Exam in Statistical Machine Learning
Statistisk Maskininlärning (1RT700)

Date and time: March 15, 2019, 08.00–13.00

Responsible teacher: Niklas Wahlström

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.
- 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).
- 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

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

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 $\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^n z_i\right) = \frac{1-\rho}{n}\sigma^2 + \rho\sigma^2$.

Linear regression and regularization:

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

  $$y = \beta_0 + \sum_{j=1}^p \beta_j x_j + \epsilon$$

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

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

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

- LASSO uses the regularization term $\gamma\|\boldsymbol{\beta}\|_1 = \gamma\sum_{j=0}^p |\beta_j|$.

Maximum likelihood: The maximum likelihood estimate is given by

$$\widehat{\boldsymbol{\beta}}_{\text{ML}} = \arg\max_{\boldsymbol{\beta}} \log \ell(\boldsymbol{\beta})$$

where $\log \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log p(y_i \,|\, \mathbf{x}_i; \boldsymbol{\beta})$ 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

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

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

$$p(y = k \,|\, \mathbf{x}_i) = \frac{e^{\boldsymbol{\beta}_k^\mathsf{T}\mathbf{x}_i}}{\sum_{l=1}^{K} e^{\boldsymbol{\beta}_l^\mathsf{T}\mathbf{x}_i}}.$$

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

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

where

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

For quadratic discriminant analysis (QDA), the model is

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

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

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

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

$$\begin{aligned}
\text{Misclassification error:} \qquad & Q_m = 1 - \max_k \widehat{\pi}_{mk} \\
\text{Gini index:} \qquad & Q_m = \sum_{k=1}^{K} \widehat{\pi}_{mk}(1 - \widehat{\pi}_{mk}) \\
\text{Entropy/deviance:} \qquad & Q_m = -\sum_{k=1}^{K} \widehat{\pi}_{mk} \log \widehat{\pi}_{mk}
\end{aligned}$$

where $\widehat{\pi}_{mk} = \frac{1}{n_m} \sum_{i:\mathbf{x}_i \in R_m} \mathbb{I}(y_i = k)$

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,

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

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 for the whole problem is capped below at 0.

   i. A nonlinear classifier can never have a linear decision boundary.

   ii. A neural network is an ensemble method.

   iii. Convolutional neural networks are well suited for classification problems where the input is an image.

   iv. Deep learning is a parametric method.

   v. A marketing company want to build a model for predicting the number of visitors to a web page. Since the number of visitors is an integer, this is best viewed as a classification problem.

   vi. One should not split datasets randomly into training and test data, but always take the last data points as the test data.

   vii. The k-NN classifier most often suffers from overfitting when $k = 1$.

   viii. Neural networks can only be used for classification problems, and not for regression problems.

   ix. Regularization can be used to avoid overfitting in linear regression.

   x. Regularization can only be used for regression methods,
   and not for classification methods.

(10p)

2. Consider the following training data

i	$\mathbf{x}$		y
	x_1	x_2	
1	0.0	5.0	1
2	4.0	1.0	0
3	1.0	2.0	1
4	4.0	2.0	0
5	0.0	3.0	1
6	1.0	8.0	1
7	9.0	0.0	0
8	6.0	5.0	0
9	8.0	6.0	0
10	5.0	7.0	1

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

(a) Illustrate the training data points in a graph with x_1 and x_2 on the two axes. Represent the points belonging to class 0 with a cross and those belonging to class 1 with a circle. Also annotate the data points with their data point indices.

(1p)

(b) Based on the training data we want to construct a random forest classifier with $B = 3$ trees, each of depth one (i.e., stumps). For this we randomly draw 3 new datasets by bootstrapping the training data (sampling with replacement). We also randomly draw an input dimension index (1 or 2), along which the split is to be performed (if the split dimension is 1, the split is on the form $x_1 < c$, etc). The following data points indices have been drawn for each of the three bootstrapped datasets:

	Data point indices i									
Dataset 1	1	2	4	4	6	7	8	9	9	10
Dataset 2	1	2	3	4	5	5	6	7	7	9
Dataset 3	1	1	4	5	5	5	6	9	9	9

	Split dimension
Tree 1	2
Tree 2	2
Tree 3	1

For each bootstrapped dataset, construct a classification stump (tree of depth one) by finding the split among the prescribed dimension which minimizes the Gini index.

(5p)

(c) The final random forest classifier predicts according to a majority vote of the three trees. Sketch the decision boundary of the final classifier.

(4p)

3. (a) Why is the training error not a good estimate of the test error? Explain in a few sentences.

(3p)

(b) What is the purpose of using cross-validation? Explain in a few sentences.

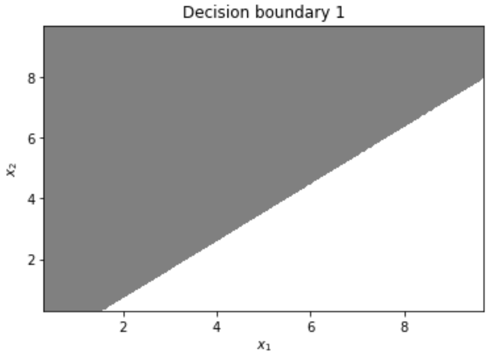
(3p)

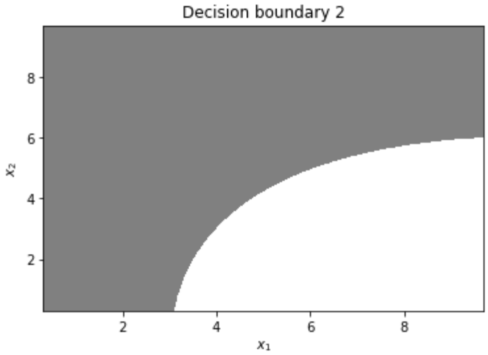
(c) For each of the following decision boundaries (1, 2, 3 and 4), tell whether it could possibly come from one of these classifiers

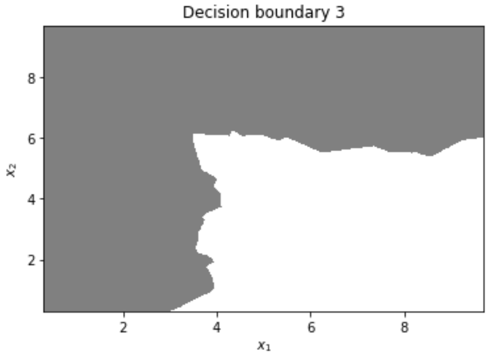
- logistic regression
- LDA
- QDA
- k-NN with $k = 3$
- A decision tree of depth 2
- A random forest with $B = 3$ trees, each of maximum depth 3

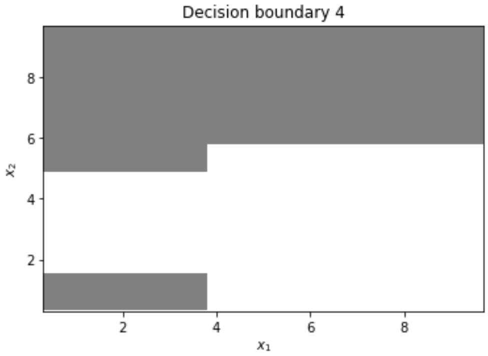
It is assumed that each classifier only uses x_1 and x_2 as inputs, and no nonlinear transformations of them.

Note! Each decision boundary could possibly origin from more than one classifier. Do not forget to include a motivation for all your answers.









(4p)

4. (a) Consider the following training data

i	$\mathbf{x}$		y
	x_1	x_2	
1	-1	2	2
2	0	0	1
3	1	-2	-3

from which we want to learn a linear regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon,$$

where we assume that ε has a Gaussian distribution. Compute $\widehat{\boldsymbol{\beta}}$!

(3p)

(b) How would $\widehat{\boldsymbol{\beta}}$ change if you were to use regularized linear regression (e.g. Ridge regression or LASSO) in (a) instead? *It is enough to provide a qualitative explanation, you do not need to compute* $\widehat{\boldsymbol{\beta}}$.

(2p)

(c) In the context of neural networks, describe, using a few sentences, the difference between a dense layer and a convolutional layer.

(2p)

(d) Describe how mini-batch gradient descent works and what the main advantage is in comparison to gradient descent.

(3p)

We have after the exam realized that the inverse in $\mathbf{H}(\gamma)$ was missing in the actual exam. We will take this typo into consideration in the grading.

5. For leave-one-out cross validation (or equivalently c-fold cross validation with $c = n$) the cross validation error E_{val} for ridge regression actually has a closed-form solution

$$E_{\text{val}} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\hat{y}_i - y_i}{1 - [\mathbf{H}(\gamma)]_{ii}} \right)^2, \tag{1}$$

where $\hat{y}_i$ is the prediction of y_i when the model is learned from *all* n data points (no data point is left out), γ the regularization parameter and $[\mathbf{H}(\gamma)]_{ii}$ is element (i, i) of the matrix $\mathbf{H}(\gamma) = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}$.

In this problem, we will let $\mathbf{x}_i^\mathsf{T}$ denote the entire ith row of $\mathbf{X}$.

(a) Let $\mathbf{X}_{-i}$ denote the matrix $\mathbf{X}$ where row i is removed, and $\mathbf{y}_{-i}$ is the column vector $\mathbf{y}$ with element i removed. Show that

$$\begin{aligned} \mathbf{X}_{-i}^\mathsf{T}\mathbf{X}_{-i} &= \mathbf{X}^\mathsf{T}\mathbf{X} - \mathbf{x}_i\mathbf{x}_i^\mathsf{T}, \\ \mathbf{X}_{-i}^\mathsf{T}\mathbf{y}_{-i} &= \mathbf{X}^\mathsf{T}\mathbf{y} - \mathbf{x}_i y_i, \text{ and} \\ [\mathbf{H}(\gamma)]_{ii} &= \mathbf{x}_i^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{x}_i \end{aligned}$$

(1p)

(b) Using the results from (a) and a special case of the matrix inversion lemma

$$(\mathbf{A} - \mathbf{v}\mathbf{v}^\mathsf{T})^{-1} = \mathbf{A}^{-1} + \frac{\mathbf{A}^{-1}\mathbf{v}\mathbf{v}^\mathsf{T}\mathbf{A}^{-1}}{1 - \mathbf{v}^\mathsf{T}\mathbf{A}^{-1}\mathbf{v}},$$

show that

$$\widehat{\boldsymbol{\beta}}_{-i} = \widehat{\boldsymbol{\beta}} + \frac{1}{1 - [\mathbf{H}(\gamma)]_{ii}}(\mathbf{X}^\mathsf{T}\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{x}_i(\hat{y}_i - y_i).$$

Here $\widehat{\boldsymbol{\beta}}_{-i}$ are the parameters learned from all data points except i, and $\widehat{\boldsymbol{\beta}}$ the parameters learned from all data.

Hint: Start from the ridge regression expression for $\widehat{\boldsymbol{\beta}}_{-i}$ as a function of $\mathbf{X}_{-i}$, $\mathbf{y}_{-i}$ and γ.

(4p)

(c) Use your result from (b) to derive eq. (1), starting from

$$E_{\text{val}} = \frac{1}{n} \sum_{i=1}^{n} \left(\widehat{\boldsymbol{\beta}}_{-i}^\mathsf{T}\mathbf{x}_i - y_i \right)^2.$$

(2p)

(d) Describe (in a few sentences) what eq. (1) can be used for in practice.

(3p)