quad BIC = -2l(\hat{\theta}) + \log(N)|\hat{\theta}|,$$

where $|\hat{\theta}|$ is the number of free parameters and N is the size of training set. The BIC is consistently higher than the AIC because it puts a higher penalty on the number of free parameters.

- Lower values of either AIC or BIC indicates a better model. The AIC is the same for the two models, and the BIC is lower for the model without

(Continued on page 2.)

fineagg. This speaks in favour of choosing Model 2. It also makes sense to remove **fineagg** because it did not have a significant effect.

c

- The expected concrete strength decreases with the water content when this is between 140 and 210kg/m³. For water contents below 140kg/m³, the strength seems to increase slightly, and it also seems to increase for water contents above 210kg/m³. However, the uncertainty is very large in these areas, indicated by few observations and wide confidence bands. Further, the expected concrete strength increases for contents of fine aggregate up to $\sim 680\text{kg/m}^3$, then is rather constant for contents between $\sim 680\text{kg/m}^3$ and $\sim 800\text{kg/m}^3$, and finally decreases for contents above $\sim 800\text{kg/m}^3$.
- The hypothesis test of Model 3 against a corresponding model without **fineagg** resulted in a very low P-value, which means that **fineagg** has a significant effect in the additive model. We also see from the test of the contribution of the non-linear part of the effect of **fineagg** that this effect is significantly non-linear (this is seen in the table "Anova for Nonparametric Effects"). This is probably the reason why **fineagg** has a significant effect in the additive model, but not in the linear model from **a**.
- According to the tests of the contribution of the non-linear parts of the effects modelled with splines, these are all significant compared to a simple linear term (this is seen in the table "Anova for Nonparametric Effects"). Further, both the AIC and BIC are significantly lower in the additive model, compared to the linear model from **a** with the same covariates, despite a higher number of degrees of freedom. The estimated effects also look quite non-linear, especially for **water**, **superplastic** and **fineagg**. This all suggests that there are non-linear effects of the covariates.

d

- The most important covariates in Model 4 are **age.gr** (but only the difference between age group 1 and the two others), **superplastic**, **cement**, **fineagg**, **slag**, **ash** and **water**, which are shared with Model 3, and except for **fineagg** with Models 1 and 2. However, the difference between age groups 2 and 3 are significant in Models 1-3, but not in Model 4.
- We know that the prediction from Model 1 for Case 3 is 58.3, and the only difference in Case 4 is that **superplastic**=8 instead of 0.7, where the estimated effect of **superplastic** is 0.470. Hence, the prediction for Case 4 must be $58.3 + (8 - 0.7) \cdot 0.470 \approx 61.7^1$. For Model 2, the prediction for Case 2 is 44.1, and the only difference in Case 1 is that **superplastic**=0.7 instead of 8, where the estimated effect of **superplastic** is 0.428. Hence, the prediction for Case 1 must be $44.1 + (0.7 - 8) \cdot 0.428 \approx 41.0$. Finally for model 4 we have:

- Case 2: follow the path "**age.gr**: 1" go left, then "**cement** < 257.7" go right, at "**superplastic** < 0.95" go left, at "**cement** < 389" go right and at "**water** < 179" go left, giving the prediction 60.6.

¹There was an error in the original exam problem, stating that the prediction for Case 3 was 58.3.

- Case 3: follow the path "age.gr: 1" go right, at "superplastic < 7.85" go left and at "fineagg < 747.65" go left, giving the prediction 50.1.
- Case 4: follow the path "age.gr: 1" go right, at "superplastic < 7.85" go right and at "cement < 305.25" go right, giving the prediction 69.4.

The table with all the predictions is then

Setting	Model 1	Model 2	Model 3	Model 4
Case 1	41.4	41.0	40.3	60.6
Case 2	44.9	44.1	48.2	49.8
Case 3	59.9	59.6	59.0	50.1
Case 4	63.3	62.7	66.8	69.4
MSE	94.4	94.2	76.8	104.9

- As opposed to Models 1-3, Model 4 is able to capture interaction effects. Since the MSE on the test set is much higher for Model 4 than for the other models, indicating a poorer prediction performance, there is no reason to believe that there are important interaction effects based on these results.

e

- A boosted tree model consists of a large collection of trees, all of which are of the same type as the regression tree in Model 4, but which are much smaller, and hence weak learners, that have poor prediction performance on their own. Further, the regression tree is fitted to the data as they are, whereas the individual trees of the boosted tree model are fitted in a stagewise manner to different versions of the data. More specifically, in each stage, the model fitted so far is kept fixed, and a single tree is added and fitted to the residuals from the model fitted so far. The way in which the small tree is fitted to the residuals is however the same as the way that the regression tree is fitted to the original data.

- The depth of the individual trees of the boosted tree model gives the maximum order of interaction effects. Since trees with depth 1 gave the best fit, this is again an indication that there are no important interaction effects.

- According to the mean squared errors on the test set, the boosting model, Model 5, is the best, closely followed by Model 3, probably because they are able to capture non-linearities. The linear models, Models 1 and 2, are next, and have very similar performance. It is therefore natural to choose the simplest model, i.e. Model 2, over Model 1. The tree model, Model 4, has by far the worst performance, maybe because its main strength, being able to capture interaction effects, does not seem to be important for the Concrete data.

Problem 2

a

- Since there are no constraints on the optimisation when fitting the logistic regression model without regularisation, the number of degrees of freedom is

(Continued on page 4.)

the same as the number of model parameters, which is 26, i.e. the number of beta coefficients. With regularisation, the number of degrees of freedom is reduced as all the parameters are no longer allowed to vary freely.

- The number of degrees of freedom of the logistic regression model is large compared to the size of the training set, and none of the fitted effects are significant, only a few are borderline significant. This is a good reason to try regularisation, to shrink the model parameters towards 0.
- We get ridge regularisation with $q = 2$ and Lasso with $q = 1$.
- Both ridge and Lasso type regularisation shrink the model parameters towards 0, but with Lasso some of the coefficients can be shrunk to exactly 0, whereas with ridge, they can come arbitrarily close to 0, but will never be set exactly to 0. This means that Lasso does both shrinkage and variable selection.

b

The procedure for doing selecting the regularisation parameter λ with K -fold cross-validation could be as follows.

- Divide the data randomly into K folds, and define a sequence $\lambda_1, \dots, \lambda_d$ of potential values for λ .
- For each λ_l , $l = 1, \dots, d$, one must fit the regularised logistic regression model with that value of λ to all the data except the observations in the current fold.
- Each fitted model is then used to predict the observations that were not used in the fitting, and the corresponding squared error loss is then computed
- This is repeated for all the fold, and the loss is averaged over all the folds.
- The λ_l that gives the lowest average loss over the folds is chosen.

c

- For a given threshold $\tau \in (0, 1)$, and the predicted probabilities $\hat{p}_i$, $i = 1, \dots, N_{te}$, for the test set, assume that we classify as

$$\hat{y}_i = \begin{cases} 1, & \hat{p}_i > \tau \\ 0, & \hat{p}_i \leq \tau \end{cases}.$$

Then, the true positive rate (tpr) is the fraction of the true positives $y_i = 1$ that are classified as positive, i.e. $\hat{y}_i = 1$ and the false positive rate (fpr) is the fraction of the true negatives $y_i = 0$ that are classified as positive, i.e. $\hat{y}_i = 1$. Hence,

$$\begin{aligned} \text{tpr} &= \frac{\sum_{i=1}^{N_{te}} I(\hat{y}_i = 1, y_i = 1)}{\sum_{i=1}^{N_{te}} I(y_i = 1)} \\ \text{fpr} &= \frac{\sum_{i=1}^{N_{te}} I(\hat{y}_i = 1, y_i = 0)}{\sum_{i=1}^{N_{te}} I(y_i = 0)}. \end{aligned}$$

(Continued on page 5.)

For the given τ , we get the pair (tpr,fpr). The receiver operating characteristic (ROC) curve is obtained by computing this pair for the full range of values of τ , and then plotting the tprs against the fprs.

- The area under the curve (AUC), is the area defined by the ROC curve, and the lower right corner of the plot. A perfect classifier has an AUC of 1 and random guessing gives an AUC of 0.5.
- The ridge model has the highest AUC on the test set, and is therefore the best performing according to this measure. The second best is the Lasso model and the third the logistic regression model without regularisation. However, one should note that the AUC of all three models is quite high, indicating quite poor prediction performance.

Problem 3

a

- We have:

$$\begin{aligned}
 & (\mathbf{y} - \mathbf{B}\boldsymbol{\theta}_x)^T \mathbf{W}_x (\mathbf{y} - \mathbf{B}\boldsymbol{\theta}_x) \\
 &= (y_1 - \mathbf{b}_1^T \boldsymbol{\theta}_x \quad \dots \quad y_N - \mathbf{b}_N^T \boldsymbol{\theta}_x) \begin{pmatrix} w(\mathbf{x}, \mathbf{x}_1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & w(\mathbf{x}, \mathbf{x}_N) \end{pmatrix} \begin{pmatrix} y_1 - \mathbf{b}_1^T \boldsymbol{\theta}_x \\ \vdots \\ y_N - \mathbf{b}_N^T \boldsymbol{\theta}_x \end{pmatrix} \\
 &= (y_1 - \mathbf{b}_1^T \boldsymbol{\theta}_x \quad \dots \quad y_N - \mathbf{b}_N^T \boldsymbol{\theta}_x) \begin{pmatrix} w(\mathbf{x}, \mathbf{x}_1)(y_1 - \mathbf{b}_1^T \boldsymbol{\theta}_x) \\ \vdots \\ w(\mathbf{x}, \mathbf{x}_N)(y_N - \mathbf{b}_N^T \boldsymbol{\theta}_x) \end{pmatrix} \\
 &= \sum_{i=1}^N w(\mathbf{x}, \mathbf{x}_i) (y_i - \mathbf{b}_i^T \boldsymbol{\theta}_x)^2 = R(\boldsymbol{\theta}_x).
 \end{aligned}$$

Further, we can write

$$R(\boldsymbol{\theta}_x) = \mathbf{y}^T \mathbf{W}_x \mathbf{y} - 2\mathbf{y}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x + \boldsymbol{\theta}_x^T \mathbf{B}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x.$$

Using the rules of differentiation, we obtain

$$\begin{aligned}
 \frac{\partial \mathbf{y}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x}{\partial \boldsymbol{\theta}_x} &= \mathbf{B}^T \mathbf{W}_x \mathbf{y} \\
 \frac{\partial \boldsymbol{\theta}_x^T \mathbf{B}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x}{\partial \boldsymbol{\theta}_x} &= (\mathbf{B}^T \mathbf{W}_x \mathbf{B} + (\mathbf{B}^T \mathbf{W}_x \mathbf{B})^T) \boldsymbol{\theta}_x = 2\mathbf{B}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x,
 \end{aligned}$$

so that

$$\frac{\partial R(\boldsymbol{\theta}_x)}{\partial \boldsymbol{\theta}_x} = -2\mathbf{B}^T \mathbf{W}_x \mathbf{y} + 2\mathbf{B}^T \mathbf{W}_x \mathbf{B} \boldsymbol{\theta}_x.$$

We obtain the estimate $\hat{\boldsymbol{\theta}}_x$ by setting $\frac{\partial R(\boldsymbol{\theta}_x)}{\partial \boldsymbol{\theta}_x}$ to $\mathbf{0}$, and solving for $\partial \boldsymbol{\theta}_x$:

$$\begin{aligned}
 -2\mathbf{B}^T \mathbf{W}_x \mathbf{y} + 2\mathbf{B}^T \mathbf{W}_x \mathbf{B} \hat{\boldsymbol{\theta}}_x &= \mathbf{0} \\
 \rightarrow \mathbf{B}^T \mathbf{W}_x \mathbf{B} \hat{\boldsymbol{\theta}}_x &= \mathbf{B}^T \mathbf{W}_x \mathbf{y} \\
 \rightarrow \hat{\boldsymbol{\theta}}_x &= (\mathbf{B}^T \mathbf{W}_x \mathbf{B})^{-1} \mathbf{B}^T \mathbf{W}_x \mathbf{y}.
 \end{aligned}$$

(Continued on page 6.)

b

- We have:

$$\begin{aligned}
\mathbf{B}^T \mathbf{W}_x \mathbf{B} &= \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} K_\lambda(x, x_1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & K_\lambda(x, x_N) \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} K_\lambda(x, x_1) \\ \vdots \\ K_\lambda(x, x_N) \end{pmatrix} = \sum_{i=1}^N K_\lambda(x, x_i),
\end{aligned}$$

so that $(\mathbf{B}^T \mathbf{W}_x \mathbf{B})^{-1} = 1 / \sum_{i=1}^N K_\lambda(x, x_i)$, and

$$\begin{aligned}
\mathbf{B}^T \mathbf{W}_x \mathbf{y} &= \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} K_\lambda(x, x_1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & K_\lambda(x, x_N) \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \\
&= \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} K_\lambda(x, x_1)y_1 \\ \vdots \\ K_\lambda(x, x_N)y_N \end{pmatrix} = \sum_{i=1}^N K_\lambda(x, x_i)y_i.
\end{aligned}$$

This gives

$$\hat{\boldsymbol{\theta}}_{\mathbf{x}} = (\mathbf{B}^T \mathbf{W}_x \mathbf{B})^{-1} \mathbf{B}^T \mathbf{W}_x \mathbf{y} = \frac{\sum_{i=1}^N K_\lambda(x, x_i)y_i}{\sum_{i=1}^N K_\lambda(x, x_i)}.$$

Finally, we have

$$\hat{f}(x) = \mathbf{b}^T \hat{\boldsymbol{\theta}}_x = 1 \cdot \hat{\boldsymbol{\theta}}_x = \frac{\sum_{i=1}^N K_\lambda(x, x_i)y_i}{\sum_{i=1}^N K_\lambda(x, x_i)},$$

i.e. the kernel smoother.

- With $\mathbf{b} = 1$ and $w(x, x_i) = I(x_i \in N_k(x))$, following the same steps as above, we obtain

$$\hat{\boldsymbol{\theta}}_{\mathbf{x}} = \frac{\sum_{i=1}^N I(x_i \in N_k(x))y_i}{\sum_{i=1}^N I(x_i \in N_k(x))}$$

so that

$$\hat{f}(x) = \mathbf{b}^T \hat{\boldsymbol{\theta}}_x = 1 \cdot \hat{\boldsymbol{\theta}}_x = \frac{\sum_{i=1}^N I(x_i \in N_k(x))y_i}{\sum_{i=1}^N I(x_i \in N_k(x))},$$

i.e. k -nearest neighbour.

(Continued on page 7.)

c

In higher dimensions, there are few, if any observations that are close to a given point $\mathbf{x}$ of interest. This is called the curse of dimensionality, and means for k -nearest neighbour, that also observations quite far away from $\mathbf{x}$ will be used in the estimation of $\hat{f}(x)$. For kernel based models like kernel smoothers and local linear regression, one has to set the λ parameter quite large, to be sure that there are observations that are close enough to $\mathbf{x}$ to contribute to $\hat{f}(x)$. The consequence is that the bias becomes large, in addition to the variance, thus counteracting the whole point of a model that can adapt locally.