Exam in Statistical Machine Learning
Statistisk Maskininlärning (1RT700)

Date and time: August 21, 2019, 08.00–13.00

Responsible teacher: Andreas Lindholm

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 most often answer (b) without answering (a)).
- If you are enrolled at any other study program than a civilingenjörsprogram, you will *not* be allowed to take a later re-exam (to improve your grade) if you score grade 3 or higher on this exam. No exceptions will be made.

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 $\boldsymbol{\beta}$ 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

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

- 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 $p(y \,|\, \mathbf{x})$ 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

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

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 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 for the whole problem is capped below at 0.

   i. The expected mean squared error for new, previously unseen data points can be decomposed into a sum of squared bias, variance and irreducible error.

   ii. Logistic regression is a regression method.

   iii. c-fold cross validation can be used for selecting a good value of k in k-NN.

   iv. One could use LDA as base classifier in boosting.

   v. The least squares problem always has a unique solution $\widehat{\boldsymbol{\beta}}$.

   vi. In binary classification, the output can take only two possible values.

   vii. LASSO and Ridge Regression are mathematically equivalent.

   viii. Regularization may prevent overfit.

   ix. k-NN is a nonparametric method.

   x. Deep neural networks can only be used for regression, not classification.

(10p)

2. Consider a dataset with grayscaled images of birds, each consisting of 40×40 scalar pixels. Each image is labeled with one of the four classes $y \in \{\texttt{eagle}, \texttt{owl}, \texttt{pigeon}, \texttt{swallow}\}$. For this data we design a small convolutional neural network with one convolutional layer and one dense layer.

   The convolutional layer is parameterized with a weight tensor $\mathbf{W}^{(1)}$ and an offset vector $\mathbf{b}^{(1)}$ producing a hidden layer $\mathbf{H}$. The convolutional layer has the following design

Number of kernels/output channels	16
Kernel rows and columns	(3×3)
Stride	[2,2]

   The stride [2,2] means that the kernel is moving by two steps (both row- and column-wise) during the convolution such that the hidden layer has half as many rows and columns as the input image.

   The dense layer is parameterized with the weight matrix $\mathbf{W}^{(2)}$ and bias vector $\mathbf{b}^{(2)}$ producing the logits $\mathbf{z}$. The logits $\mathbf{z}$ are pushed through a softmax function to produce predictions of the four class probabilities $p(y = \texttt{eagle} \,|\, \mathbf{x})$, etc.

   (a) How many elements are there in the weight tensor $\mathbf{W}^{(1)}$, the offset vector $\mathbf{b}^{(1)}$ and the hidden layer $\mathbf{H}$?

   (2p)

   (b) How many elements are there in the weight tensor $\mathbf{W}^{(2)}$, the offset vector $\mathbf{b}^{(2)}$ and the logits $\mathbf{z}$?

   (2p)

   (c) What is the total number of parameters used to paramaterize this network?

   (1p)

   (d) Describe, in a few sentences, the difference between a dense layer and a convolutional layer in a neural network.

   (2p)

   (e) What is the difference between gradient descent and stochastic/mini-batch gradient descent?

   (1p)

   (f) What is an epoch in stochastic/mini-batch gradient descent?

   (1p)

   (g) What is the advantage of using stochastic/mini-batch gradient descent instead of regular gradient descent?

   (1p)

3. (a) Explain ($\sim$ 1/2 page) how regularization relates to bias and variance. Use the connection between bias and variance to also argue why regularization can be useful in practice.

(3p)

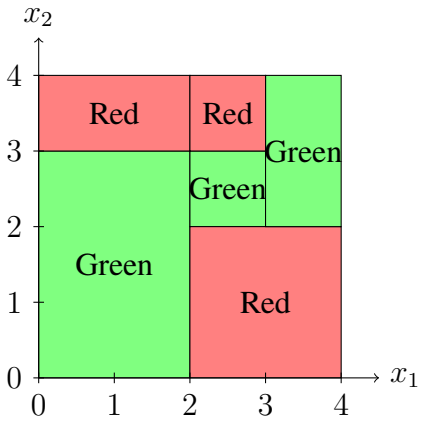
(b) What is overfit, and *why* do you want to avoid it? Explain in a few sentences.

(2p)

(c) What is the extension of random forests, compared to bagging? What is the benefit of that extension?

(3p)

(d) Sketch a classification tree which corresponds to the following partitioning:



(2p)

4. Consider the following $n = 5$ training data points for a binary classification problem ($y = \{-1, 1\}$ here) with one input variable:

x_i	1.75	2	2.5	2.75	3
y_i	1	-1	-1	1	1

(a) A classifier is constructed as $\widehat{y}(x) = \text{sign}(x - 2.625)$. Compute both the misclassification loss and the exponential loss for each training data point, for this classifier. Also compute the corresponding average loss for all data points.

(2p)

(b) Would it be possible for a LDA classifier to obtain zero misclassifications on this data? Would it be possible for a QDA classifier to obtain zero misclassifications on this data?

Note: You do not need to perform any computations for answering this question.

(2p)

(c) Learn a LDA classifier from the training data (that is, compute all the parameters in the LDA classifier). Also compute its decision boundary. What misclassification rate do you achieve for the training data?

(3p)

(d) Learn a QDA classifier from the training data (that is, compute all the parameters in the QDA classifier). Also compute its decision boundary. What misclassification rate do you achieve for the training data?

(3p)

5. (a) Training a model using maximum likelihood amounts to solving

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \log p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta}).$$

To solve this optimization problem, we need to have a mathematical expression for $p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta})$. Derive an expression for the logarithm of $p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta})$ for the logistic regression model, in the binary classification case when the output classes are $y = \{-1, 1\}$. In more detail, show that

$$\log p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta}) = \sum_{i=1}^{n} -\log(1 + \exp(-y_i \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}_i)), \tag{1}$$

where p denotes the probability density, $\mathbf{y}$ is the vector of output samples y_i, $\mathbf{X}$ is a matrix with input samples $\mathbf{x}_i$, and $\boldsymbol{\beta}$ is a vector with the logistic regression parameters.

(Note: The output class labeling $y = \{-1, 1\}$ is important when deriving (1)*. Another labeling, such as $\{0, 1\}$, would give another expression.)*

(7p)

(b) Consider a qualitative input variable that can take the four different values $\{\texttt{red}, \texttt{green}, \texttt{blue}, \texttt{pink}\}$. We want to use this input for a classification problem, for example using it in (1). How can this qualitative input be encoded using dummy variables?

(3p)