EXAM IN STATISTICAL MACHINE LEARNING
STATISTISK MASKININLÄRNING

DATE: March 13, 2024

RESPONSIBLE TEACHER: Jens Sjölund

NUMBER OF PROBLEMS: 5

ALLOWED AIDS: 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 the 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

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

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

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

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:

$$\begin{aligned} \text{Misclassification error:} \quad & Q_\ell = 1 - \max_m \widehat{\pi}_{\ell m} \\ \text{Gini index:} \quad & Q_\ell = \sum_{m=1}^{M} \widehat{\pi}_{\ell m}(1 - \widehat{\pi}_{\ell m}) \\ \text{Entropy/deviance:} \quad & Q_\ell = -\sum_{m=1}^{M} \widehat{\pi}_{\ell m} \log \widehat{\pi}_{\ell m} \end{aligned}$$

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 $\widehat{y}(\mathbf{x}) = \operatorname{sign}\{C(\mathbf{x})\}$, 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,

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

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 problems where you are confident. You do not need to answer all questions.

   i. Boosting is only applicable to tree-based methods.

   ii. In expectation, stochastic gradient descent converges in fewer iterations than ordinary gradient descent.

   iii. If your model is underfitting, training on more data can be expected to improve performance.

   iv. Convolutional neural networks can be used for both regression and classification tasks.

   v. Using machine learning methods for credit risk assessments (e.g. approving a loan) is a way to guarantee fair and equal treatment of all applicants.

   vi. The more folds we use in cross validation, the better the hold-out validation error approximates the new data error.

   vii. A k-nearest neighbor classifier is a non-parametric model.

   viii. Assume we have an LDA model where predicting the most likely class takes 0.1 s. If we double the amount of training data and re-train the model, the prediction time also doubles.

   ix. Creating additional inputs using nonlinear transformations can improve the performance of a linear classifer.

   x. In bagging, you create multiple datasets by resampling inputs (features) with replacement.

(10p)

2. An aerospace company considers measurements of a material property ρ that is used for quantifying the durability of components used for building planes. Based on the value of ρ, they want to predict whether a component will withhold an array of stress tests, i.e., whether it is faulty or not. The following dataset has been gathered:

i	1	2	3	4	5	6	7	8	9	10	11	12
ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2
y	0	0	0	0	0	1	0	0	0	1	1	1

Here, i is an index and a value of $y = 1$ means that a fault has been detected, whereas $y = 0$ indicates a functioning component. A logistic regression model of the form

$$g(\mathbf{x}) = \frac{e^{\theta^T \mathbf{x}}}{1 + e^{\theta^T \mathbf{x}}}, \tag{1}$$

has been trained on the dataset. Here, $p(y = 1|\mathbf{x})$ is modelled by $g(\mathbf{x})$ and $\theta^T \mathbf{x} = \theta_0 + \theta_1 \rho$. The parameter values have been determined as $\theta_0 = -5.85$ and $\theta_1 = 6.81$. Predictions are made using the threshold $r = 0.5$.

(a) Determine the decision boundary $\rho_{\text{db}} \in \mathbb{R}$ of the classifier. (2p)

(b) Determine the confusion matrix and explain the meaning of its entries. Also give the misclassification rate. (2p)

(c) As even minor problems can have disastrous consequences in aviation, it is of the utmost importance that all components of a plane are functioning and durable. To improve safety, determine a value r' for the threshold such that the false negative rate becomes zero for the given dataset (while keeping the misclassification rate as low as possible). What happens to the false positive rate and what does that mean for the company producing the components? (2p)

(d) We now shift our attention to multiclass logistic regression with M classes, in which the class probabilities are modelled by the vector-valued function

$$\mathbf{g}(\mathbf{x}) = \mathbf{softmax}(\mathbf{z}), \tag{2}$$

where

$$\mathbf{softmax}(\mathbf{z}) = \frac{1}{\sum_{m=1}^{M} e^{z_m}} \begin{bmatrix} e^{z_1} \\ e^{z_2} \\ \vdots \\ e^{z_M} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \beta_1^T \mathbf{x} \\ \beta_2^T \mathbf{x} \\ \vdots \\ \beta_M^T \mathbf{x} \end{bmatrix}.$$

Suppose that we have learned the model above in a setting with $M = 2$ classes. The learned model has parameters $\beta_1^T = [\beta_{10}, \beta_{11}]$ and $\beta_2^T = [\beta_{20}, \beta_{21}]$. Show that binary logistic regression in Equation (1) is a special case of multiclass logistic regression as stated in Equation (2) by showing that there exists a parameter vector $\alpha^T = [\alpha_0, \alpha_1]$ for the binary logistic regression model that gives the same predicted class probabilities for inputs $\mathbf{x}^T = [1, \rho]$ as the multiclass logistic regression model with the learned parameter vectors β_1 and β_2. Determine α_0 and α_1.

Hint: Do the computed class probabilities change if we subtract a constant vector $\mathbf{c}$ from the parameter vectors in multiclass logistic regression?

(4p)

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

3. An RLC circuit is an electrical circuit consisting of a resistor (R), an inductor (L), and a capacitor (C). Radio receivers and television sets use them for tuning to select a narrow frequency range from ambient radio waves. Mathematically, any voltage or current in the circuit can be described by a second-order differential equation. In particular, the circuit forms a harmonic oscillator where the resistor dampens the oscillations.

By connecting a voltmeter to an RLC circuit, you record the following voltages v (in volts) at times t (in seconds):

i	1	2	3	4	5	6	7	8
t	0.4	0.9	1.9	3.0	3.8	4.1	5.3	6.0
v	9	3	-3	3	5	4	1	1

(a) Suppose you train a binary regression tree on all the data. The resulting predictions are shown in Figure 1. Sketch a regression tree of depth three that would produce these predictions. Clearly specify the cut points and the predictions for each leaf. Are there equivalent but shallower trees? (3p)

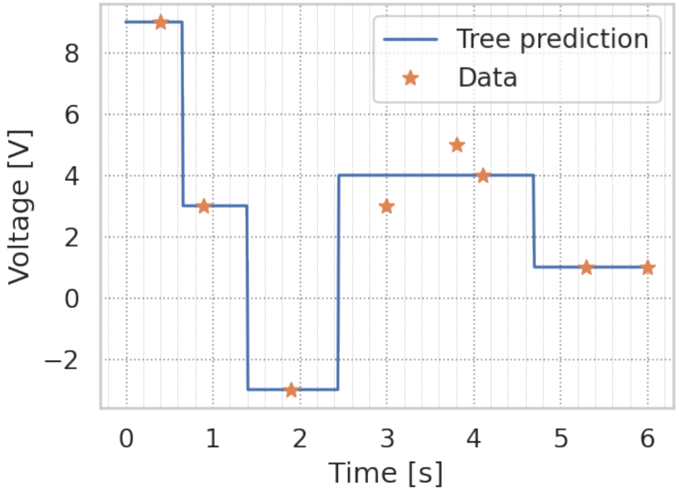


Figure 1: Predictions of the regression tree in question 3a.

(b) Let $i_{\text{train}} = \{1, 3, 5, 8\}$, $i_{\text{val}} = \{2, 6\}$, and $i_{\text{test}} = \{4, 7\}$ define the indices of the training, validation and test sets, respectively. Consider $k \in \{1, 2, 3\}$ and determine the best value for a k-nearest neighbor regressor in terms of the mean-squared error. Estimate the corresponding expected new data error. (4p)

(c) Your colleague, who is an experienced radio engineer, tells you that for this particular circuit the voltage exhibits underdamped oscillations according to the model

$$u(t) = 10\, e^{-t/3} \cos\left(\frac{\pi t}{2}\right).$$

However, the voltmeter has an unknown offset[1] $b \in \mathbb{R}$ in addition to the random measurement noise that can be considered independent and identically distributed Gaussian with mean zero. Help your colleague to calibrate the voltmeter by determining b using maximum likelihood. Use the entire dataset. (3p)

Note: Questions 3(a)-3(c) can be solved independently.

[1] a constant scalar error

4. (a) We want to predict if there will be a flood in a certain city ($y = 1$) or not ($y = -1$) based on the rainfall in the last 24 hours (x_1) and the city's sewer system drainage capacity (x_2). Given the data in Table 1, compute the parameters for the LDA model. (3p)

index	x_1	x_2	y_i
1	20	1.7	-1
2	50	0.8	1
3	30	1.2	-1
4	40	0.6	1
5	10	1.0	-1

Table 1: Training data for the LDA model.

(b) Generative models can be used to generate new datapoints after the training since they model the joint probability distribution. Explain how using the LDA model a new sample can be obtained. *You do not have to compute an actual sample in this task but only explain how one can be obtained.* (2p)

(c) LDA is a generative model. Describe how this is different from discriminative models. (1p)

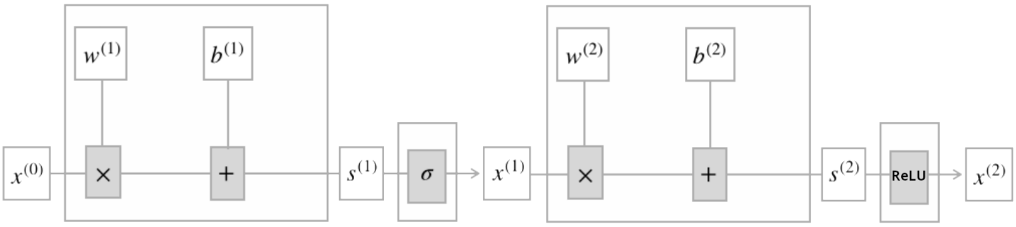
(d) In LDA we assume all classes share the same variance matrix, i.e., $\hat{\Sigma} \overset{\text{def}}{=} \hat{\Sigma}_1 = \cdots = \hat{\Sigma}_M$, where M is the number of classes. Show that the most probable prediction $\hat{y}_\star = \arg\max_m \{p(y = m|\mathbf{x})\}$ is given by

$$\begin{aligned}\hat{y}_\star &= \arg\max_m \{p(y = m|\mathbf{x})\} \\ &= \arg\max_m \{-\frac{1}{2}(\mathbf{x} - \hat{\mu}_m)^\mathsf{T}\hat{\Sigma}^{-1}(\mathbf{x} - \hat{\mu}_m) + \log\hat{\pi}_m\}.\end{aligned}$$

Hint: 1. Check the discriminant analysis, Gaussian distribution, and maximum likelihood in the formula sheet; 2. The denominator in $p(y = m|\mathbf{x})$ can be ignored; 3. You do not need to insert actual numbers, just derive the formula. (4p)

Note: Questions 4(a)-4(d) can be solved independently.

5. (a) Consider the following neural network



consisting of (i) a two-dimensional input $x^{(0)}$, (ii) a hidden layer with three neurons whose value before the sigmoid activation σ is $s^{(1)}$ and after the activation σ is $x^{(1)}$, and (iii) an output layer with one neuron and a ReLU activation.

Mathematically, we can write the relationship between input $x^{(0)}$ and output $x^{(2)}$ as

$$x^{(2)} = \text{ReLU}\left(w^{(2)}\sigma\left(w^{(1)}x^{(0)} + b^{(1)}\right) + b^{(2)}\right),$$

where the parameters are

$$w^{(1)} = \begin{pmatrix} 0.5 & -1.0 \\ 0.1 & 1.5 \\ -2.0 & 0.2 \end{pmatrix}, \qquad b^{(1)} = \begin{pmatrix} -2.5 \\ 9.6 \\ -0.4 \end{pmatrix},$$
$$w^{(2)} = \begin{pmatrix} 1.2 & -2.0 & 4.2 \end{pmatrix}, \qquad b^{(2)} = \begin{pmatrix} 0.2 \end{pmatrix}.$$

The first activation function is a sigmoid, given by

$$\sigma(z) = \frac{1}{1+\exp(-z)}$$

and the second activation function is a ReLU, given by

$$\text{ReLU}(z) = \begin{cases} z, & \text{if } z \geq 0, \\ 0, & \text{if } z < 0. \end{cases}$$

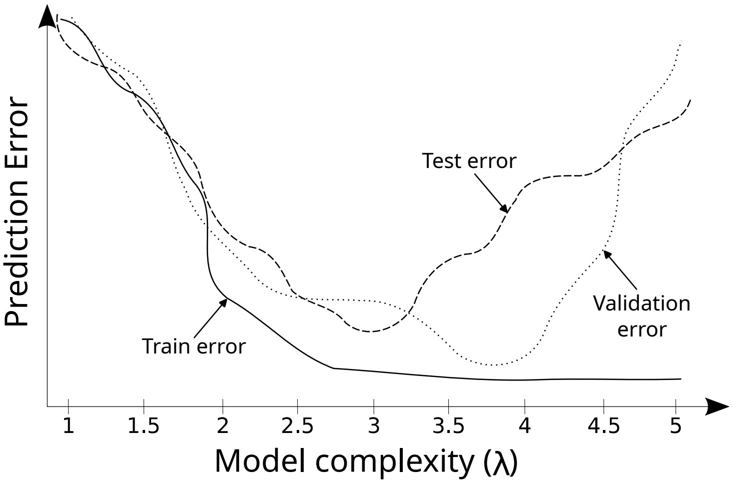
Note that the activation functions are applied elementwise. Let the input be $x^{(0)} = \begin{pmatrix} 4.0 \\ -2.0 \end{pmatrix}$.

i. Is this neural network rather modeling a regression or classification task? Why? (1p)

ii. Compute and provide the prediction $x^{(2)}$ as well as the intermediate results $s^{(1)}$, $x^{(1)}$, and $s^{(2)}$. Round every result to two digits after the decimal. (3p)

iii. Suppose we want to train this neural network using stochastic gradient descent. Name three hyperparameters. (2p)

(b) Suppose we trained several neural networks with different model complexities. The following figure shows a curve of the prediction error on the training, validation, and test sets under different complexities, measured as some quantity λ. The prediction errors in the training set decrease as the model complexity λ increases.



i. Which model (in terms of λ) would you choose and why? (1p)

ii. For which values of λ does underfitting and overfitting occur? Motivate. (2p)

iii. Suppose a fixed test set was given to you and you split the available data into 80% train and 20% validation data yourself. Your colleague did the same but according to her split, a significantly different λ is optimal. She conjectures that it may depend on the seed that was used to do the split. What could you do to alleviate this effect? (1p)

Note: All sub-questions can be solved independently.