Statistical Machine Learning
Solutions for Exam 2020-03-17

1. i. **True**
   ii. **False**
   iii. **True**
   iv. **False**. It is not true in general. However, it is true if $\widehat{\pi}_1 = \widehat{\pi}_2 = \frac{1}{2}$.
   v. **False**, consider e.g. k-nearest-neighbors with $k = 10$.
   vi. **True**
   vii. **False**
   viii. **False**
   ix. **False**, boosting is typically used with simple base models with high bias and low variance.
   x. **False**

2. (a) The least squares estimate is given boundary

$$(X^\mathsf{T}X)^{-1}X^\mathsf{T}y,$$

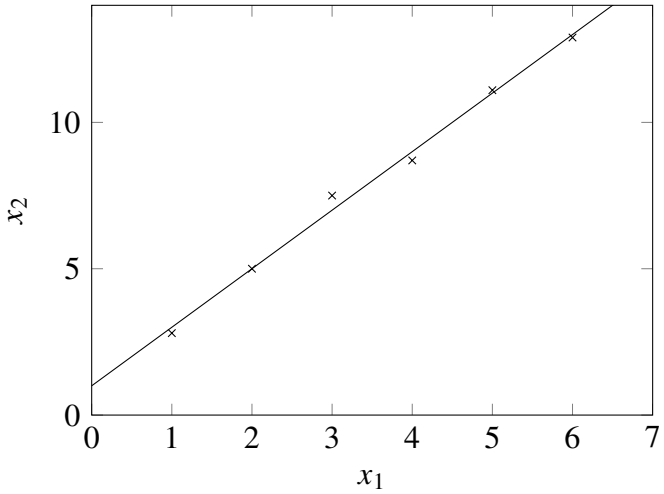
where

$$X^\mathsf{T}X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix}$$

$$X^\mathsf{T}y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 2.8 \\ 5.0 \\ 7.5 \\ 8.7 \\ 11.1 \\ 12.9 \end{bmatrix} = \begin{bmatrix} 48 \\ 203 \end{bmatrix}$$

We get

$$\widehat{\boldsymbol{\theta}} = \left(X^\mathsf{T}X\right)^{-1} X^\mathsf{T}y = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix}^{-1} \begin{bmatrix} 48 \\ 203 \end{bmatrix} = \frac{1}{6 \cdot 91 - 21^2} \begin{bmatrix} 91 & -21 \\ -21 & 6 \end{bmatrix} \begin{bmatrix} 48 \\ 203 \end{bmatrix}$$
$$= \frac{1}{105} \begin{bmatrix} 105 \\ 210 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$



(b) We will reach zero training error since a 5:th order polynomial has 6 parameters and we have 6 observations.

(c) LASSO gives sparse solutions. Ridge regression shrinks the parameter values towards the origin. Hence, we get the pairing (a)-LASSO, (b)-Least squares, (c)-Ridge regression.

(d) We no longer have 6 unique points. Since we have fewer observations than parameters, the least squares problem does not have a unique solution but rather infinitely many. Most software chooses the minimum norm solution. That is, the $\boldsymbol{\theta}$ with the smallest norm $\|\boldsymbol{\theta}\|_2$ that is still a solution.

(e) The prediction is the average of the prediction from each individual model.

$$\widehat{y}_{\text{bag}}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}\widehat{y}^{(b)}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}\widehat{\boldsymbol{\theta}}_{(b)}^{\mathsf{T}}\boldsymbol{x} = \left[\frac{1}{B}\sum_{b=1}^{B}\widehat{\boldsymbol{\theta}}_{(b)}\right]^{\mathsf{T}}\boldsymbol{x}$$

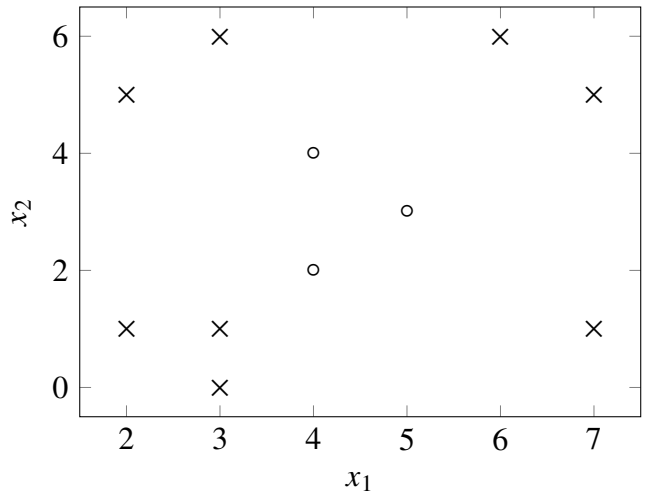
That is,

$$\widehat{\boldsymbol{\theta}}_{\text{bag}} = \frac{1}{3}\begin{bmatrix} -29.8 - 1.09 - 2.25 \\ 56.4 + 5.15 + 4.73 \\ -28.1 - 1.46 + 1.39 \\ 4.31 + 0.187 - 1.36 \\ 0.0894 + 010141 + 0.31 \\ -0.0425 - 0.00296 - 0.0221 \end{bmatrix} \approx \begin{bmatrix} -11 \\ 22.1 \\ -9.39 \\ 1.05 \\ 0.138 \\ -0.0225 \end{bmatrix}$$

To predict $x = 2$ we need first to compute feature vector $\boldsymbol{x} = [1, x, x^2, x^3, x^4, x^5] = [1, 2, 4, 8, 16, 32]$. Finally, we get

$$\widehat{y}_{\text{bag}}(\boldsymbol{x}) = \begin{bmatrix} -11 & 22.1 & -9.39 & 1.05 & 0.138 & -0.0225 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \\ 16 \\ 32 \end{bmatrix} = 5.43$$

3. (a) Se the figure

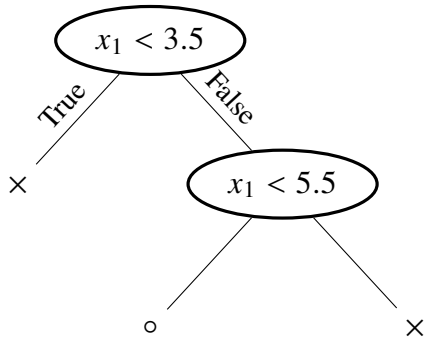


(b) Directly from the graph, we see that we have two possible splits along the x_1 and two along x_2.

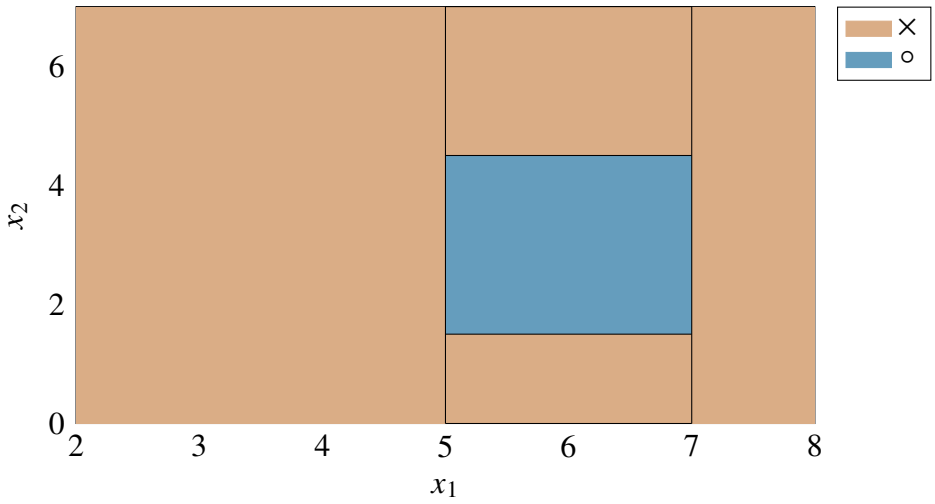
	n_1	$\circ$	$\times$	Q_1	n_2	$\circ$	$\times$	Q_2	$n_1Q_1 + n_2Q_2$
$x_1 < 3.5$	5	0	5	0	6	3	3	$\frac{1}{2}$	3
$x_1 < 5.5$	8	3	5	$\frac{30}{64}$	3	0	3	0	$\frac{30}{8} > 3$
$x_2 < 1.5$	4	0	4	0	7	3	4	$\frac{24}{49}$	$\frac{24}{7} > 3$
$x_2 < 4.5$	7	3	4	$\frac{24}{49}$	4	0	4	0	$\frac{24}{7} > 3$

For the next split, we have three possible splits one and only one of them separates the two classes completely. Hence, the optimal split is $x_1 < 5.5$.

We can now draw the tree.



(c) See the regions in the graph below.

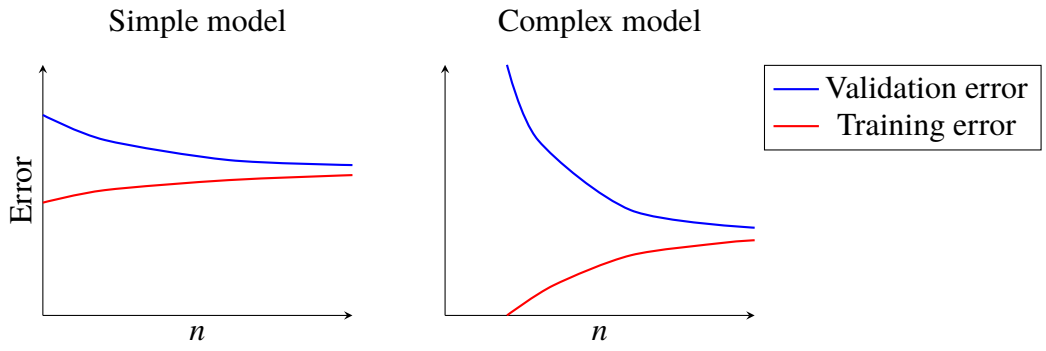


(d) The dataset is not linearly separable. Hence, a linear classifier like logistic regression will not work. Both kNN and QDA can separate the data completely.

4. (a) **Small** n: A complex model can often fit the training data exactly and gets zero training error. This leads to large validation error. A simple model can not achieve zero training error for n moderately larger than 0. Since there is less overfit, the validation error will not be much larger than the training error.

Large n: The training error increases with n for both the complex and the simple model. The complex model will overfit less and the validation error will decrease. The same will happen with the simple model.

The errors will tend towards each other but not meet. The errors errors wil be smaller for the complex model.



(b) Cross-validation is a technique to indicate how well a model generalizes to new data. The data are partitioned in two sets, a training set and a validation set. The training set is used to estimate a model and its performance is evaluated on the validation set. Typically, multiple rounds are performed using different partitions and the results are averaged.

Describing k-fold cross-validation also gives full points.

(c) The error shown is an estimate of the new data error. The lowest value corresponds to $k \approx 23$.

(d) Note that for k-NN, the complexity decreases with increasing k. In the figure, bias is increasing with k, hance the increase in error for large k. At the same time, variance increases with decreasing k. This accounts for the increased error for small values of k. The minima corresponds to a balanced fit where both the bias and the variance are small.

5. (a) The range of any probability is $[0, 1]$.

   (b) The range of the odds is $[0, \infty)]$. Taking the logarithm transforms the range to $(-\infty, \infty)$.

   (c) Modelling $p(y = 1 \mid x) = g(x)$ means that $p(y = -1 \mid x) = 1 - g(x)$. Inserting this in the log odds, we get

$$\begin{aligned}
\ln \frac{g(x)}{1 - g(x)} &= \theta^\mathsf{T} x && \iff \\
\frac{g(x)}{1 - g(x)} &= e^{\theta^\mathsf{T} x} && \iff \\
g(x)\left(1 + e^{\theta^\mathsf{T} x}\right) &= e^{\theta^\mathsf{T} x} && \iff \\
g(x) &= \frac{e^{\theta^\mathsf{T} x}}{1 + e^{\theta^\mathsf{T} x}}
\end{aligned}$$

   which is the logistic regression model.

   (d) Logistic regression is an example of generalized linear models, where the output is transformed to be estimated by linear regression. In this case, we are interested in binary classification and the probability of the positive class. This demands us to map the range of the probability $[0, 1]$ to $(-\infty, \infty)$ This is fulfilled by all three functions. In general, the inverse of a CDF will always work.

      i. Called the *probit link function*. It is obviously the inverse of a CDF.

      ii. Sometimes called the *cauchit link function*. It is the inverse CDF of a Cauchy distribution

      iii. Called the complementary *log-log link function* or the *gompit link*. It is the inverse of the Gompertz curve, $f(t) = ae^{-be^{-ct}}$ with $a = b = c = 1$. The Gompit link function is not an inverse CDF.
      Note: There was a mistake in the original exam where it said $\ln(1 - \ln(1 - \mu))$. *This is not a suitable link function.*