PROPOSED SOLUTIONS FOR EXAM IN STATISTICAL MACHINE LEARNING STATISTISK MASKININLÄRNING

DATE AND TIME: January 10, 2023

1. i. False
   ii. False
   iii. True
   iv. False
   v. True
   vi. False
   vii. False
   viii. False
   ix. False
   x. True

2. (a) The average squared-error loss of linear model:

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i;\boldsymbol{\theta}))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\theta})^2 = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 6 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 4 \\ 7 \\ 11 \end{bmatrix}$$

A minimizing solution $\boldsymbol{\theta}$ satisfies the normal equations and when $(\mathbf{X}^\mathsf{T}\mathbf{X})$ is invertible, we have a unique solution:

$$\widehat{\boldsymbol{\theta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \begin{bmatrix} 3 & 12 \\ 12 & 56 \end{bmatrix}^{-1}\begin{bmatrix} 22 \\ 102 \end{bmatrix} = \frac{1}{24}\begin{bmatrix} 56 & -12 \\ -12 & 3 \end{bmatrix}\begin{bmatrix} 22 \\ 102 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{7}{4} \end{bmatrix}$$

(b) The expected new squared error can be expanded into:

$$\begin{aligned}
\mathbb{E}[(y - f(\mathbf{x};\widehat{\boldsymbol{\theta}}))^2] &= \mathbb{E}[(\alpha x_1 + \varepsilon - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}})^2] \\
&= \mathbb{E}[(\alpha x_1 - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}})^2 + 2\varepsilon(\alpha x_1 - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}}) + \varepsilon^2] \\
&= \mathbb{E}[(\alpha x_1 - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}})^2] + 2\mathbb{E}[\varepsilon]\mathbb{E}[(\alpha x_1 - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}})] + \mathbb{E}[\varepsilon^2] \\
&= \mathbb{E}[(\alpha x_1 - \mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}})^2] + 0 + \sigma^2
\end{aligned}$$

where the third equality uses the independence between ε and $\mathbf{x}$ and the last equality uses the properties of ε. Finally, we use $\mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\theta}} = \widehat{\theta}_0 + \widehat{\theta}_1 x_1$ and insert it into the expression above to obtain:

$$\begin{aligned}
\mathbb{E}[(y - f(\mathbf{x};\widehat{\boldsymbol{\theta}}))^2] &= \mathbb{E}[((\alpha - \widehat{\theta}_1)x_1 - \widehat{\theta}_0)^2] + \sigma^2 \\
&= \mathbb{E}[(\alpha - \widehat{\theta}_1)^2 x_1^2 - 2(\alpha - \widehat{\theta}_1)x_1\widehat{\theta}_0 + \widehat{\theta}_0^2] + \sigma^2 \\
&= (\alpha - \widehat{\theta}_1)^2\mathbb{E}[x_1^2] - 2(\alpha - \widehat{\theta}_1)\mathbb{E}[x_1]\widehat{\theta}_0 + \widehat{\theta}_0^2 + \sigma^2 \\
&= (\alpha - \widehat{\theta}_1)^2\mathbb{E}[x_1^2] + 0 + \widehat{\theta}_0^2 + \sigma^2
\end{aligned}$$

where the last equality follows from x_1 having zero mean.

Interpretation: The second term is the irreducible noise variance that is inherent to the problem problem. The first term is a reducible error component, which in this case would be minimized by $\widehat{\theta}_1 = \alpha$ and $\widehat{\theta}_0 = 0$.

(c) By adding a new input, we get a new input data matrix:

$$\widetilde{\mathbf{X}} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 4 & 4 \\ 1 & 6 & 5 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 4 \\ 7 \\ 11 \end{bmatrix}$$

and as before a minimizing solution $\widetilde{\boldsymbol{\theta}}$ satisfies the normal equations. However, the columns of $\widetilde{\mathbf{X}}$ are in this case linearly dependent and thus $(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}})$ is not invertible! Therefore there is no unique solution, but an infinite set of least-squares models with parameters that satisfy the normal equations.

3. a) $p(y = 1|\mathbf{x}; \boldsymbol{\theta}) = h(\boldsymbol{\theta}^\top \mathbf{x}) = \frac{\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}{1+\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}$. Therefore, $p(y = -1|\mathbf{x}; \boldsymbol{\theta}) = 1 - p(y = 1|\mathbf{x}; \boldsymbol{\theta}) = 1 - \frac{\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}{1+\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}} = \frac{1}{1+\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}} = \frac{\mathbf{e}^{-\boldsymbol{\theta}^\top \mathbf{x}}}{1+\mathbf{e}^{-\boldsymbol{\theta}^\top \mathbf{x}}}$.

This means that $p(y|\mathbf{x}; \boldsymbol{\theta})$ can be written as,

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \begin{cases} \frac{\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}{1+\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}, & y = 1 \\ \frac{\mathbf{e}^{-\boldsymbol{\theta}^\top \mathbf{x}}}{1+\mathbf{e}^{-\boldsymbol{\theta}^\top \mathbf{x}}}, & y = -1 \end{cases} = \frac{\mathbf{e}^{y\boldsymbol{\theta}^\top \mathbf{x}}}{1 + \mathbf{e}^{y\boldsymbol{\theta}^\top \mathbf{x}}}.$$

The maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{ML}}$ is then given by,

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{ML}} &= \arg\max_{\boldsymbol{\theta}} \ln \ell(\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln p(y_i|\mathbf{x}_i; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln \frac{\mathbf{e}^{y_i \boldsymbol{\theta}^\top \mathbf{x}_i}}{1 + \mathbf{e}^{y_i \boldsymbol{\theta}^\top \mathbf{x}_i}} \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln \frac{1}{1 + \mathbf{e}^{-y_i \boldsymbol{\theta}^\top \mathbf{x}_i}} \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} -\ln(1 + \mathbf{e}^{-y_i \boldsymbol{\theta}^\top \mathbf{x}_i}) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln(1 + \mathbf{e}^{-y_i \boldsymbol{\theta}^\top \mathbf{x}_i}). \end{aligned}$$

b) We know that the decision boundary is located where the function for the positive class $p(y = 1|\mathbf{x}; \boldsymbol{\theta})$ and the function for the negative class $p(y = -1|\mathbf{x}; \boldsymbol{\theta})$ intersect (where the two classes are predicted to be equally probable). The decision boundary can thus be computed by solving the equation,

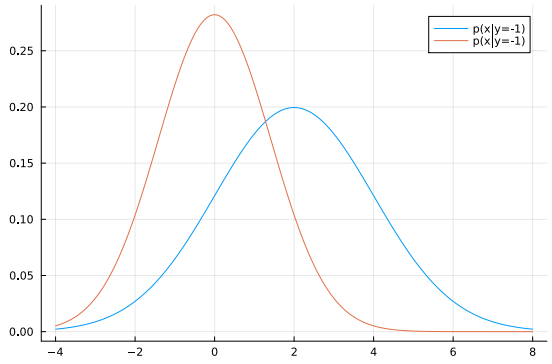
$$\begin{aligned} p(y = 1|\mathbf{x}; \boldsymbol{\theta}) &= p(y = -1|\mathbf{x}; \boldsymbol{\theta}) \\ \frac{\mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}}{1 + \mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}} &= \frac{1}{1 + \mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}}} \\ \mathbf{e}^{\boldsymbol{\theta}^\top \mathbf{x}} &= 1 \\ \boldsymbol{\theta}^\top \mathbf{x} &= 0, \end{aligned}$$

which describes a linear hyperplane in $\mathbf{x}$.

From this it becomes clear that if $\boldsymbol{\theta}^\top \mathbf{x} < 0$ we will predict the negative class ($\hat{y}(\mathbf{x}) = -1$), and if $\boldsymbol{\theta}^\top \mathbf{x} > 0$ we will predict the positive class ($\hat{y}(\mathbf{x}) = 1$). Hence, we have $\hat{y}(\mathbf{x}_*) = \text{sign}(\boldsymbol{\theta}^\top \mathbf{x}_*)$.

c) Logistic regression is a linear classifier (its decision boundaries are linear hyperplanes), whereas k-NN is a non-linear classifier. k-NN is thus a more flexible model which in general can fit training data more accurately. Therefore, we can *not* expect logistic regression to perform better than k-NN on the *training* set. We can however expect logistic regression to perform better on the *test* set, since it is less prone to overfitting.

4. (a) Two Gaussian pdfs $\mathcal{N}(x;\mu_{-1},\sigma_{-1}^2)$ and $\mathcal{N}(x;\mu_1,\sigma_1^2)$, as illustrated below. One is centered at $x=0$ and $\sim$95% of the probability mass is within $[-4,4]$. The other other is centered at $x=2$ and is wider.



(b) Recall that the best classifier is given by

$$f_0(x) = \arg\max_y p(y|x) = \arg\max_y \frac{p(x|y)p(y)}{p(x)} = \arg\max_y p(x|y)p(y)$$

using the chain rule. This means that the classifier outputs:

$$f_0(x) = \begin{cases} 1, & p(x|1)p(1) > p(x|-1)p(-1) \\ -1, & \text{otherwise} \end{cases}$$

Using logarithms, the inquality condition can also be expressed as:

$$\ln p(x|1) + \ln p(1) > \ln p(x|-1) + \ln p(-1)$$

Inserting the information we have about these distributions, we obtain

$$\ln \mathcal{N}(x;2,4) + \ln\frac{1}{3} > \ln \mathcal{N}(x;0,2) + \ln\frac{2}{3}$$

and so:

$$-\frac{(x-2)^2}{8} - \frac{\ln 4}{2} + \ln\frac{1}{3} > -\frac{(x-0)^2}{4} - \frac{\ln 2}{2} + \ln\frac{2}{3}$$

or

$$-\frac{(x-2)^2}{2} + x^2 > \underbrace{4(-\frac{\ln 2}{2} + \frac{\ln 4}{2} + \ln\frac{2}{3} - \ln\frac{1}{3})}_{=\ln 64}$$

(c) Evaluate classifier $f_0(x)$ at test input $x_\star = 1$. Since

$$-\frac{(1-2)^2}{2} + 1^2 = \frac{1}{2} \leq \ln 64,$$

the classifier output is $f_0(x_\star) = -1$. That is, no medication is detected.

5. (a) The first split divides X_1 into two half-spaces. The region $X_2 \geq 2.8$ corresponds to leaf node R_1. The second split divides the region $X_2 < 2.8$ at $X_1 = 2.2$ where the region $X_1 \geq 2.2$ corresponds to node R_2. Finally, the third split divides the region $X_2 < 2.8$ and $X_1 < 2.2$ at $X_1 = 0.9$ resulting in two regions where R_3 corresponds to $X_1 < 0.9$ and R_4 corresponds to $X_1 \geq 0.9$. The partitioning of the input space is thus as follows

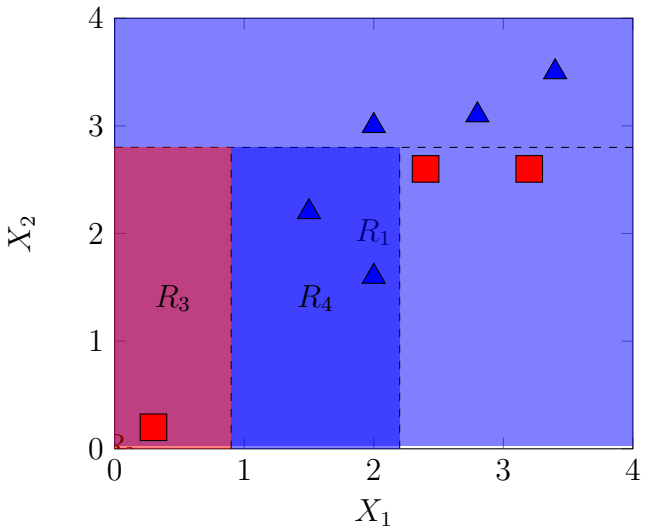


Figure 1: Partitioning of the input space for Alice's decision tree in Problem **??**.

(1p)

(b) Since $X_2^* = 1.8 < 2.8$, $X_1^* = 1.6 < 2.2$, and $X_1^* = 1.6 \geq 0.9$, the test point belongs to region R_4. To compute the predicted output we also need to know which regions the training data points fall into. We do this for all eight data points and get

X_1	X_2	Y	Region
2.4	2.6	`red`	R_2
3.2	2.6	`red`	R_2
2.0	3.0	`blue`	R_1
0.3	0.2	`red`	R_3
3.4	3.5	`blue`	R_1
2.8	3.1	`blue`	R_1
2.0	1.6	`blue`	R_4
1.5	2.2	`blue`	R_4

Thus, there are two training data points in region R_4, namely $\begin{bmatrix}1.5 & 2.2\end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix}2.0 & 1.6\end{bmatrix}^\mathsf{T}$. Since both data points are `blue`, the predicted output is `blue`. (1p)

(c) Alice used the Gini index as impurity measure whereas Bob used the classification error rate. There are multiple ways to approach this problem.

One indication is that in Alice's tree, the splits seem to always yield one pure and one mixed region. On the other hand, the second and third split in Bob's tree create two mixed regions. This can be seen also in Figure 2.

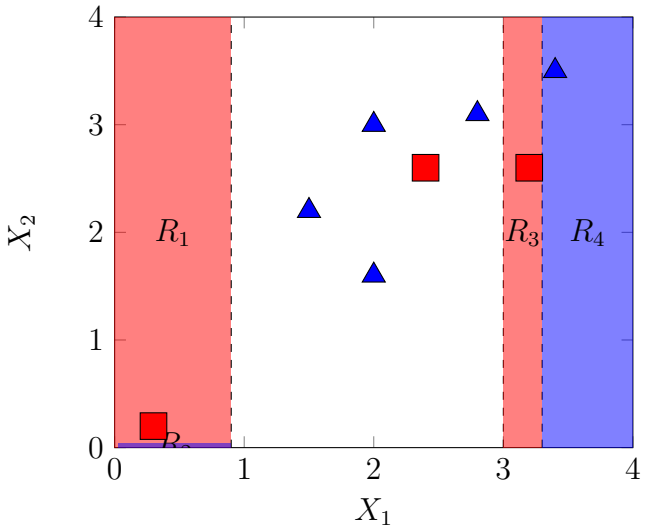


Figure 2: Partitioning of the input space for Bob's decision tree in Problem **??**.

Alternatively, one can study the first split in both trees and evaluate the objective functions with the Gini index and misclassification rate. The splits create regions with the following misclassification rates Q_m and Gini indices Q_g:

	Region: True					Region: False				
Split	n	$\widehat{\pi}_{\texttt{blue}}$	$\widehat{\pi}_{\texttt{red}}$	Q_m	Q_g	n	$\widehat{\pi}_{\texttt{blue}}$	$\widehat{\pi}_{\texttt{red}}$	Q_m	Q_g
$X_1 < 0.9$	7	5/7	2/7	2/7	20/49	1	0/1	1/1	0	0
$X_2 < 2.8$	5	2/5	3/5	2/5	12/25	3	3/3	0/3	0	0

Hence in total the values of the cost function are

	Cost function	
Split	Q_m	Q_g
$X_1 < 0.9$	2	20/7
$X_2 < 2.8$	2	12/5

We see that based on the Gini index one should rather choose the split $X_2 < 2.8$ (cost $12/5 = 2.4$) than the split $X_1 < 0.9$ (cost $20/7 \approx 2.86$). Thus we conclude that Alice used the Gini index as impurity measure. Since Bob used a different measure, apparently he used the misclassification error rate. (1p)

(d) A bagging classifier is constructed by averaging over an ensemble of flexible base models trained on different versions of the training dataset (often bootstrapped). In this way, the variance can be reduced without significantly increasing the bias, hence improving the model performance. (2p)

(e) In a bagging classifier, the base model predictions are identically distributed but correlated, since the bootstrapped data sets are generated from the same original data set. With a large number of ensemble members, the variance reduction of the bagging classifier is limited by the correlation ρ between the base models. If the same few features dominate in all of the bootstrapped data sets, the base models in an ensemble of decision trees can end up with very similar (or even identical) early splits, thus producing very similar (and highly correlated) predictions. (2p)

(f) Optimizing over a random subset of the input features randomly perturbs the training of the base models, which reduces the correlation between them. As a result, the averaged prediction can exhibit a larger variance reduction compared to a bagging classifier, for which the variance reduction is limited by the correlation between the base models. (1p)

(g) Hyperparameter optimization using k-fold cross validation or out-of-bag estimation. (1p)