Exam in Statistical Machine Learning
Statistisk Maskininlärning (1RT700)

Date and time: June 17, 2020, 08.00–13.00

Responsible teacher: Johan Wågberg

Number of problems: 5

Aiding material: Calculator, mathematical handbooks, the course book.

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 θ in the linear regression model

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

  is given by the solution $\widehat{\theta}_{\text{LS}}$ to the normal equations $\mathbf{X}^{\mathsf{T}}\mathbf{X}\widehat{\theta}_{\text{LS}} = \mathbf{X}^{\mathsf{T}}\mathbf{y}$, 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 $\lambda\|\theta\|_2^2 = \lambda\sum_{j=0}^p \theta_j^2$.
  The ridge regression estimate is $\widehat{\theta}_{\text{RR}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$.
- 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. A k-nearest neighbors classifier always attains zero training error for $k = 1$ for datasets where no inputs are repeated, i.e. $\mathbf{x}_i \neq \mathbf{x}_j \; \forall i \neq j$.

   ii. Linear discriminative analysis (LDA) is a non-parametric model.

   iii. A convolutional layer in a neural network can only appear directly after the input layer.

   iv. A classifier is called linear if the function that maps each input to a predicted class is linear in the parameters.

   v. To talk about the bias-variance tradeoff only has meaning when learning by minimizing the mean squared error cost function.

   vi. Bootstrap aggregating (bagging) works best on simple models with high bias and low variance.

   vii. Logistic regression and linear discriminative analysis (LDA) will always produce the same decision boundary for binary classification problems.

   viii. Regularization, like ridge regression and LASSO, adds an extra term to the cost function.

   ix. A model with lower bias always performs better than a model with higher bias in terms of the mean squared error on test data.

   x. Linear regression requires all input variables to be numerical (quantitative).

   (10p)

2. Consider the dataset in Table 1.

x_1	x_2	y
-2.0	-1.0	$\circ$
2.0	-1.0	$\circ$
0.0	2.0	$\circ$
-4.0	3.0	$\times$
0.0	-5.0	$\times$
4.0	3.0	$\times$

Table 1: Data for problem 2, where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}$ is the input and y the class label.

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

(1p)

(b) A fellow student has worked with the data and has learned six different classifiers

i. Logistic regression
ii. LDA
iii. QDA
iv. kNN with $k = 1$
v. kNN with $k = 3$
vi. A classification tree with a single binary split based on misclassification rate

The student has also computed the misclassification error on the training data for each classifier. The results are shown Table 2. Unfortunately, the labels are missing.

50%	33%	33%	17%	0%	0%

Table 2: Misclassification errors.

Which misclassification error corresponds to which classifier?

Hint: *It is possible to solve the exercise without any complicated calculations. A correct pairing without a proper motivation scores zero points!*

(6p)

(c) Modify one (1) coordinate $\mathbf{x}$ such that all classifiers give zero misclassification error for the training data.

Hint: *You do not have to explicitly compute all the classifiers. It is enough to give a convincing argument that there exist parameters in eg. logistic regression that give zero misclassification error.*

(3p)

3. Consider a regression problem with two inputs x_1 and x_2 and one output y. The data is given in Table 3.

x_1	x_2	y
1.4	1.5	0.5
0.2	0.9	0.2
1.8	1.2	0.7
1.2	0.9	0.7
1.6	0.2	0.1
1.8	0.9	0.7
1.1	0.7	0.6
0.2	1.8	0.1

Table 3: Data for problem 3.

A regression tree shown in Figure 1 has already been constructed.

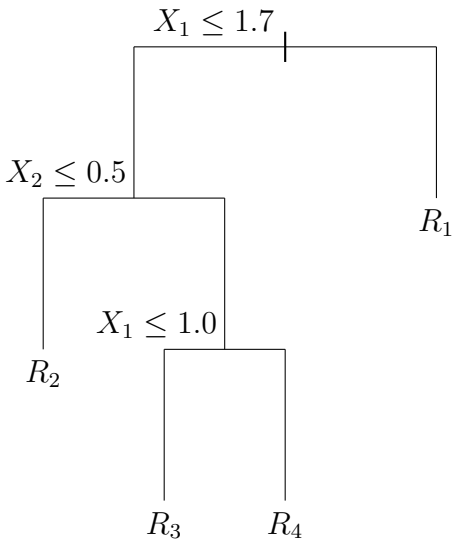

Figure 1: Decision tree for problem 3

(a) Draw the corresponding input partitioning for this tree. Mark the leaf nodes $R_1, \ldots, R_4$.

(2p)

(b) Continue to grow the tree in Figure 1 until there are at most two data points in each leaf node by minimizing the mean-square-error. Which region(s) do you split and where?

(3p)

(c) Boosting and bagging (bootstrap aggregating) are two algorithms that can be applied on top of decision trees. Describe similarities and differences between the two algorithms in terms of how they use the base model and what types of base models that are best suited for each algorithm in terms of the their complexity. Relate these concepts to the hyperparameters of a tree.

Note: The hyperparameters of a tree can, for example, be the depth of the tree or the size of the leaf nodes or the impurity measure used when splitting and so on.

(5p)

4. A company is working on a machine learning problem with $p = 5000$ input variables $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_{5000} \end{bmatrix}^\mathsf{T}$ with the aim to predict a single binary output $y \in \{-1, 1\}$. They have collected $N = 200$ datapoints with equally many examples from each class.

   (a) From previous experience, most of the features are useless for predicting y. To build a model, the company has executed the following steps:

      i. Compute the linear correlation between each input and the output and keep only the 100 most correlated input variables (positive or negative correlation).
      ii. Learn a logistic regression model.
      iii. Evaluate the performance of the model by running 10-fold cross-validation to estimate the accuracy on new, unseen data.

   The cross-validation indicated an accuracy of over 95% for the model. However, when they tried it in production, the performance was much worse.

   Did the company do something wrong or was the problem simply too hard? Give advice as to how the company should act.

   (4p)

   (b) The company also tried LASSO. They learned the parameters of a logistic regression model by minimizing the cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{200} \ln\left(1 + e^{-y_i \boldsymbol{\theta}^\mathsf{T} \mathbf{x}_i}\right) + \frac{1}{\lambda} \sum_{i=1}^{5000} |\theta_i|.$$

   To learn the regularization parameter λ, they have produced the graph in Figure 2.

   Explain why the graph looks the way it does. Specifically, explain the shape in terms of the bias-variance tradeoff and which term dominates where. Explain also what bias and variance mean in terms of model complexity and how this relates to λ.

   (4p)

   (c) The left part of the graph in Figure 2 is completely flat. Why?

   (2p)

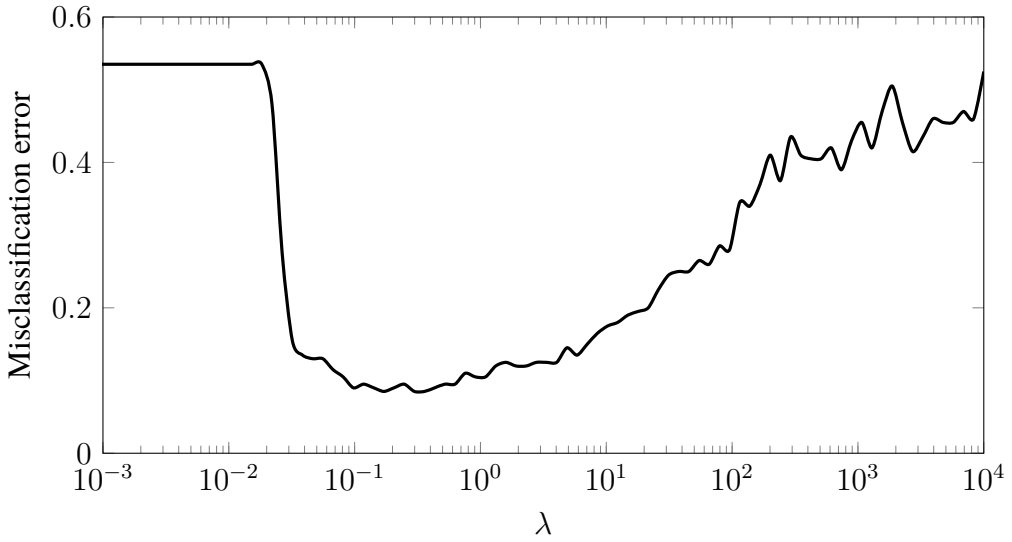

Figure 2: Misclassification error estimated using 10-fold cross-validation for different values of the regularization parameter λ. Used in problem 4 (b) and (c)

5. Given N input and output data points $\{(x_i, y_i)\}_{i=1}^N$, consider the model

$$y(x) = f(x) + \epsilon,$$

where ϵ is independent zero-mean noise with variance σ^2 and

$$f(x) = \sum_{i=1}^{N} \theta_i \phi_{x_i}(x), \text{ with } \phi_{x_i}(x) = e^{-(x-x_i)^2}.$$

That is, we model the function f as a sum of functions $\phi_c(x)$ centered on every point in the input data. The parameters $\boldsymbol{\theta}$ are learned by minimizing the mean squared error.

(a) Is the model parametric or non-parametric?

(1p)

(b) Show that the parameters can be learned using linear regression.
Hint: *Write the model as a matrix multiplication* $\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}$ *and give an expression for* $\mathbf{X}$

(2p)

(c) To avoid overfit, we use ridge regression with weight λ and learn the parameters by minimizing

$$J(\boldsymbol{\theta}) = \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 + \lambda\|\boldsymbol{\theta}\|^2,$$

where $\|\cdot\|$ is the standard ℓ_2-norm[1]. Show that the prediction of a test point $x_\star$ is given by

$$\widehat{f}(x_\star) = \mathbf{X}_\star \left(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \mathbf{X}^\mathsf{T}\mathbf{y},$$

where $\mathbf{X}_\star$ is formed analog to $\mathbf{X}$.

(2p)

(d) Let us now extend the model by placing basis functions *everywhere* on the real line. For fixed $D < \infty$, we put basis functions in the interval $[-D, D]$ at points $\xi_i = \frac{i}{2^D}$ for $i = -D2^D, \cdots, D2^D$. This gives us the model

$$f(x) = \sum_{i=-D2^D}^{D2^D} \theta_i \phi_{\xi_i}(x).$$

We learn the parameters using ridge regression with $\lambda = \sigma^2 2^D$. This scales the regularization with the distance between the centers of the basis functions and the noise variance. That is, the cost function is given by

$$J(\boldsymbol{\theta}) = \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 + \sigma^2 2^D \|\boldsymbol{\theta}\|^2.$$

[1] $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$

Let $D \to \infty$ so that we place basis functions everywhere on the real line. Show that the predictions can be computed as

$$\widehat{f}(x_\star) = \mathbf{k}_\star^\mathsf{T} \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y},$$

where

$$\mathbf{k}_\star = \begin{bmatrix} k(x_\star, x_1) & k(x_\star, x_2) & \cdots & k(x_\star, x_N) \end{bmatrix}^\mathsf{T},$$
$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) \end{bmatrix}$$

and

$$k(x, x_\star) = \sqrt{\frac{\pi}{2}} e^{-\frac{1}{2}(x - x_\star)^2}.$$

(Using this trick, we have transformed our scalar input to an infinite amount of inputs but are still able to compute the predictions analytically without any approximations.)

Hint: *You might find the following expressions useful:*

$$\left(\mathbf{I} + \mathbf{B}^\mathsf{T}\mathbf{B}\right)^{-1} \mathbf{B}^\mathsf{T} = \mathbf{B}^\mathsf{T} \left(\mathbf{I} + \mathbf{B}\mathbf{B}^\mathsf{T}\right)^{-1}$$
$$\frac{1}{2^D} \sum_{i=-D2^D}^{D2^D} \phi_i(x)\phi_i(x_\star) \overset{D\to\infty}{\to} \int_{-\infty}^{\infty} \phi_t(x)\phi_t(x_\star) dt$$
$$\int_{-\infty}^{\infty} e^{-a(x+b)^2} dx = \sqrt{\frac{\pi}{a}}$$

(5p)