EXAM IN STATISTICAL MACHINE LEARNING
STATISTISK MASKININLÄRNING

DATE: March 13, 2023

RESPONSIBLE TEACHER: Jens Sjölund

NUMBER OF PROBLEMS: 5

AIDING MATERIAL: Calculator, mathematical handbooks

PRELIMINARY GRADES:

grade 3	23 points
grade 4	33 points
grade 5	43 points

Some general instructions and information:

- Your solutions should be given in English.
- Only write on one page of the paper.
- Write your exam code and a page number on all pages.
- Do not use a red pen.
- Use separate sheets of paper for the different problems (i.e. the numbered problems, 1–5).

With the exception of Problem 1, all your answers must be clearly motivated! A correct answer without a proper motivation will score zero points!

Good luck!

1. This problem is composed of 10 true-or-false statements. You only have to classify these as either `true` or `false`. For this problem *(only!)* no motivation is required. Each correct answer scores 1 point and each incorrect answer scores -1 point (capped at 0 for the whole problem). Answers left blank score 0 points.

   i. False.

   ii. True.

   iii. False.

   iv. False.

   v. False.

   vi. True.

   vii. False.

   viii. False.

   ix. True.

   x. True. (10p)

2. (a) The design matrix X and corresponding outputs $\mathbf{y}$ are

$$X = \begin{pmatrix} u_1^{(\mathrm{A})} \\ u_2^{(\mathrm{A})} \\ u_3^{(\mathrm{A})} \end{pmatrix} = \begin{pmatrix} 59.8 \\ 59.2 \\ 58.4 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} u_2^{(\mathrm{A})} \\ u_3^{(\mathrm{A})} \\ u_4^{(\mathrm{A})} \end{pmatrix} = \begin{pmatrix} 59.2 \\ 58.4 \\ 58.5 \end{pmatrix}.$$

Note that these are just vectors, hence the ridge regression estimate reduces to

$$\begin{aligned} \theta_{\mathrm{RR}} &= (X^\top X + \lambda I)^{-1} X^\top \mathbf{y} \\ &= \frac{\mathbf{x}^\top \mathbf{y}}{\mathbf{x}^\top \mathbf{x} + \lambda} \approx \frac{10414}{10491 + \lambda}. \end{aligned}$$

For $\lambda = (10, 100, 1000)$, the corresponding estimates are $\theta = (0.992, 0.983, 0.906)$.

$\lambda = 0$ corresponds to the unregularized least-squares estimate. As $\lambda \to \infty$, $\theta \to 0$.

(b) Because the data for company B comes from a different distribution than the data for company A. In particular, the scale of both the inputs and outputs, as well as the noise, differs widely. Hence, a different regularization strength is needed to achieve the same effect.

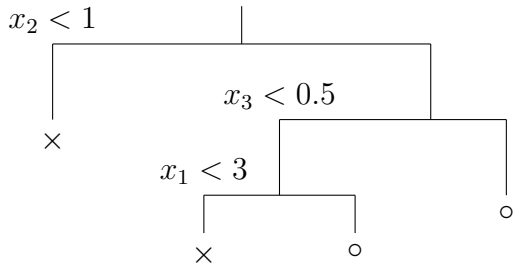
(c) Using the algebraic rules for the logarithm, we can rewrite the model as

$$\ln u_{k+1} = v \ln u_k + w + \epsilon.$$

This corresponds to a linear regression model $\mathbf{y} \approx X\boldsymbol{\theta}$ where

$$\begin{aligned} \boldsymbol{\theta} &= \begin{pmatrix} w \\ v \end{pmatrix}, \\ X &= \begin{pmatrix} 1 & \ln u_1^{(\mathrm{A})} \\ 1 & \ln u_2^{(\mathrm{A})} \\ 1 & \ln u_3^{(\mathrm{A})} \end{pmatrix} = \begin{pmatrix} 1 & 4.091 \\ 1 & 4.081 \\ 1 & 4.067 \end{pmatrix}, \\ \mathbf{y} &= \begin{pmatrix} \ln u_2^{(\mathrm{A})} \\ \ln u_3^{(\mathrm{A})} \\ \ln u_4^{(\mathrm{A})} \end{pmatrix} \approx \begin{pmatrix} 4.081 \\ 4.067 \\ 4.069 \end{pmatrix}. \end{aligned}$$

3. (a)



(b) When considering the Gini index $Q_\ell(T) = \sum_{m=1}^{M} \widehat{\pi}_{\ell m}(1-\widehat{\pi}_{\ell m})$, the cost function takes the following values for the different splits:

Split	n_1	$\widehat{\pi}_{1\circ}$	$\widehat{\pi}_{1\times}$	$Q_1(T)$	n_2	$\widehat{\pi}_{2\circ}$	$\widehat{\pi}_{2\times}$	$Q_2(T)$	$n_1Q_1(T) + n_2Q_2(T)$
$x_1 < 1$	2	1	0	0	5	1/5	4/5	0.32	1.6
$x_1 < 2$	4	3/4	1/4	0.375	3	0	1	0	1.5

Hence, the split at $x_1 < 2$ should be performed.

The Gini index, when compared to the misclassification loss, prioritizes splits that isolate comparatively pure regions. After a few splits, this can result in better classification than what would be achieved with the misclassification loss.

(c) i) *Relationship:* Bagging can use any type of base models. Random forest is an extension of bagging and its base model is a decision tree. *Differences:* Bagging reduces the correlation among base models by bootstrapping datasets. Random forests achieve furher decorrelation by injecting additional randomness in the construction of the tree (more specifically, when making the splits).

ii) A flexible model often suffers from overfitting (or equivalently, high variance). Bagging can often reduce the variance of the base model without signficantly increasing its bias, thus improving the overall performance.

(d) i) The most common choice in practice is to use a shallow tree, or even a decision stump (a tree of depth one). This is because boosting can reduce bias efficiently and can thereby learn good models despite using a very weak (high-bias) base model, and shallow trees can be trained quickly.

ii) As the number of base models increases, the boosted model becomes more and more flexible. While each additional model may reduce the bias, at some point the variance will start to increase faster, resulting in overfitting.

4. (a) To show that the two classifiers are equivalent, we manipulate the optimization objective step by step:

$$\begin{aligned}
\widehat{y} &= \underset{m=\{1,\ldots,M\}}{\arg\max}\ p(y=m|\mathbf{x}) \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ p(\mathbf{x}\,|\,m)p(m) \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ \log\Big(p(\mathbf{x}\,|\,m)p(m)\Big) \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ \log\Big(p(m)\Big) + \log\Big(p(\mathbf{x}\,|\,m)\Big) \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ \log(\pi_m) - \frac{1}{2}(\mathbf{x}-\mu_m)^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\mu_m) \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ \log(\pi_m) + \mathbf{x}^{\mathsf{T}}\Sigma^{-1}\mu_m - \frac{1}{2}\mu_m^{\mathsf{T}}\Sigma^{-1}\mu_m \\
&= \underset{m=\{1,\ldots,M\}}{\arg\max}\ \delta_m(\mathbf{x})
\end{aligned}$$

In each step of the derivation above, we have removed terms that are independent of the class m, since these terms do not affect the maximizing argument.

(b) The decision boundary between the two classes a and b is given by

$$p(y=a|\mathbf{x}) = p(y=b|\mathbf{x}) \Rightarrow \delta_a(\mathbf{x}) - \delta_b(\mathbf{x}) = 0$$

Using the expression for the score function given in (a), we find:

$$\begin{aligned}
\delta_a(\mathbf{x}) - \delta_b(\mathbf{x}) &= \log(\pi_a) + \mathbf{x}^{\mathsf{T}}\Sigma^{-1}\mu_a - \frac{1}{2}\mu_a^{\mathsf{T}}\Sigma^{-1}\mu_a \\
&\quad - \Big(\log(\pi_b) + \mathbf{x}^{\mathsf{T}}\Sigma^{-1}\mu_b - \frac{1}{2}\mu_b^{\mathsf{T}}\Sigma^{-1}\mu_b\Big) \\
&= \log\frac{\pi_a}{\pi_b} + \mathbf{x}^{\mathsf{T}}\Sigma^{-1}(\mu_a-\mu_b) - \frac{1}{2}(\mu_a+\mu_b)^{\mathsf{T}}\Sigma^{-1}(\mu_a-\mu_b),
\end{aligned}$$

which is a linear function in $\mathbf{x}$.

(c) The closest neighbors are $\{[2.3, 1.5, \times][1.5, 2, \bigcirc], [1.75, 1.25, \bigcirc]\}$. When k=3, by majority voting, the prediction $\hat{y}_\star$ is $\bigcirc$.

(d) To choose k, we iterate through a sequence of candidate k values. For each k-value, we perform 10-fold cross validation as follows.

Given the parameter k, we split the data set into 10 segments and iterate through them. For each segment, we take it as the validation dataset, and use the other 9 segments as training dataset. We then train the k-NN classifier with the given k on the train set and evaluate it on the validation set. Finally, we take the mean value of the 10 evaluation scores as the 10-fold cross-validation score for the given parameter k. We select the k with the best validation score.

(e) Plot 1 is on test set, plot 2 is on train set. Test error is generally higher than train error. Besides, when k = 1, train error of a k-NN model is 0.

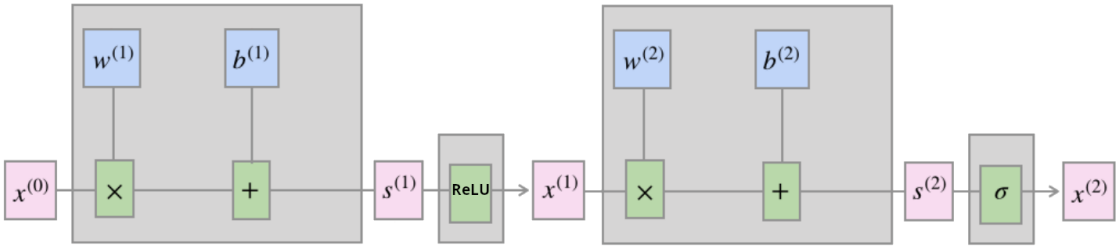


Figure 1: A neural network.

5. (a) Neural Network Task

   i. The (intermediate) results are (4p)

$$s^{(1)} = \begin{pmatrix} -0.3 \\ -1.2 \\ 1.6 \end{pmatrix}, \qquad x^{(1)} = \begin{pmatrix} 0.0 \\ 0.0 \\ 1.6 \end{pmatrix},$$
$$s^{(2)} = \begin{pmatrix} -0.6 \\ 1.5 \end{pmatrix}, \qquad x^{(2)} \approx \begin{pmatrix} 0.35 \\ 0.82 \end{pmatrix}.$$

   ii. The neural network has two layers (hidden and output) and can be used for (multi-label) classification (stating classification is enough). (1p)

   (b) Optimization Task

   i. Stochastic Gradient Descent (4p)

```
initialize r=10.00, use eta=0.30
epoch=0 minibatch=[6 2 4]
    obj=38.67 grad=12.00 r=10.00
    update r
    obj=8.43 grad=4.80 r=6.40
epoch=0 minibatch=[5 1 3]
    obj=14.23 grad=6.80 r=6.40
    update r
    obj=4.52 grad=2.72 r=4.36
epoch=1 minibatch=[6 2 4]
    obj=2.80 grad=0.72 r=4.36
    update r
```

```
    obj=2.69 grad=0.28 r=4.14
epoch=1 minibatch=[5 1 3]
    obj=3.97 grad=2.28 r=4.14
    update r
    obj=2.88 grad=0.92 r=3.46
```

Alternative of i. Gradient Descent (2p)

```
initialize r=10.00, use eta=0.30
t=0
    obj=45.17 grad=13.00 r=10.00
    update r
    obj=9.68 grad=5.20 r=6.10
t=1
    obj=9.68 grad=5.20 r=6.10
    update r
    obj=4.00 grad=2.08 r=4.54
t=2
    obj=4.00 grad=2.08 r=4.54
    update r
    obj=3.09 grad=0.84 r=3.92
t=3
    obj=3.09 grad=0.84 r=3.92
    update r
    obj=2.95 grad=0.34 r=3.67
t=4
    obj=2.95 grad=0.34 r=3.67
    update r
    obj=2.92 grad=0.14 r=3.57
t=5
    obj=2.92 grad=0.14 r=3.57
    update r
    obj=2.92 grad=0.06 r=3.53
```

ii. When the learning rate is too large GD might overshoot and diverge. We expect a very slow convergence by using a too small learning rate. (1p)