EXAM IN STATISTICAL MACHINE LEARNING
STATISTISK MASKININLÄRNING

DATE AND TIME: March 10, 2022, 8.00–13.00

RESPONSIBLE TEACHER: Jens Sjölund

NUMBER OF PROBLEMS: 5

AIDING MATERIAL: Calculator, mathematical handbook

PRELIMINARY GRADES:	grade 3	23 points
	grade 4	33 points
	grade 5	43 points

Some general instructions and information:

- Your solutions should be given 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).

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!

Formula sheet for Statistical Machine Learning

Warning: This is not a complete list of formulas used in the course, some exam problems may require expressions not listed here. Furthermore, the formulas below are not self-explanatory, you need to be familiar with the expressions to interpret them.

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 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 $\boldsymbol{\theta}$ 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

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

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

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

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

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

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

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

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

Loss functions for classification: For a binary classifier expressed as $\widehat{y}(\mathbf{x}) = \operatorname{sign}\{C(\mathbf{x})\}$, 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,

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

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 and each incorrect answer scores -1 point (capped at 0 for the whole problem).

   i. The bias of the model decreases as the size of the training dataset goes to infinity.

   ii. Solving a logistic regression problem using gradient descent can lead to multiple local optimum solutions.

   iii. A higher value of the regularization hyperparameter in a linear regression problem with L^1 regularization (also called $LASSO$) leads to a more sparse model, where fewer model parameters are non-zero.

   iv. In a neural network model, a convolutional layer uses significantly fewer parameters compared to the dense layer with the same number of hidden units.

   v. A too large number of ensemble members leads to an increased complexity of a bagging model and results in a higher variance.

   vi. Bagging allows to estimate the expected new data error E_{new} without cross-validation.

   vii. It is easy to parallelize the training of a boosted model.

   viii. For models that are trained iteratively, a lower training error E_{train} can be achieved by training longer.

   ix. The model $y = \theta_1 x_1 + \theta_1^2 x_2 + \epsilon$ is an example of a linear regression model with a parameters θ_1, θ_1^2, input variables x_1, x_2 and a noise term ϵ.

   x. Enforcing a maximum depth for the tree can help reduce overfitting in decision trees. (10p)

2. A toy production company wants to specify a model that can predict the maximum speed v of their latest electric walking dinosaur given the maximum torque τ of the motor used in the controller. They collect a series of measurements and contact Liz, a machine learning engineer working as a consultant at a nearby firm. The company asks Liz to fit a linear regression model of the form $v = \beta_0 + \beta_1 \tau + \epsilon$ to their dataset using maximum likelihood.

   "This is easy! All I have to do is estimate the model parameters β_0 and β_1 using the least squares cost," Liz says.

   (a) Do you agree with Liz? Motivate your answer. (1p)

   Due to new EU-regulations, the company also needs to conduct safety classification tests on each toy. They ask Liz to find a model that classifies each toy as 'safe' or 'not safe' depending on the toy's total mass m and maximum speed v, and send her the measurements from the latest safety tests.

   Liz decides to train a logistic regression classifier using the following label transformation: $\{\text{safe, not safe}\} = \{-1, 1\}$. The input $\tilde{\mathbf{x}} = [1\ m\ v]^T$ gives the model parameter vector $\Theta = [-6.75\ 0.5\ 1]^T$.

   (b) To evaluate her model, Liz has set aside a test dataset according to Table 1. What is the decision boundary of the model, and what misclassification rate on the test data should she report back to the company? (3p)

m	7	4	9	7	3	8
v	4	5	4	3	2	3
class	1	1	1	-1	-1	-1

Table 1: Test data for problem 2b).

   The company thinks that more experimental data can improve the model. They ask Liz to include the material of the electric walking dinosaur toys as an additional model feature and send her a new dataset including the feature column 'material', where each toy is classified according to material: 'rubber', 'plastic', 'wood' or 'metal'. Liz determines that it is not possible to incorporate categorical input features in a logistic regression model.

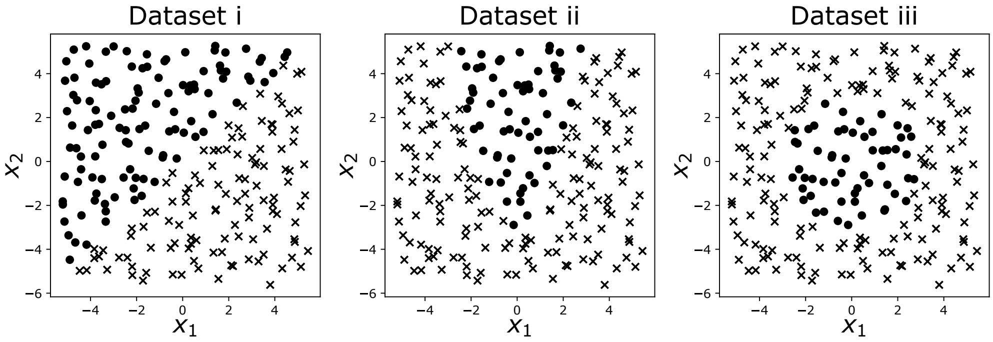

Figure 1: Datasets for problem 2d).

(c) Is Liz correct? If yes, explain why it is not possible. If not, how do you suggest that she incorporate the new feature into the model? (2p)

The company is impressed with the quality of the simple logistic regression classifier. They send Liz the three independent datasets depicted in Figure 1. Details regarding what the features and labels represent are classified, but they ask if it is possible to find a logistic regression classifier that can separate the classes in each of the three datasets.

(d) Give a general criterion as to when logistic regression can be expected to perform well. Then, for each of the three datasets, determine whether or not it is possible to separate the classes using logistic regression. For the datasets where logistic regression would work well, give the input features that you would use. When it is not possible, explain why. (4p)

Note: Question 2(a)-2(d) can be solved independently.

3. To complete a marathon (42.2 km) in less than 3 hours is considered a major milestone by many runners. Given a runner's age and their split time at the halfway mark of a marathon, we want to be able to predict whether they will complete the race in less than 3 hours ($y = 1$) or not ($y = -1$). We will consider two specific runners (runner A and B), which have the following data:

   - Runner A is 40 years old and passes the halfway mark in 90 min., i.e. $x_A = \begin{bmatrix}40 & 90\end{bmatrix}^\mathsf{T}$.
   - Runner B is 30 years old and passes the halfway mark in 84 min., i.e. $x_B = \begin{bmatrix}30 & 84\end{bmatrix}^\mathsf{T}$.

   We consider three models to predict the success of our runners in this problem: k-NN, LDA and QDA.

   We will use real-world data from Stockholm marathon 2021 to build our models.

   a) In this part, we consider 6 data points with runner id$= \{1, \dots, 6\}$ as training data, as shown in Table 2. Compute the missing $L2$ distances to our runner A in the table. Using the complete table, then make a prediction y for our runner A using:

      i k-NN with $k = 1$ and

      ii k-NN with $k = 3$.

Runner id	x		y	$\|x - x_A\|_2$
1	38	82	1	missing
2	32	85	1	9.4
3	33	83	-1	missing
4	28	85	-1	13.0
5	44	93	-1	5.0
6	57	97	-1	18.4

Table 2: Available data points and distances to runner A for k-NN.

(2p)

   b) We now want to fit a QDA model to the 6 training data points given in Table 2. A QDA model has three learnable parameters for each class. Compute all parameters corresponding to the $y = 1$ class. (3p)

c) When fitting an LDA model to a larger training dataset with 100 runners, we obtain the following parameters for the classifier:

- $\hat{\pi}_1 = 0.31$, $\hat{\pi}_{-1} = 0.69$
- $\hat{\mu}_1 = \begin{bmatrix} 36.4 & 83.1 \end{bmatrix}^\mathsf{T}$, $\hat{\mu}_{-1} = \begin{bmatrix} 41.2 & 93.5 \end{bmatrix}^\mathsf{T}$
- $\hat{\Sigma} = \begin{bmatrix} 25.5 & 3.3 \\ 3.3 & 90.0 \end{bmatrix}$

Compute the predictions of the model for Runner A and Runner B. The following values of the normal distribution are given:

- $\mathcal{N}(x_A|\hat{\mu}_1, \hat{\Sigma}) = 0.0020$
- $\mathcal{N}(x_A|\hat{\mu}_{-1}, \hat{\Sigma}) = 0.0030$
- $\mathcal{N}(x_B|\hat{\mu}_1, \hat{\Sigma}) = 0.0014$
- $\mathcal{N}(x_B|\hat{\mu}_{-1}, \hat{\Sigma}) = 0.00019$

Then, obtain “hard” predictions from the LDA model by thresholding at $r = 0.5$, and compute the average misclassification error of the predictions given that both runners finish the race in less than 3 hours. (3p)

d) Now draw a scatter plot of fictional data (i.e. the data does not have to be realistic) with half marathon time split on the x-axis and age on the y-axis, using "○" for class $y = -1$ and "×" for class $y = 1$. Draw the data in such a way that QDA has a clearly better fit to the data than LDA. Draw the decision boundary for QDA and explain in one sentence why QDA is a better fit for the data you drew. (2p)

4. The minimax loss is sometimes used to optimize worst-case performance. It is defined as the maximum absolute error and is thus mathematically equivalent to the uniform norm on the errors, i.e.,

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_\infty = \max_{i=1,\ldots,n} |y_i - \hat{y}_i|.$$

In this problem, your task is to construct various tree-based models for regression based on the minimax loss. The base model is a stump, i.e., a regression tree with a single split (depth 1).

Table 3 gives the training data, and the predictions $\hat{\mathbf{y}}$ for different models.

x	1	3	5	7	$L(\mathbf{y}, \hat{\mathbf{y}})$
y	0	5	2	1	
$\hat{y}_{\text{stump}}$	0	8/3	8/3	8/3	7/3
$\hat{y}_{\text{bagging}}$	?	?	?	?	?
$\hat{y}_{\text{minimax}}$	?	?	?	?	?

Table 3: Training data and predictions for the different tree-based models.

(a) Is the minimax loss *robust*? Explain! (1p)

(b) Only consider the cutpoints $s \in \{2, 4, 6\}$ and show that $s = 2$ is the optimal cutpoint, in terms of the minimax loss, for the base model when using the squared error (as in the lectures) to construct predictions for each leaf.
Plot the predictions $\hat{y}_{\text{stump}}(x)$ for $x \in [0, 8]$, and verify that the predictions agree with Table 3. (2p)

(c) Only consider the cutpoints $s \in \{4, 6\}$ and use bagging (bootstrap aggregation) to create an ensemble of models, constructed as in part (b), from the two bootstrapped datasets below:

$$\mathcal{T}^{(1)} = \{(3, 5), (5, 2), (5, 2)\},$$
$$\mathcal{T}^{(2)} = \{(1, 0), (5, 2), (7, 1)\},$$

where we use the notation $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^n$.
Plot the predictions $\hat{y}_{\text{bagging}}(x)$ for $x \in [0, 8]$, and clearly state the value of the entries in the $\hat{y}_{\text{bagging}}$ row in Table 3 which are indicated with a question mark. (3p)

(d) Consider a stump with the single cutpoint $s = 2$ and use the minimax loss to construct predictions for each leaf. Clearly state the value of the entries in the $\hat{y}_{\text{minimax}}$ row in Table 3 which are indicated with a question mark. (2p)

(e) Should you, in general, expect bagging to improve the performance of this base model? What about boosting?
Hint: think in terms of bias and variance. (2p)

Note: Question 4(a)-4(e) can be solved independently.

5. (a) The sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$ is a commonly used non-linear activation function in neural networks. Show that $\sigma(-z) = 1 - \sigma(z)$. (2p)

(b) The main operation within convolutional neural networks (CNNs) is the convolution. Let $\mathbf{x} \in \mathbb{R}^{4\times3}$ be the input and $\mathbf{f} \in \mathbb{R}^{3\times1}$ be a filter/kernel. We are interested in the convolution of $\mathbf{x}$ and $\mathbf{f}$.

i. Compute the convolution of $\mathbf{x} = \begin{pmatrix} 0 & 0 & 0 \\ 4 & -2 & 1 \\ 3 & 8 & 5 \\ 0 & 0 & 0 \end{pmatrix}$ and $\mathbf{f} = \begin{pmatrix} 2 \\ 4 \\ -1 \end{pmatrix}$.

Hint: The result is in $\mathbb{R}^{2\times3}$. (2p)

ii. CNNs are known to have a weight-sharing property. Consider the vectorized form of the input $\mathbf{x}$ and express the convolution as a matrix-vector product $W\mathbf{x} = \mathbf{s}$, where the filter/kernel $\mathbf{f}$ is part of the weight matrix W. Convince yourself that the weight matrix is sparse[1] and weights are shared.
Hint: It holds that $\mathbf{x} \in \mathbb{R}^{12}$, $W \in \mathbb{R}^{6\times12}$, and $\mathbf{s} \in \mathbb{R}^{6}$. (3p)

(c) The softmax function is frequently used in multiclass classification as an activation function of the output layer. Let $\mathbf{x} \in \mathbb{R}^d$ be a vector. The softmax function softmax : $\mathbb{R}^d \to (0,1)^d$ is given by

$$\mathbf{p} = \mathrm{softmax}(\mathbf{x}) = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{pmatrix} = \begin{pmatrix} \exp(x_1) \\ \exp(x_2) \\ \vdots \\ \exp(x_d) \end{pmatrix} \Big/ \left(\sum_{j=1}^{d} \exp(x_j) \right)$$

and returns probabilities $\mathbf{p}$, i.e., $p_j \geq 0$ and $\sum_{j=1}^{d} p_j = 1$.
Show that the derivative of softmax with respect to $\mathbf{x}$ is

$$\frac{\partial p_j}{\partial x_i} = p_j(\delta_{ij} - p_i),$$

where δ_{ij} is 1 if $i = j$ and 0 otherwise.
Hint: Do a case distinction ($i = j$ and $i \neq j$) of $\frac{\partial p_j}{\partial x_i}$. (3p)

[1] In a sparse matrix most of the elements are zero.