# Exam in Statistical Machine Learning
# Statistisk Maskininlärning (1RT700)

**Date and time:** June 15, 2019, 09.00–14.00

**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)).

*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

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

**Sum of identically distributed variables:** For identically distributed random variables $\{z_i\}_{i=1}^n$ with mean $\mu$, variance $\sigma^2$ and average correlation between distinct variables $\rho$, 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

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

- 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

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

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 $\widehat{y}(\mathbf{x}) = \operatorname{sign}\{C(\mathbf{x})\}$, 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 for the whole problem is capped below at 0.

   i. LASSO and Ridge Regression are two different methods for regularization

   ii. Regularization decreases the bias of the model

   iii. Deep learning is a nonparametric method

   iv. The model $y = \beta_0 + \beta_1 x_1 + \beta_2 \sin(x_2) + \varepsilon$ is a linear regression model ($\beta_0, \beta_1$ and $\beta_2$ are the unknown parameters)

   v. Classification problems have only qualitative inputs

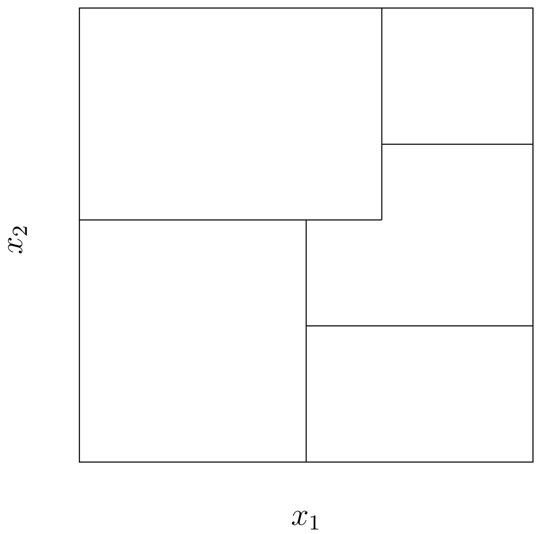
   vi. $k$-NN is a linear classifier if $k = 1$

   vii. Random forest is a special version of boosting with trees

   viii. The training error usually increases when we increase the model flexibility

   ix. An epoch, in the context of stochastic gradient descent, is the number of iterations required for the training to converge

   x. The partitioning of the input space shown below could be generated by recursive binary splitting



(10p)

2. A wholesaler of chocolate has asked you to build a model for predicting the purchase price of different chocolate bars based on various types of information about the chocolates origin, production, etc. They have collected a database with their data, containing the following columns:

   - `id` – a unique identification number for each row of the database, specified as an integer value
   - `bean` – one out of 3 different cocoa beans (`Forastero`, `Trinitario`, `Criollo`)
   - `percentage` – percentage of cocoa in the chocolate, specified as a real number between 0 and 1 (e.g. $0.65$ for $65\%$)
   - `year` – harvest year of the cocoa bean, specified as an integer in the range 2010–2019
   - `origin` – one out of 52 different regions where the cocoa been has been produced (e.g. `Ghana`, `Cameroon`, `Brazilia`, etc.)
   - `producer` – an integer in the range 1–394, where each integer represents a different producer (the name of the producer is found in another database)
   - `milk` – whether the chocolate contains milk or not (`yes` or `no`)
   - `timestamp` – a time stamp specifying the time when the row was entered into the database, on the format `'YYYY-MM-DD HH:MM:SS'`
   - `weight` – the weight of the bar in gram, specified as an integer (typically in the range 50-400)
   - `price` – the price (in SEK) that was paid at the last purchase of this type of chocolate bar, specified as an integer (typically in the range 10–500)

   *Don't forget to clearly motivate all your answers!*

   (a) The customer want's to try a simple model first, like a linear regression or a logistic regression. Which one of these two methods do you suggest? (2p)

   (b) For each column of the customer's database as listed above, specify whether you would consider that variable as an input of the model, an output of the model, or neither. (3p)

   (c) For each of the inputs and outputs of your model (from the previous question), specify whether that variable is best viewed as quantitative or qualitative. (3p)

   (d) In a previous attempt to design such a system, the inputs used were `origin` (treated as qualitative), `producer` (treated as qualitative) and `percentage` (treated as quantitative). At that time, the database contained 183 rows. No satisfactory performance was obtained. Give a plausible explanation why. (2p)

3. Consider the following training data

| $i$ | $\mathbf{x}$ | | $y$ |
|---|---|---|---|
| | $x_1$ | $x_2$ | |
| 1 | 3 | -2 | 1 |
| 2 | 3 | -7 | 0 |
| 3 | 9 | -3 | 1 |
| 4 | 10 | -5 | 1 |
| 5 | 2 | -2 | 1 |
| 6 | -7 | 1 | 0 |
| 7 | 0 | 5 | 0 |
| 8 | 9 | -8 | 0 |

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) Perform leave-one-out cross-validation (which is equivalent to 8-fold cross-validation here) for $k$-NN with $k = 1$ and $k = 3$ to estimate the misclassification rate for new data points.

*It is OK to determine the closest neighbors graphically using your figure from problem (a), as long as your approach is well documented by your solution. A misclassification rate stated without any comments or explanations will score 0 points.*

(4p)

(c) What is your conclusion from (b) regarding a good choice of $k$ in $k$-NN for this problem?

(1p)

(d) Describe ($\sim 1/2$ page) how the flexibility of $k$-NN is different for different values of $k$ (in general, not restricted to the specific problem in (a)–(c)), and discuss how it relates to the bias-variance trade-off. What does it mean in practice for a user of $k$-NN, and why can we not achieve low bias *and* low variance at the same time?

(3p)

(e) For the binary classification problem, what potential issue is there to use an even $k$ (such as $k = 2$) in $k$-NN, and how can it be handled in practice?

(1p)

4. (a) Figure 1 shows the training data for a binary classification problem with two inputs, where the two classes are marked by blue dots and orange crosses, respectively. A logistic regression classifier is constructed for this problem, which attains a zero misclassification training error. Describe how this is possible despite the fact that logistic regression is a linear classifier (i.e., it has linear decision boundaries).

(2p)

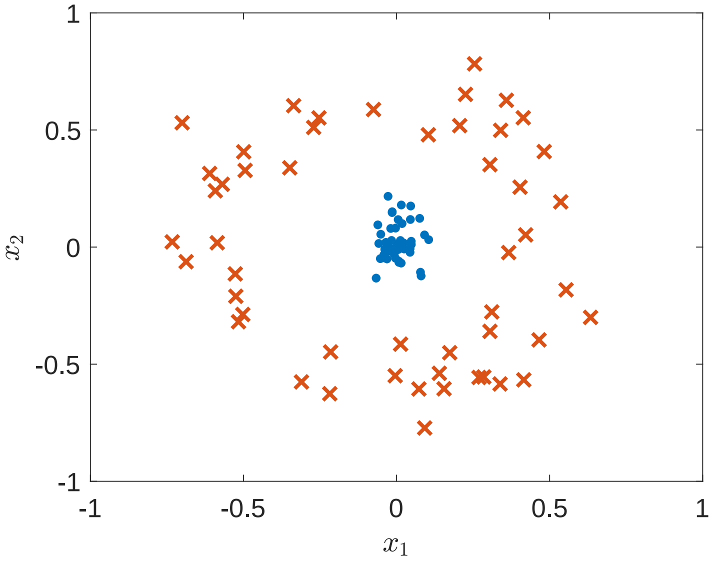


Figure 1: Training data for the classification problem 4a

(b) Derive an expression for the decision boundary in QDA for a binary classification problem, when the 'decision threshold' is 0.5. The expression should be on the format $\{\mathbf{x} : \mathbf{x}^\mathsf{T}\mathbf{v} + \mathbf{x}^\mathsf{T}\mathbf{B}\mathbf{x} = c\}$, with $c$ a scalar, $\mathbf{v}$ a vector and $\mathbf{B}$ a matrix.

(3p)

(c) Give an example on an application of binary classification where it could be motivated to use a 'decision threshold' different than 0.5.

(2p)

(d) Your colleague has a regression problem he needs to solve, with two possible inputs $x_1$ and $x_2$. He tries two different linear regression models

(M1) $y = \beta_0 + \beta_1 x_1 + \varepsilon$, and

(M1) $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$.

He train the two models successfully from the same data set using least squares, and obtain

(T1) $\widehat{y} = 7.2 + 1.1 x_1$, and

(T1) $\widehat{y} = 8.1 - 2.9 x_1 + 5.1 x_2$.

Your colleague studies the result, becomes puzzled and asks you: "If I increase the input $x_1$ with one unit in model (T1), my prediction $\widehat{y}$ will increase by 1.1. However, a unit increase in $x_1$ for model (T2) will instead decrease my prediction $\widehat{y}$ with 2.9. How could it be that the two models do not even agree on whether an increase in $x_1$ should decrease or increase $y$?"

Give a plausible explanation to your colleague on possible reasons for this situation. (3p)

5. (a) Explain using a few sentences ($\sim$ max 1/2 page) the differences and similarities between bagging and boosting. (4p)

(b) Consider a binary classification problem with one input and with training data set

$$\mathcal{T} = \{(x_1, y_1),\ (x_2, y_2),\ (x_3, y_3)\} = \{(0, +1),\ (1, -1),\ (2, +1)\}.$$

An ensemble classifier with $B$ ensemble members is given by

$$\widehat{y}_{\text{boost}}(x) = \text{sign}\left(\sum_{b=1}^{B} \alpha_b \widehat{y}^b(x)\right)$$

where $\alpha_b > 0$ is the "confidence" of the $b$th classifier, and each base classifier is assumed to be linear, i.e., on the form

$$\widehat{y}^b(x) = +\text{sign}\,(x - \beta_b) \qquad \text{or} \qquad \widehat{y}^b(x) = -\text{sign}\,(x - \beta_b)$$

for some parameter value $\beta_b$.

Find a classifier $\widehat{y}_{\text{boost}}(x)$ with $B = 3$ ensemble members

$$\widehat{y}^1(x), \widehat{y}^2(x), \widehat{y}^3(x)$$

and their corresponding confidences $\alpha_1, \alpha_2, \alpha_3$, which completely separates the training data. (4p)

(c) Show that it is not possible to attain zero training error using only two ensemble members in the previous question. You may assume that $\alpha_1 \neq \alpha_2$ for simplicity (though, the statement holds also for $\alpha_1 = \alpha_2$). (2p)

*N.B. It is possible to solve 5(c) without solving 5(b).*

*(For mathematical rigour we use the convention* $\text{sign}(0) = 0$, *but it is not necessary to use this fact to solve the problem.)*