EXAM IN STATISTICAL MACHINE LEARNING
STATISTISK MASKININLÄRNING

DATE: June 8, 2023

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

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

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

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 and each incorrect answer scores -1 point (capped at 0 for the whole problem). Answers left blank score 0 points.

   Hint: It is often better to only answer statements where you are confident. You do not need to classify all statements.

   i. In maximum likelihood we maximize the likelihood of the data with respect to the parameters.

   ii. Regularization usually decreases the training error.

   iii. k-nearest-neighbor is a linear classifier if $k = 1$.

   iv. In bootstrapping we randomly sample from a data set *without replacement*.

   v. Linear discriminant analysis (LDA) is a parametric model with M mean vectors $\mu_1, \ldots, \mu_M$ (one for each of the M classes) and one single covariance matrix Σ as its parameters.

   vi. Regression models have quantitative outputs.

   vii. The correlation between any pair of ensemble members of a bagged regression model tends to zero as the number of ensemble members tends to infinity.

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

   ix. The model $Y = \beta_0 + \beta_1 X_1 + \beta_0 \beta_1 X_2 + \varepsilon$ (where β_0 and β_1 are model parameters) is a linear regression model.

   x. The model bias typically tends to zero as the number of training data points tends to infinity.

   (10p)

2. Consider the following dataset with $n = 5$ samples of bell pepper with four attributes: `Domestic`, `Weight`, `Diameter`, and `Color`.

`Domestic`	`No`	`Yes`	`No`	`Yes`	`No`
`Weight (kg)`	`0.3`	`0.2`	`0.3`	`0.3`	`0.4`
`Diameter (cm)`	`6.5`	`5.2`	`5.8`	`6.1`	`6.9`
`Color`	`Green`	`Yellow`	`Yellow`	`Red`	`Red`

Based on this dataset, we want to train a binary classifier to predict the attribute `Domestic` based on the other three attributes. We consider a logistic regression model

$$\Pr(y = 1|\mathbf{x}; \beta) = \frac{e^{\beta^\top \mathbf{x}}}{1 + e^{\beta^\top \mathbf{x}}},$$

where the parameter vector β is estimated by maximizing the log-likelihood

$$\hat{\beta} = \arg\max \Big(\log \ell(\beta) \Big), \quad \text{where} \quad \ell(\beta) = \prod_{i=1}^{n} \Pr(y = y_i|\mathbf{x}_i; \beta).$$

(a) Which of the four attributes `Domestic`, `Weight`, `Diameter`, and `Color` are quantitative and qualitative, respectively? (1p)

(b) Provide numerical values of the training data $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ to be used in the formulas above. (2p)

Note: Multiple correct answers possible.

(c) Write down an explicit expression for the log-likelihood function $\log \ell(\beta)$ for the training data $\mathcal{T}$. (3p)

Note: You don't have to do the actual maximization, only provide an explicit expression of the function $\log \ell(\beta)$ *to be maximized. Also expand the product/sum.*

(d) Now consider a regression problem where we want to predict the `Weight` based on the two attributes `Domestic` and `Diameter`. We neglect the attribute `Color`. Assume that we want to use a linear regression model and optimize the average square error over our dataset. State the objective function $J(\theta)$ and compute the gradient $\frac{\partial J(\theta)}{\partial \theta}$ using the dataset and an initial $\theta = \mathbf{1}$, where $\mathbf{1}$ is a vector of ones. (4p)

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

x	1.75	2	2.5	2.75	3
y	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 losses across 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? (2p)

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

(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)

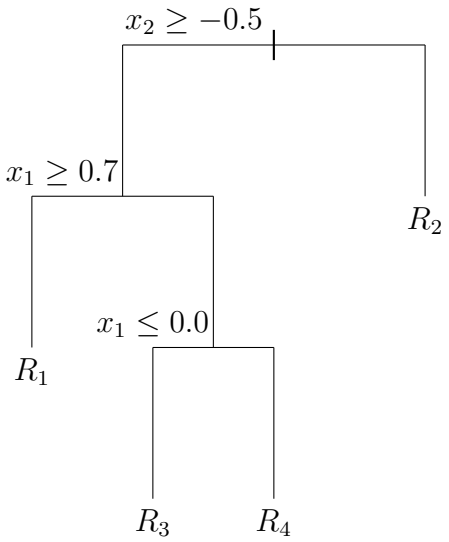

Figure 1: Regression tree of the input space for Problem 4

4. Consider a regression problem with a two input variables given by x_1 and x_2, and one output y. Alice has constructed a regression tree as shown in Figure 1 based on the following training data:

x_1	0.4	-0.8	0.8	0.2	0.6	0.8	0.1	-0.8
x_2	0.5	-0.1	0.2	-0.1	-0.8	-0.1	-0.3	-0.6
y	0.6	0.2	0.7	0.7	0.1	0.7	0.5	0.1

   (a) Draw the corresponding input partitioning to this tree. Mark the regions with the names of the leaf nodes, i.e., R_1, R_2, R_3, R_4. (2p)

   (b) Use the regression tree to predict the output of the test input $x^\star = [x_1^\star,\ x_2^\star]^\mathsf{T} = [0.5\ 0.2]^\mathsf{T}$ (3p)

   (c) Continue to grow the tree in Figure 1 such that there are at most two data points in each region by minimizing the mean-square-error. Which region(s) do you split and where? (3p)

   Note: There are multiple possible splits that are equally good.

   (d) Explain briefly (a couple of sentences) the disadvantage of growing a decision/regression tree too deep. (2p)

5. (a) In the context of neural networks, describe, using a few sentences, the difference between a fully-connected[1] layer and a convolutional layer. (2p)

   (b) Assume we have access to 784-dimensional data for which we want to predict one out of ten classes. For this multi-class classification task we build a neural network which has two fully-connected hidden layers consisting of 256 and 32 output neurons, respectively, followed by ReLU activation functions. In addition, we have a fully-connected output layer with ten output neurons that is followed by a softmax activation.

   Show that there are in total 209 514 parameters to learn. (3p)

   (c) Consider a dataset with $n = 100\,000$ data points for which we train a model with stochastic gradient descent where each mini-batch has the size $b = 100$. We run the algorithm for 10 epochs. How many iterations have been completed during training, i.e., how many times have the parameters in the network been updated? (2p)

   (d) What is the main advantage of stochastic gradient descent where $b \ll n$ in comparison to normal gradient descent where $b = n$? (1p)

   (e) In gradient descent, describe the impact of the learning rate γ to the learning. What happens if the learning rate is too low? What happens if the learning rate is too high? (2p)

[1] Other names for fully-connected layers are linear layers and dense layers.