Exercises

1	2	3	4	5

Surname, First name

____

Machine Learning 1 (52041MAL6Y)
Resit Exam

○1	○1	○1	○1	○1	○1	○1	○1
○2	○2	○2	○2	○2	○2	○2	○2
○3	○3	○3	○3	○3	○3	○3	○3
○4	○4	○4	○4	○4	○4	○4	○4
○5	○5	○5	○5	○5	○5	○5	○5
○6	○6	○6	○6	○6	○6	○6	○6
○7	○7	○7	○7	○7	○7	○7	○7
○8	○8	○8	○8	○8	○8	○8	○8
○9	○9	○9	○9	○9	○9	○9	○9
○0	○0	○0	○0	○0	○0	○0	○0

Resit Exam for Machine Learning 1 - 10 January 2024

Pay attention to the following

- **Write your name and student number on the front page** (don't forget to mark the digits)
- Write all your answers on this exam booklet; it will be scanned and digitally graded.
- Write your **answers inside the boxes** (it is ok, to slightly go over the margins though).
- The answer boxes should be large enough for your answer
- **If you need to empty the answer box in order to start over, ask the invigilator for a blank sticker**

During the exam you are allowed to use:

- One double-sided handwritten cheat sheet.

About the multiple choice questions:

- You should fill the boxes and not just check them. E.g.

Correct marking: ■ ● Incorrect marking: ☒ ☑ ⊗ ✓

- In case you want to correct your answer, clear indicate this (e.g. by filling all boxes and use another indicator such as an arrow to indicate your choice). We can then resolve this during grading.
- In this exam, *there can only be one correct answer* per multiple choice question.

About the open questions:

- We work with a partial grading: you can get points even if you don't manage to solve the full question.
- Keep your answers to open-ended questions short and to the point!

Please scan over the questions before you start to get an impression of the content

- In total **47.5 points** can be earned divided over the following 5 categories.
- You have **3 hours** for the exam (except for those with pre-approved extensions)

- Exercises 1: Multiple choice (8.0 pts)
- Exercise 2: Modeling Categorical data (10.5 pts)
- Exercise 3: Principal Component Analysis (8 pts)
- Exercise 4: Support Vector Regression (8.5 pts)
- Exercise 5: Neural Networks (12.5 pts)

Multiple Choice

1p **1a** Which of the following statements about linear models is **incorrect**?

ⓐ Linear discriminant analysis and logistic regression have the same expressive power (i.e. in the types of decision boundaries that they can represent).

ⓑ For logistic regression, the decision boundary lies perpendicular to the parameter vector.

ⓒ Logistic regression models the difference in the class probabilities, i.e. $P(C_1|X) - P(C_0|X)$, where C_k denotes the class and X the features.

ⓓ Quadratic discriminant analysis can represent all possible decision boundaries that linear discriminant analysis can represent.

1p **1b** Consider building a Bayesian predictive model: let X denote the features, t the label, θ the parameters, $p(\theta)$ the prior, and $p(t|X,\theta)$ the likelihood. Which of the following quantities does the Bayesian framework *ideally* use to make predictions on test data, denoted X^* and t^*?

ⓐ the posterior mode (a.k.a. MAP estimator): $p(t^*|X^*, \hat{\theta}_{MAP})$

ⓑ the maximum likelihood estimator (MLE): $p(t^*|X^*, \hat{\theta}_{MLE})$

ⓒ the posterior predictive distribution: $p(t^*|X^*, t, X) = \int_\theta p(t^*|X^*, \theta) \cdot p(\theta|t, X) d\theta$

ⓓ the marginal likelihood: $p(t^*|X^*) = \int_\theta p(t^*|X^*, \theta) \cdot p(\theta) d\theta$

1p **1c** Which of the following statements about Gaussian mixture models (GMMs) is **incorrect**?

ⓐ Fitting a GMM with the EM algorithm usually returns the globally optimal parameter setting.

ⓑ GMMs are a more flexible clustering model / algorithm than k-means, meaning that they can represent clustering configurations that k-means cannot.

ⓒ GMMs assume that each data point is drawn from one of K Gaussian distributions (where K is the total number of mixture components).

ⓓ A GMM represents a distribution that is at least as expressive as any single one of its component distributions.

1p **1d** The kernel trick is a key concept used in support vector machines to solve non-linear problems. How does it achieve this?

ⓐ By reducing the dimensionality of the feature space, making it easier to find a linear separator.

ⓑ By transforming the original finite-dimensional space into a higher-dimensional space, potentially making the data separable by a hyperplane.

ⓒ By calculating the convex hulls for classes in the feature space and finding the linear separators between them.

ⓓ By applying stochastic gradient descent in the primal space to ensure non-linearity in the decision boundary.

1p **1e** Which of the following statements about kernel methods is **true**?

ⓐ A kernel always operates in an infinite-dimensional space.

ⓑ Kernel methods can only be applied to SVMs.

ⓒ Using more complex kernels always results in better model performance.

ⓓ Kernel methods allow SVMs to classify non-linearly separable data.

1p **1f** Which of the following statements about *the bootstrap* algorithm is **true**?

ⓐ The bootstrap takes weak models and ensembles them to achieve strong aggregate performance.

ⓑ The bootstrap aims to simulate the noise / variation introduced when the original data set was sampled from the underlying population.

ⓒ The bootstrap creates new data sets that are smaller in size (i.e. number of data points) to improve computational efficiency.

ⓓ The bootstrap should never be used with high-dimensional data as it will likely result in overfitting.

1p **1g** Which of the following models is **not** a generative model?

ⓐ Probabilistic PCA

ⓑ Quadratic Discriminant Analysis

ⓒ Gaussian Mixture Model

ⓓ Decision Tree

1p **1h** You find that your decision tree is overfitting. Which of the following steps would best help reduce the overfitting? (choose just one answer)

ⓐ Prune the decision tree so that its depth is decreased.

ⓑ Prune the decision tree so that its width is decreased.

ⓒ Remove feature bagging.

ⓓ Re-fit the tree and do not use information gain as the splitting heuristic.

Modelling Categorical Data

Imagine a renowned casino that has just unveiled its latest slot machine. On each pull, the machine produces one of the K distinct outcomes. Although the casino claims that each of the K outcomes is equally likely, your goal in this exercise is to test this assertion. Having recently completed a Machine Learning 1 course, you begin by collecting a dataset of independent pulls, $\mathcal{D} = \{x_1, \ldots, x_N\}$, where each $x_n \in \{1, \ldots, K\}$. You then assume a categorical distribution to model the outcomes of the slot machine:

$$p(x_n = k|\boldsymbol{\theta}) = \theta_k$$

where we assume $1 \geq \theta_k \geq 0,\ \forall k$ (otherwise the likelihood is not defined). Note that since only one of the K events can be observed for each pull, the parameter vector $\boldsymbol{\theta}$ must sum to one: $\sum_{k=1}^{K} \theta_k = 1$.

4p **2a** Derive the maximum likelihood estimate $\boldsymbol{\theta}_{MLE}$ based on the dataset $\mathcal{D}$. *Hint:* Do not forget to take the constraint on the model parameters into consideration.

1.5p **2b** As an experienced modeler, you know that you cannot always rely on the MLE, especially in small-data settings. Therefore, you decide to assume a prior distribution over $\boldsymbol{\theta}$. The Dirichlet distribution

$$\mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

is a natural choice as its support corresponds to K-dimensional vectors satisfying the constraint specified above. Here $\alpha_k > 0$ are the so-called concentration parameter. The normalization constant $B(\boldsymbol{\alpha})$ is a multivariate beta function.

Derive the log-posterior. You do not have to provide explicit expression for $B(\boldsymbol{\alpha})$ or any other terms that do not depend on $\boldsymbol{\theta}$. That is, you can treat them as a constant.

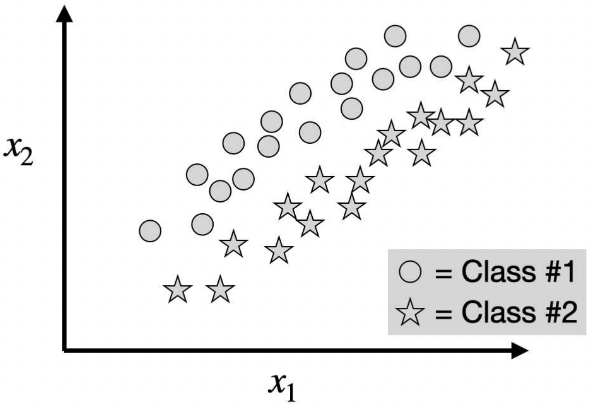
3p **2c** Find the maximum a posteriori estimate θ_{MAP}. *Hint:* Do not forget to take the constraint on the model parameters into consideration.

1p **2d** For what value of concentration parameters α, will the resulting θ_{MAP} estimator reduce back to the maximum-likelihood estimator θ_{MLE} ?

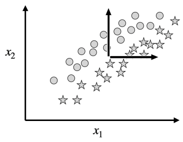
1p **2e** A friend of yours, who works at the casino, discreetly told you that he heard a rumour about even outcomes ($k = 2, 4, \ldots$) being more likely than odd outcomes ($k = 1, 3, \ldots$). How would you incorporate this additional information into your model of the slot machine's behaviour?

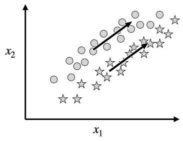
Principal Component Analysis (PCA)

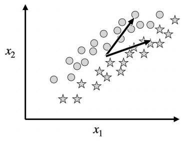
Consider applying ***principal component analysis*** **(PCA)** to the following two-dimensional data set, which has two classes.

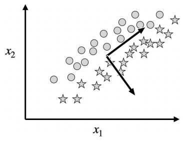


2p **3a** Which of the following images shows the components (i.e. the eigenvectors) for this data set (from which we would select the principal one)?

a


b


c


d


2p **3b** Assume that the eigenvalues for the two components are $\lambda_1 = 7/10$ and $\lambda_2 = 1/10$. What *fraction* of the data's variance will be retained if we used PCA to reduce the data's dimensionality down to one dimension?

a 7/10 b 7/8 c 1/10 d 1/8 e 8/10 f 2/8

1p **3c** Assume we are interested in training a classifier on this data set. Should we **(A)** apply PCA to the data first (to reduce it down to one feature dimension), or **(B)** leave the data as is (in two dimensions)?

a **A**: Apply PCA, reducing the data to one feature dimension

b **B**: Don't apply PCA, leave the data in two dimensions, as is shown above.

1p **3d** Give your reasoning behind your answer to the preceding question (for choosing A vs B).

2p **3e** Let X represent a $(N \times D)$-matrix of data with N observations and D features. Let W be a $(D \times K)$-semi-orthogonal matrix, with $D > K$, meaning that W's columns are orthonormal vectors.

Write down a loss function in terms of X and W such that it, when minimized with respect to W, is equivalent to fitting a PCA model with K components. Assume that your optimizer maintains the semi-orthogonality property of W. You may also assume the data is zero centered, i.e., $\sum_{n=1}^{N} \mathbf{x}_n = \mathbf{0}$ with $\mathbf{x}_n$ being the rows of X.

Support Vector Regression

You recently learned about Support Vector Machines for classification and are curious about the regression setting. Assume a dataset $\mathcal{D} = \{(\mathbf{x}_1, t_1) \dots, (\mathbf{x}_N, t_n)\}$, where $\mathbf{x}_n \in \mathbb{R}^D$ and $t_n \in \mathbb{R}$. The regression prediction is given by $y(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n + b$ with $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$.

Consider further, the ϵ-insensitive cost function $l_\epsilon(y(\mathbf{x}), t) = \max(0, |y(\mathbf{x}) - t| - \epsilon)$. This cost function is 0 if the absolute difference between prediction and target is smaller then ϵ. Using this cost function, we can now formulate the regression task as a constraint optimization problem. To soften the assumptions and allow for errors, we introduce slack variables $\{\xi_n\}$ and $\{\xi_n^*\}$.

We will now state the primal problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*) \quad \text{subject to} \quad \begin{cases} \forall n : (\mathbf{w}^T\mathbf{x}_n - b) - t_n \leq \epsilon + \xi_n \\ \forall n : t_n - (\mathbf{w}^T\mathbf{x}_n - b) \leq \epsilon + \xi_n^* \\ \forall n : \xi_n \geq 0 \\ \forall n : \xi_n^* \geq 0 \end{cases}$$

1.5p **4a** Introduce Lagrange multipliers for the constraints and write down the primal Lagrangian. Use the following notation: $\{\alpha_i\}$ and $\{\alpha_i^*\}$ are the Lagrange multipliers for the first two constraints and $\{\mu_i\}$ and $\{\mu_i^*\}$ for the last two constraints.

1.5p **4b** Write down all KKT conditions.

2p **4c** Derive the stationary conditions (by computing $\partial \ell / \partial \rho = 0$, where ℓ is the primal Lagrangian and ρ is a primal variable).

1.5p **4d** Define the dual Lagrangian (no need to derive it) and explain how you could obtain it using your results from (c).

1p **4e** Suppose we work with high dimensional features (large D) and notice that the regression model is very flexible / expressive, in the sense that it could easily overfit. We can control how well the model generalizes by tuning C. *Mark the correct answer*: In order to *prevent overfitting*, C should be

ⓐ large ⓑ small

1p **4f** Give your reasoning behind your answer to the preceding question (for increasing or decreasing C).

Neural Networks

Consider a neural network with two hidden layers and a skip connection (aka residual connection):

$$\begin{aligned} f(\mathbf{x}; \mathbf{w}_0, w_1, w_2) &= w_2 \cdot h_2 + h_1 \\ h_2 &= \sigma(w_1 \cdot h_1) \\ h_1 &= \sigma(\mathbf{w}_0^T \mathbf{x}) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^D$ is a D-length (column) vector, $\mathbf{w}_0 \in \mathbb{R}^D$ are the first-layer weights, $w_1 \in \mathbb{R}$ is the second-layer weight, and $w_2 \in \mathbb{R}$ is the hidden-to-output weight.

2p **5a** Write down how the chain rule is implemented to compute the derivative $\partial f / \partial \mathbf{w}_0$.. *Show all partial derivatives involved to the finest granularity allowed*.

For example, if $g(a; b, c) = (a \cdot b)/c$, then $\partial g / \partial a = (\partial g / \partial (a \cdot b))(\partial (a \cdot b)/\partial a)$. Feel free to define any intermediate computations, such as $a \cdot b$ in the example, with a variable.

2p **5b** Based on your prevoius answer, give the exact expression for the derivative $\partial f/\partial w_{0j}$, where w_{0j} denotes the jth component of $\mathbf{w}_0$, by evaluating all partial derivatives. Let $\sigma(\cdot)$ denote a logistic activation function

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

and assume that the activations are logistic functions from this question forward.

To return to the above example, the derivative explicitly gives $\partial g/\partial a = (1/c)(b) = b/c$.

2p **5c** Write down the loss function for performing *ridge regression* using this neural network. Assume we observe a training dataset containing features $\{\mathbf{x}_n\}_{n=1}^N$, with $\mathbf{x}_n \in \mathbb{R}^D$, and corresponding responses $\{t_n\}_{n=1}^N$, with $t_n \in \mathbb{R}$, where N is the number of data points and D is the feature dimensionality.

3p **5d** The ridge regression loss could also be derived from the Bayesian modeling principles. The Bayesian modeling viewpoint entails that we solve our problem using prior beliefs we might have in our problem. To obtain ridge regression from this point of view, 1) *what* kind of distribution should the neural network (NN) parametrize, and 2) *how* should the NN parametrize it? Additionally, 3) are there any other distributions that need to be modeled? 4) How then is the ridge regression loss obtained this probabilistic model?

1p **5e** Is the ridge regression loss for the described neural network convex with respect to its parameters?

Ⓐ Yes Ⓑ No

1p **5f** Consider making w_2 a constant such that its value can *only* be zero: $w_2 = 0$. In other words, gradient descent would not change its value from zero. Under this assumption, to which of the following models is this neural network now equivalent?:

- ⓐ Linear regression
- ⓑ Kernel regression with the kernel determined by the neural network's hidden units.
- ⓒ Logistic regression
- ⓓ Linear classifier with 3 or more classes

1.5p **5g** Neural networks are typically optimized using stochastic gradient descent (SGD), as opposed to full-batch gradient descent (GD). Give two advantages of using SGD vs full-batch GD.