FYS-2021 – Mandatory Assignment #1, Autumn 2022

September 15, 2022

Introduction

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 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.

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* 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! Of course, all implementations asked for in the problems should be your own work.

Hand-in format

Your report must be a single compressed `.zip` file containing the report as a single `.pdf` file and the code. The file name *has to* follow the format `assignment1_XX.zip` (replace `XX` with your ***name*** (`firstname_lastname`)). The `.pdf` file should follow the same naming convention as the `.zip` file. Do not put your candidate number in the report or code. Failure to do this may compromise your anonymity for the home exam and exam as we will be able to see the name in the Canvas submission. Be advised that the name of the files are visible to the reviewers.

The code you write for this assignment should be included *both* in the appendix of the report *and* in the `.zip` file.

Resources

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

Problem 1

In a regression problem, the goal is to learn how to predict a continuous value r (response) based on some measured values $\mathbf{x}$. The standard model for this prediction is given by

$$r = f(\mathbf{x}) + \epsilon,$$

where $f(\mathbf{x})$ is an unknown function. In order to predict responses, we try to *estimate* $f(\mathbf{x})$ using a function $g(\mathbf{x}|\boldsymbol{\theta})$.

(1a) Explain (briefly) all the different components in this model and which assumptions we make in linear regression.

(1b) What are the learnable parameters in our model and how do we find them?

When training models of this type, it is common to split the dataset into a training set and a test set, where the training set is used to learn parameters in the model and the test set is used to evaluate the performance of the model. In the real world, datasets containing ground truth tend to be quite small. In this case, it might be a good idea to utilize a so–called leave–one–out cross–validation scheme to evaluate our model. The idea is quite simple. Given a dataset of size N with inputs $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N$ and responses $r^1, r^2, \ldots, r^N$, you train a model using $N-1$ datapoints and predict the response of the remaining datapoint (the one left out) using the trained model. This is done iteratively, such that you end up with predictions for all datapoints in your dataset, but each prediction is done by a model trained on the other $N-1$ datapoints. These predictions are then used to calculate some metric to evaluate performance.[1]

In this problem, we want to predict *popularity* of songs using linear regression. The data file `spotify_data.csv` contains 9 features extracted from $N = 50$ different songs, along with a popularity rating[2]. These features are as follows: *BPM, energy, danceability, loudness, liveness, valence, length, acousticness* and *speechiness*. The first column in the CSV file contains the popularity rating, which we will treat as our responses. You can find additional metadata (song name, artist and genre) in `spotify_metadata.csv`. These are not needed for the analysis, but are made available for those who are interested.

(1c) Use linear (multivariate) regression in order to predict the popularity of songs in the **spotify** dataset. You need to implement the regression algorithm yourself and use leave–one–out cross–validation. Make a plot containing both your predictions $\hat{r}$ (estimated responses) and the ground truth responses r as a function of the song index.

Note: When studying a plot like this, results might look worse than they are. This is due to the range on the y-axis being constrained to the min/max values.

(1d) Plot a histogram of the residuals/errors $e^t = r^t - \hat{r}^t$ and compute RMSE (root mean squared error) and R^2. Comment on the results. Is the model performing well?

(1e) Most of the songs in the dataset have `popularity > 80`. What happens if we remove all datapoints with `popularity < 80` from our dataset?

[1] You can read more about cross–validation in the book, chapter 19 (20 in the fourth edition).
[2] The dataset was sourced from Kaggle: https://www.kaggle.com/leonardopena/top50spotify2019/

Problem 2

In this problem, we will design a classification system for classifying handwritten 0's an 1's. The file `optdigits-1d-train.csv` contains one–dimensional linear projections[3] of the original images, along with their respective labels. Each row is formatted as `<label,x>`.

By looking at a histogram of the data, it is tempting to assume that the projected 0's ($\mathcal{C}_0$) follow a Gamma distribution

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

with $\alpha = 9$ and β unknown. On the other hand, the projected 1's ($\mathcal{C}_1$) looks like they might follow a normal distribution:

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

where μ and σ are unknown.

The gamma *function* is defined as

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} \, \mathrm{d}t$$

with the following properties:

$$\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$$

And for $n \in (0, 1, 2, \dots)$:

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

(2a) Show that the Maximum Likelihood estimators for the unknown parameters are

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

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

(2b) Use the training set and the expressions from (2a) to calculate the point–estimates $\hat{\beta}$, $\hat{\mu}$ and $\hat{\sigma}$, as well as the prior probabilities $P(\mathcal{C}_0)$ and $P(\mathcal{C}_1)$. Plot the histograms of the training observations and their corresponding estimated distributions. Are the distributional assumptions reasonable?

(2c) Use the expressions from (2a) to implement your own Bayes' classifier. Report the confusion matrix when evaluating your classifier on the training set. Use the confusion matrix to calculate the accuracy, precision and recall. Why is everything not correctly classified?

The file `optdigits-1d-test.csv` contains projected images forming a binary–encoded secret message. Your task is to decode this message using your classification system.

(2d) Load the test set and classify each entry. When you have the binary label array, you can use the function `get_msg_for_labels` from the file `assignment1_util.py` (or `get_msg_for_labels.m` for MATLAB), to convert the array to a text-string. What is the secret message?

[3] The projections were obtained using Principal Component Analysis.