UiT

THE ARCTIC UNIVERSITY OF NORWAY

Assignment 2. Due: Sunday 05.10.2025 23:59

FYS-2021 Assignment
Department of Physics and Technology
Faculty of Science and Technology

Guidelines

The assignment is mandatory. You get a pass/fail grade. If you fail it you can't come to the exam and you fail the entire course. To pass, you need to write a short report (pdf file) where you provide your answers to the questions asked, and upload it on Canvas. You need to answer correctly at least 50% of the questions.

Additional important guidelines

Learning to write a scientific report is an important skill that many of the courses at the Faculty of Science and Technology, including this one, aim to improve. Therefore any question that you answer should be contained within the report of this assignment. Answers outside of the written report, (e.g. in the comment of the code, or within a Jupyter Notebook), will not be considered as a part of your answer of the problem. You can structure your report by having a separate (sub)section with the answer for each question. The report should contain text, figures, excerpts of code and anything that helps you answering the questions. The report and code should be your own individual work. Remember to cite all sources.

Make sure your report shows that you understand what you are doing. More specifically, it is important to elaborate your answers such that essential theory, equations, and intuition is included in your answers. However, your answers should still remain concise and stay focused on the core problem, e.g. there is no need to derive or prove an equation unless the problem asks you to.

Problems that ask for numeric values or plots should include these in the answer of the report. Tables and figures should always have captions with a short and concise description of what is shown. Additionally each table and figure should be addressed and referenced in the main text body. The code should be commented in such a way that any person with programming knowledge should be able to understand how the program works. Like your report, the code must be your own individual work.

You are permitted to use standard built-in functions and/or packages (e.g. numpy, pandas and matplotlib in Python) for reading the data and basic calculations. However: make sure that the packages you use do not over simplify your implementation (using Scikit-Learn is not permitted!). Of course, all implementations asked for in the problems should be your own work.

Submission

For your final submission, upload the report as a single `.pdf` file on Canvas under 'Assignments' > 'Mandatory assignment 1' by the announced submission due. Make sure you have your name written on the report.The naming convention for the file should be `assignment1_firstname_lastname.pdf`

Plagiarism

Plagiarism is a serious academic offense and will not be tolerated. Always ensure that you provide proper attribution for any work, ideas, or concepts that are not originally yours. Should the Canvas plagiarism detection system flag your submission, your report will not be considered for assessment.

Resources

All datasets required to answer the exercises can be found in the Canvas room for the course.

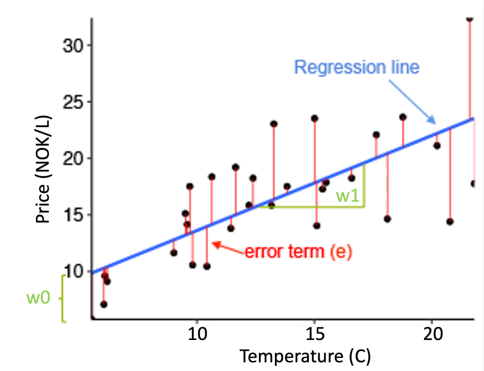
Problem 1

Throughout the lectures and accompanying exercises of the course, we have seen the standard analytical solution for the minimum of the linear regression loss function. In this problem, however, we utilize the gradient descent algorithm to search the optimal values for these weights.

The graph displayed below depicts how the annual global average temperature, which we will refer to as x, correlates with oil prices, denoted as y. We can model this relationship mathematically using a linear regression equation, expressed as:

$$\hat{y} = w_1 x + w_0.$$

In this equation, $\hat{y}$ is the predicted oil price, w_1 is the slope of the line, and w_0 is the y-intercept. Note: in this problem there is only one feature, so x is a scalar, not a vector.



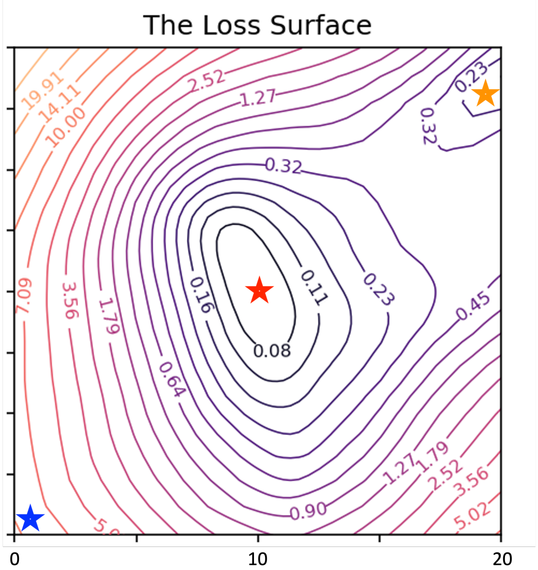
(1a) The choice of loss function plays a critical role in learning models. Discuss the most common type of loss function that is typically used for linear regression. Your answer should include:

- Your choice of the loss function $\mathcal{L}_{reg}(\hat{y}, y)$ and its mathematical formula.
- Derivation of the gradient of this loss function with respect to each weight parameter, i.e., $\frac{\partial \mathcal{L}_{reg}}{\partial w_1}$ and $\frac{\partial \mathcal{L}_{reg}}{\partial w_0}$. **Hint:** Use chain rule.

(1b) Explain why smooth loss functions are preferred when using gradient descent. Elaborate on the potential issues that could arise when using mean absolute error (MAE) as a loss function, defined as $\mathcal{L}_{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$. **Hint:** Start from the gradient of MAE, e.g., $\frac{\partial \mathcal{L}_{MAE}}{\partial w_1}$.

(1c) Elaborate on how the weights can be optimised using gradient descent. Your answer should include the step-by-step process of the gradient descent, **at least** encompassing weight initialization, the formula for iterative updates with each variable explicitly stated, and termination conditions. You may use pseudo-code or bullet points to show the different steps.

The figure below shows the loss surface, denoted as $\mathcal{L}$, in the weight space. The weight w_0 is on the horizontal axis ranging (0, 20), the weight w_1 is on the vertical axis ranging (0, 2), and the losses, $l = \mathcal{L}(w_0, w_1)$, are represented as a set of contour lines. Assume the initial weights are $w_0^{(0)} = 0.01$ and $w_1^{(0)} = 0.01$, marked by a blue star in the left-bottom corner of the figure. Additionally, the global minimum of the loss function is achieved at the optimal weight values, $w_0^{(*)} = 10$ and $w_1^{(*)} = 1$, where a red star is indicated.



(1d) Draw approximately the iterative learning process by gradient descent on the loss surface. That is, depict the trajectory of the weight values, which gradually move from their initial states to their optimal values. We assume a sufficiently small learning rate. The gradient's direction should be illustrated with an arrow per step. No computations are required here. **Note:** The figure is provided as a separate file in canvas, so you can use it to draw the gradient steps and include it in your report.

(1e) Elaborate on how you derive the trajectory in Problem (1d), grounded by the theoretical aspects outlined in Problems (1a)-(1c). **Hint:** Place emphasis on the graphical interpretation of the gradient and the learning rate.

(1f) Draw another trajectory to illustrate the iterative learning process. In this case, the weights fail to converge to the optimal values because of an excessively large learning rate. Include a visualization with **10 updates** of the weights. Draw your trajectory with arrows on the figure and embed it in your report.

(1g) During the optimization process, the weights may get stuck in a local minimum, which is represented by an orange star on the loss surface. To address this issue, suggest **at least one strategy** that can solve this problem. Explain each of your suggestions in plain language and avoid using any mathematical formulas.

Problem 2

(2a) Data is appended as `data_problem2.csv`. Load the data and report general information of the data. Additionally plot (as histograms) the data and discuss the separability.

(2b) Lets assume that data from class $\mathcal{C}_0$ follows a Gamma distribution:

$$p(x|\mathcal{C}_0) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}$$

and that data from class $\mathcal{C}_1$ follows a Gaussian distribution:

$$p(x|\mathcal{C}_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

The Gamma distribution has the parameters α and β, in this case α is known: $\alpha = 2$, but β is unknown. The Gaussian distribution has the parameters μ and σ where both are unknown.

Remember that the Gamma *function* for $n \in \{0, 1, 2, ...\}$ can be computed as:

$$\Gamma(n) = (n-1)!$$

Show that the maximum likelihood estimations of the parameters are:

$$\hat{\beta} = \frac{1}{n_0 \alpha} \sum_{j=1}^{n_0} x_0^j, \quad \hat{\mu} = \frac{1}{n_1} \sum_{j=1}^{n_1} x_1^j, \quad \hat{\sigma}^2 = \frac{1}{n_1} \sum_{j=1}^{n_1} (x_1^j - \hat{\mu})^2$$

where $x_0^1, ..., x_0^{n_0}$ are the training samples from $\mathcal{C}_0$ and $x_1^1, ..., x_1^{n_1}$ are the training samples from $\mathcal{C}_1$.

(2c) Split the data into training and test data. Use the maximum likelihood estimations to estimate the parameters based on the training data. Use the point-estimations of the parameters to implement a Bayes' classifier. Report the test accuracy.

(2d) Explain why the Bayes' classifier minimizes the probability of miss-classification when the probability distribution of the data is known.

Show the missclassified data in a plot along with the rest of the data and explain why it was miss-classified. Does it follow the conclusion in *a*)?