COMP8043 - MACHINE LEARNING

ASSIGNMENT 3 - Regression & optimisation

Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

SUBMISSION & DUE DATE

This assignment should be submitted to Canvas before 11:59pm on **Friday 06/12/2024**. The usual late submission penalties apply in accordance with MTU's marks and standards policy.

Please submit a single ZIP file with your student number and name in the filename. Your submission should contain **exactly 2 files**:

- A detailed documentation of all code you developed, including the tests and evaluations you carried out. Please make sure that you include a .pdf document with every result you produce referencing the exact subtask and lines of code it refers to.
- All Python code you developed in a single .py file that can be executed and that generates the outputs you are referring to in your evaluation. The file needs to be readable in a plain text editor, please do NOT submit a notebook file or link. Please also make sure that you clearly indicate in your comments the exact subtask every piece of code is referring to.

Please do NOT include the input files in your submission.

EVALUATION PROCEDURE

You can achieve a total of 35 points as indicated in the tasks. For each subtask you are given full marks for correct answers in your submission, 70% for minor mistakes, and 35%, 15%, or 0% for major mistakes or omissions depending on severity.

Most tasks ask you for explanations, so it is vital for this assignment that you always explicitly answer **why** your implementation achieves the intended results.

OBJECTIVE

The goal of this assignment is to train and evaluate a regression function that can be used to predict the expected thermal loads of a building based on a selection of input variables that have been identified in the following publication:

> *Athanasios Tsanas, Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy and Buildings, Volume 49, 2012, Pages 560-567, ISSN 0378-7788, https://doi.org/10.1016/j.enbuild.2012.03.003.*

(the full paper can be accessed through the library when on the MTU network)

Reading the paper is not required for completing the assignment and is only relevant to provide some background information on the dataset that is being used in the following. Please note in particular, that the methodology proposed in the paper is different to the approach to be implemented in the following. The full dataset can be downloaded from Canvas in the file "energy_performance.csv", which contains 768 instances of example buildings that have been generated by the authors of the above paper using a building energy performance simulation tool.

The input variables for the regression function are the following basic building characteristics:

- Relative compactness
- Surface area
- Wall area
- Roof area
- Overall height
- Orientation
- Glazing area
- Glazing area distribution

The goal is to predict how much energy is required to either heat or cool the building as indicated by the two output variables:

- Heating load
- Cooling load

The following tasks will guide you through the process of implementing two polynomial regression functions that take the basic eight building parameters as input and calculate either the expected heating load or the expected cooling load of the building as output.

TASK 1 (input data, 6 points)

Load the data from the file and split it into features [1 point] and targets [1 point]. Determine and output the minimum and maximum heating and cooling loads of buildings present in the dataset [4 points].

TASK 2 (model function, 4 points)

Create a polynomial model function that takes as input parameters the degree of the polynomial, a list of feature vectors as extracted in task 1, and a parameter vector of coefficients and calculates the estimated target vector using a multi-variate polynomial of the specified degree [3 points]. Create a second function that determines the correct size for the parameter vector from the degree of the multi-variate polynomial [1 point].

TASK 3 (linearization, 4 points)

Create a function that calculates the value of the model function implemented in task 2 and its Jacobian at a given linearization point using the numerical linearisation procedure discussed in the lectures/labs. The function should take the degree of the polynomial, a list of feature vectors as extracted in task 1, and the coefficients of the linearization point as input and calculate the estimated target vector and the Jacobian at the linearization point as output.

In your code's comments, clearly indicate and explain where the model function implemented in task 2 is called and why [2 points], and where the partial derivatives for the Jacobian are calculated and how [2 points].

TASK 4 (parameter update, 4 points)

Create a function that calculates the optimal parameter update from the training target vector extracted in task 1 and the estimated target vector and Jacobian calculated in task 3 following the procedure discussed during the lectures/labs. To do that start with calculating the normal equation matrix; make sure that you add a regularisation term to prevent the normal equation system from being singular. Now calculate the residual and built the normal equation system. Solve the normal equation system to obtain the optimal parameter update. The function should take the training target vector and the estimated target vector and Jacobian at the linearization point as input and calculate the optimal parameter update vector as output.

In your code's comments, clearly indicate where the normal equation matrix is calculated and how it is regularised [2 points]. Also indicate exactly where the residuals are calculated and explain how [2 points].

TASK 5 (regression, 5 points)

Create a function that calculates the coefficient vector that best fits the training data. To do that, initialise the parameter vector of coefficients with zeros. Then setup an iterative procedure that alternates linearization and parameter update following the approach discussed during the lectures/labs. The function should take the degree of the polynomial, the training data features, and the training data targets as input and return the best fitting polynomial coefficient vector as output.

In your code's comments, clearly indicate the parameter vector and how it is updated [2 points]. How do you expect the parameter update and the residuals calculated in the previous task to evolve in the iterations? [2 points] How could you use this to determine the number of iterations required? [1 point]

TASK 6 (model selection, 6 points)

Setup two cross-validation procedures, one for the heat loads and one for cooling loads [1 point]. Calculate the difference between the predicted target and the actual target for the test set in each cross-validation fold [1 point] and output the mean of absolute differences across all folds for both the heating load estimation as well as the cooling load estimation [2 points]. Using this as a quality metric, evaluate polynomial degrees ranging between 0 and 2 to determine the optimal degree for the model function for both the heating as well as the cooling loads [2 points].

TASK 7 (evaluation and visualisation of results, 6 points)

Now using the full dataset, estimate the model parameters for both the heating loads as well as the cooling loads using the selected optimal model function as determined in task 6 [1 point]. Calculate the predicted heating and cooling loads using the estimated model parameters for the entire dataset [1 point]. Plot the estimated loads against the true loads for both the heating and the cooling case [2 points]. Calculate and output the mean absolute difference between estimated heating/cooling loads and actual heating/cooling loads [2 points].

TASK 8 (optional, no points)

Compare your results to the results reported by [Tsanas and Xifara, 2012].