Exam in Statistical Machine Learning
Statistisk Maskininlärning (1RT700)

Date and time: March 17, 2019, 14.00–19.00

Responsible teacher: Johan Wågberg

Number of problems: 5

Aiding material: Calculator, mathematical handbooks,
one (1) hand-written sheet of paper with notes and formulas (A4, front and back)

Preliminary grades:

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

Some general instructions and information:

- Your solutions can be given in Swedish or in English.
- Write only on one side of the paper.
- Write your exam code and 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).
- For subproblems (a), (b), (c), . . . , it is usually possible to answer later subproblems independently of the earlier subproblems (for example, you can answer (b) without answering (a)).

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!

Some relevant formulas

Pages 1–3 contain some expressions that may or may not be useful for solving the exam problems. *This is not a complete list of formulas used in the course*, but some of the problems may require knowledge about certain expressions not listed here. Furthermore, the formulas listed below *are not self-explanatory*, meaning that you need to be familiar with the expressions to be able to interpret them. They are possibly a support for solving the problems, but *not* a comprehensive summary of the course.

The Gaussian distribution: The probability density function of the p-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is

$$\mathcal{N}\left(\mathbf{x} \,|\, \boldsymbol{\mu},\, \boldsymbol{\Sigma}\right) = \frac{1}{(2\pi)^{p/2}\sqrt{\det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \qquad \mathbf{x} \in \mathbb{R}^p.$$

Sum of identically distributed variables: For identically distributed random variables $\{z_i\}_{i=1}^n$ with mean μ, variance σ^2 and average correlation between distinct variables ρ, it holds that $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n z_i\right] = \mu$ and $\text{Var}\left(\frac{1}{n}\sum_{i=1}^n z_i\right) = \frac{1-\rho}{n}\sigma^2 + \rho\sigma^2$.

Linear regression and regularization:

- The least-squares estimate of $\boldsymbol{\theta}$ in the linear regression model

$$y = \theta_0 + \sum_{j=1}^{p} \theta_j x_j + \epsilon$$

  is given by the solution $\widehat{\boldsymbol{\theta}}_{\text{LS}}$ to the normal equations $\mathbf{X}^{\mathsf{T}}\mathbf{X}\widehat{\boldsymbol{\theta}}_{\text{LS}} = \mathbf{X}^{\mathsf{T}}\mathbf{y}$, where

$$\mathbf{X} = \begin{bmatrix} 1 & -\mathbf{x}_1^{\mathsf{T}}- \\ 1 & -\mathbf{x}_2^{\mathsf{T}}- \\ \vdots & \vdots \\ 1 & -\mathbf{x}_n^{\mathsf{T}}- \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \text{ from the training data } \mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$$

- Ridge regression uses the regularization term $\lambda\|\boldsymbol{\theta}\|_2^2 = \lambda\sum_{j=0}^p \theta_j^2$.
  The ridge regression estimate is $\widehat{\boldsymbol{\theta}}_{\text{RR}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$.
- LASSO uses the regularization term $\lambda\|\boldsymbol{\theta}\|_1 = \lambda\sum_{j=0}^p |\theta_j|$.

Maximum likelihood: The maximum likelihood estimate is given by

$$\widehat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \ln \ell(\boldsymbol{\theta})$$

where $\ln \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(y_i \,|\, \mathbf{x}_i; \boldsymbol{\theta})$ is the log-likelihood function (the last equality holds when the n training data points are modeled to be independent).

Logistic regression: The logistic regression combines linear regression with the logistic function to model the class probability

$$p(y = 1 \,|\, \mathbf{x}) = \frac{e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}}}{1 + e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}}}.$$

For multi-class logistic regression we use the *softmax* function and model

$$p(y = m \,|\, \mathbf{x}_i) = \frac{e^{\boldsymbol{\theta}_m^\mathsf{T}\mathbf{x}_i}}{\sum_{j=1}^{M} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}_i}}.$$

Discriminant Analysis: The linear discriminant analysis (LDA) classifier models $p(y \,|\, \mathbf{x})$ using Bayes' theorem and the following assumptions

$$p(y = m \,|\, \mathbf{x}) = \frac{p(\mathbf{x} \,|\, m)p(y = m)}{\sum_{j=1}^{M} p(\mathbf{x} \,|\, j)p(y = j)} \approx \frac{\mathcal{N}\left(\mathbf{x} \,|\, \widehat{\boldsymbol{\mu}}_m,\, \widehat{\boldsymbol{\Sigma}}\right) \widehat{\pi}_m}{\sum_{j=1}^{M} \mathcal{N}\left(\mathbf{x} \,|\, \widehat{\boldsymbol{\mu}}_j,\, \widehat{\boldsymbol{\Sigma}}\right) \widehat{\pi}_j},$$

where

$$\begin{aligned}
\widehat{\pi}_m &= n_m/n \text{ for } m = 1, \ldots, M \\
\widehat{\boldsymbol{\mu}}_m &= \frac{1}{n_m} \sum_{i:y_i=m} \mathbf{x}_i \text{ for } m = 1, \ldots, M \\
\widehat{\boldsymbol{\Sigma}} &= \frac{1}{n - M} \sum_{m=1}^{M} \sum_{i:y_i=m} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}.
\end{aligned}$$

For quadratic discriminant analysis (QDA), the model is

$$p(y = m \,|\, \mathbf{x}) \approx \frac{\mathcal{N}\left(\mathbf{x} \,|\, \widehat{\boldsymbol{\mu}}_m,\, \widehat{\boldsymbol{\Sigma}}_m\right) \widehat{\pi}_m}{\sum_{j=1}^{M} \mathcal{N}\left(\mathbf{x} \,|\, \widehat{\boldsymbol{\mu}}_j,\, \widehat{\boldsymbol{\Sigma}}_j\right) \widehat{\pi}_j},$$

where $\widehat{\boldsymbol{\mu}}_m$ and $\widehat{\pi}_m$ are as for LDA, and

$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{n - 1} \sum_{i:y_i=m} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}.$$

Classification trees: The cost function for tree splitting is $\sum_{\ell=1}^{|T|} n_\ell Q_\ell$ where T is the tree, $|T|$ the number of terminal nodes, n_ℓ the number of training data points falling in node ℓ, and Q_ℓ the impurity of node ℓ. Three common impurity measures for splitting classification trees are:

$$\begin{aligned}
\text{Misclassification error:} \qquad & Q_\ell = 1 - \max_m \widehat{\pi}_{\ell m} \\
\text{Gini index:} \qquad & Q_\ell = \sum_{m=1}^{M} \widehat{\pi}_{\ell m}(1 - \widehat{\pi}_{\ell m}) \\
\text{Entropy/deviance:} \qquad & Q_\ell = -\sum_{m=1}^{M} \widehat{\pi}_{\ell m} \log \widehat{\pi}_{\ell m}
\end{aligned}$$

where $\widehat{\pi}_{\ell m} = \frac{1}{n_\ell} \sum_{i:\, \mathbf{x}_i \in R_\ell} \mathbb{I}(y_i = m)$

Loss functions for classification: For a binary classifier expressed as $\widehat{y}(\mathbf{x}) = \text{sign}\{C(\mathbf{x})\}$, for some real-valued function $C(\mathbf{x})$, the margin is defined as $y \cdot C(\mathbf{x})$ (note the convention $y \in \{-1, 1\}$ here). A few common loss functions expressed in terms of the margin, $L(y, C(\mathbf{x}))$ are,

$$\begin{aligned}
\text{Exponential loss:} \qquad & L(y, c) = \exp(-yc). \\
\text{Hinge loss:} \qquad & L(y, c) = \begin{cases} 1 - yc & \text{for } yc < 1, \\ 0 & \text{otherwise.} \end{cases} \\
\text{Binomial deviance:} \qquad & L(y, c) = \log(1 + \exp(-yc)). \\
\text{Huber-like loss:} \qquad & L(y, c) = \begin{cases} -yc & \text{for } yc < -1, \\ \frac{1}{4}(1 - yc)^2 & \text{for } -1 \leq yc \leq 0, \\ 0 & \text{otherwise.} \end{cases} \\
\text{Misclassification loss:} \qquad & L(y, c) = \begin{cases} 1 & \text{for } yc < 0, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

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, each incorrect answer scores -1 point and each missing answer scores 0 points. The total score will never be less than 0.

   i. Logistic regression is a linear classifier.

   ii. For classification, the input variables have to be categorical.

   iii. Cross-validation can be used to learn the regularization parameter λ in ridge regression.

   iv. When using LDA for binary classification, the mid point between two clusters $\overline{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2}$ will always give $p(y = 1|\mathbf{x} = \overline{\boldsymbol{\mu}}) = \frac{1}{2}$.

   v. A non-parametric model for classification always achieve zero training error since the complexity grows with data.

   vi. A neural network with linear activation functions is linear in the input variables.

   vii. When bootstrapping a dataset, it is important to sample without replacement.

   viii. Normalizing the dataset is important for the performance of a classification tree.

   ix. Boosting is typically used to improve large models with small bias and high variance.

   x. K-nearest neighbors is a generative model.

(10p)

2. Consider the following training data

i	x	y
1	1.0	2.8
2	2.0	5.0
3	3.0	7.5
4	4.0	8.7
5	5.0	11.1
6	6.0	12.9

where i is an index, $\mathbf{x}$ is the one-dimensional input variable and y the output.

(a) Fit a straight line to the data $y = \theta_0 + \theta_1 x$ using linear regression. Illustrate the training data in a graph with x and y on the two axes together with your estimated line.

(4p)

(b) If you try to fit a 5:th order polynomial to the training data

$$\widehat{y}_{\boldsymbol{\theta}}(x) = \sum_{\ell=0}^{5} \theta_\ell x^\ell = \boldsymbol{\theta}^\mathsf{T}\mathbf{x}, \quad \mathbf{x} = \begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 \end{bmatrix}^\mathsf{T}$$

using the cost function $J(\boldsymbol{\theta}) = \sum_{i=1}^{6} \left(y_i - \widehat{y}_{\boldsymbol{\theta}}(x_i)\right)^2$. What training error will you attain?

Note: You should not try to compute $\widehat{\boldsymbol{\theta}}$.

(1p)

(c) A colleague gives you three different estimates $\widehat{\boldsymbol{\theta}}$ but they are only labeled a, b, and c. The colleague has tried LASSO and ridge regression in addition to the unregularized estimate in (b). Which of the estimated $\widehat{\boldsymbol{\theta}}$ corresponds to which type of regularization?

$$\widehat{\boldsymbol{\theta}}_\text{a} = \begin{bmatrix} 0.9 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$$
$$\widehat{\boldsymbol{\theta}}_\text{b} = \begin{bmatrix} 15 & -29 & 23 & -7.9 & 1.2 & -0.07 \end{bmatrix}^\mathsf{T}$$
$$\widehat{\boldsymbol{\theta}}_\text{c} = \begin{bmatrix} 1.9 & 0.44 & 0.069 & 0.01 & 0.0014 & 0.0002 \end{bmatrix}^\mathsf{T}$$

Note: The values have been rounded to two significant digits.

Note: You do not have to "prove" which parameter estimate corresponds to which regularization. A short argument is enough.

(2p)

(d) Instead of regularization you decide to try bagging (bootstrap aggregating). You sample three datasets with replacement. The indices of the sampled datasets can be seen in Table 1.

Dataset	Indices
$\mathcal{T}^{(1)}$	5, 0, 5, 1, 4, 4
$\mathcal{T}^{(2)}$	3, 1, 0, 4, 1, 5
$\mathcal{T}^{(3)}$	4, 3, 0, 5, 2, 2

Table 1: Indices of the datapoints in each bootstrapped dataset.

If you try to fit the 5th order polynomial, do you forsee any potential problems with this approach?

(1p)

(e) From the bootstraped datasets in (d), you get three different parameter estimates

$$\widehat{\boldsymbol{\theta}}^{(0)} = \begin{bmatrix} -29.8 & 56.4 & -28.1 & 4.31 & 0.0894 & -0.0425 \end{bmatrix}^\mathsf{T},$$
$$\widehat{\boldsymbol{\theta}}^{(1)} = \begin{bmatrix} -1.09 & 5.15 & -1.46 & 0.187 & 0.0141 & -0.00296 \end{bmatrix}^\mathsf{T},$$
$$\widehat{\boldsymbol{\theta}}^{(2)} = \begin{bmatrix} -2.25 & 4.73 & 1.39 & -1.36 & 0.31 & -0.0221 \end{bmatrix}^\mathsf{T}.$$

The final bagging predictor is again a linear model $\widehat{y}_{\text{bag}}(x) = \widehat{\boldsymbol{\theta}}_{\text{bag}}^\mathsf{T}\boldsymbol{x}$. Compute $\widehat{\boldsymbol{\theta}}_{\text{bag}}$ and the prediction for $x = 2$.

(2p)

3. Consider the dataset in Table 2.

i	x_1	x_2	y
1	6.0	6.0	$\times$
2	4.0	2.0	$\circ$
3	5.0	3.0	$\circ$
4	2.0	5.0	$\times$
5	7.0	5.0	$\times$
6	3.0	0.0	$\times$
7	3.0	1.0	$\times$
8	4.0	4.0	$\circ$
9	7.0	1.0	$\times$
10	2.0	1.0	$\times$
11	3.0	6.0	$\times$

Table 2: Dataset for question 3, where i is an index, $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}$ is the two-dimensional input and y is the class label.

(a) Illustrate the dataset in a graph with x_1 and x_2 on the two axes.

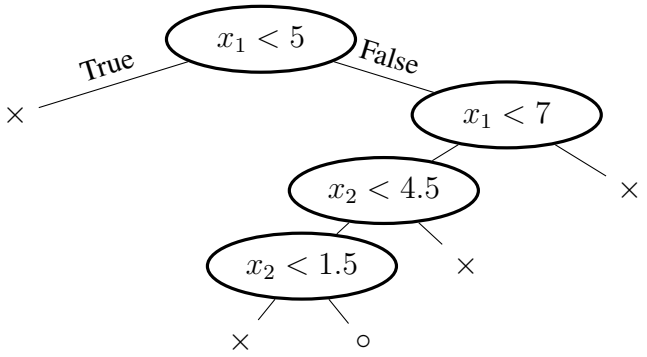
(2p)

(b) Construct a classification tree using recursive binary splitting and Gini index as impurity measure. Stop when all leaves contains a single class. Present the tree similar to the one in task (c) below.

(4p)

(c) Draw a graph showing the partitioning induced by the decision boundaries of the classification tree below.

Note: Do not draw the partitioning for the tree you got in task (b)



(2p)

(d) Which of the following methods can attain zero training error (for some (hyper-) parameter setting) for the dataset in Table 2? Only consider x_1 and x_2 as inputs and no transformations of the features are allowed.

i. k-nearest-neighbors.

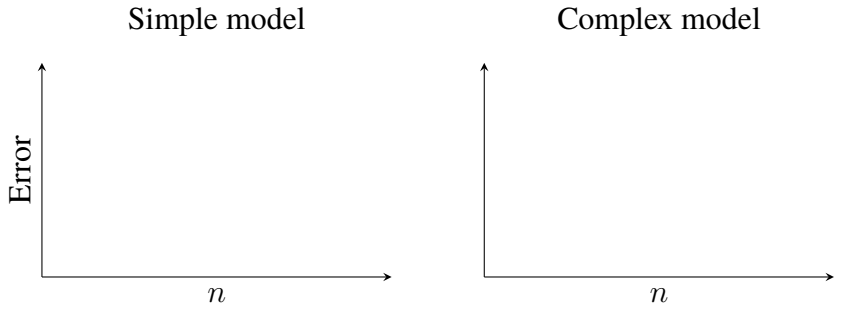
ii. QDA (Quadratic discriminative analysis).

iii. Logistic regression.

(2p)

4. (a) Draw two graphs showing the typical *training* and *validation error* as a function of the number of training data points n for a simple model and a complex model. Clearly label the *two* curves in each graph and make sure the graphs are comparable in terms of x- and y-axes. Comment on the similarities and differences between the two graphs.

   It might look something like the figure below.



(3p)

(b) Describe cross-validation in a few sentences (max 1/2 page).

(3p)

(c) You want to decide on k in a k-nearest neighbors classifier. Using cross-validation, you have produced the graph in Figure 1. Which value of k do you choose if you want to minimize the expected "in production" error?

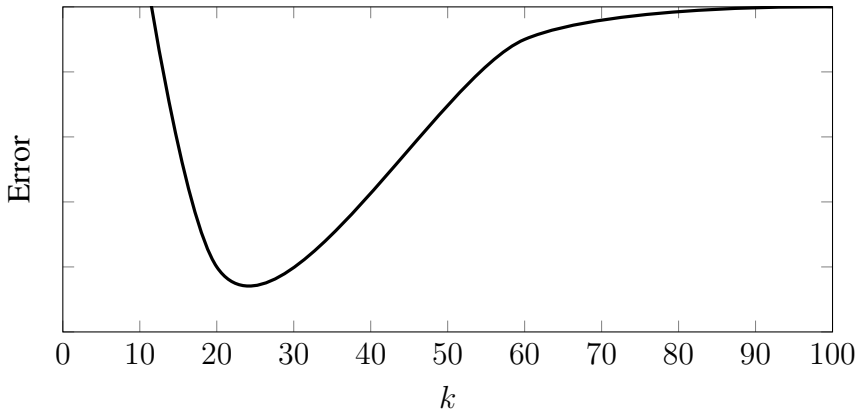


Figure 1: Error estimated using cross-validation as a function of k.

(1p)

(d) Describe why the curve in Figure 1 looks the way it does in terms of the bias-variance trade-off.

(3p)

5. Consider binary classification where we model the probability. The so-called log odds is defined as the logarithm ratio of the probabilities of the two classes,

$$\text{log odds} = \ln \frac{p(y=1|x)}{p(y=-1|x)}. \tag{1}$$

(a) What is the range[1] of $p(y = 1 \mid x)$?

(1p)

(b) What is the range[1] of the log odds?

(2p)

(c) Model the positive class by $p(y = 1|x) = g(x)$. Show that fitting a linear regression model to the log odds in (1) gives the same model as the one used in logistic regression.

(4p)

(d) A function that transforms the output to make it suitable for linear regression is sometimes called a link function. In (1), we transform the probability via the logarithm of the odds which is also called the logit function

$$\text{logit}(\mu) = \ln \frac{\mu}{1-\mu}.$$

The logit function is the canonical link function for binary classification. (The inverse of the logit function is called the logistic function.) Though the logit function is the canonical link function, it is not the only possible choice of link function. Which of the following functions would also work as a link function for binary classification?

i. $g(\mu) = \Phi^{-1}(\mu)$, the inverse of the cumulative distribution function for a standard Gaussian random variable.

ii. $g(\mu) = \tan\left(\pi\left(\mu - \frac{1}{2}\right)\right)$.

iii. $g(\mu) = \ln(-\ln(1-\mu))$.

(3p)

[1]The range is all the possible values a function can attain.