Introduction to Machine Learning

ECE 580
Spring 2025

HW #1, Due 01/23/25 3:00pm

Submission Instructions

Submit your work to the corresponding assignment in Gradescope. Although Gradescope accepts multiple file formats, they strongly recommend submitting your assignment as a single PDF file.

It is your responsibility to ensure the uploaded file is: 1) the correct file, 2) complete (includes all pages), 3) legible, and 4) submitted on-time as determined by the Gradescope server system clock.

It is your responsibility to submit a multi-page PDF file and tag the pages that correspond to each question. Pages may be tagged after submission, even if the submission deadline has passed. If you are submitting close to the submission deadline, submit your assignment first then immediately return to tag pages.

When code is requested, submit a PDF print-out of your code. Submitting a URL for a cloud-based repository is insufficient.

Late Submissions

Late submissions will be accepted up to 5 days after the submission deadline, with the following point penalty applied if its late submission is not excused: [1]

1 day (0^+ to 24 hours) late: 2 point deduction ($\frac{1}{5}$ letter grade)
2 days (24^+ to 48 hours) late: 5 point deduction ($\frac{1}{2}$ letter grade)
3 days late: 10 point deduction (1 letter grade)
4 days late: 20 point deduction (2 letter grades)
5 days late: 30 point deduction (3 letter grades)
6 or more days late: score = 0 (not accepted for credit)

The late policy is designed to be minimally punitive for submissions up to 3 days late, yet encourage staying current with the coursework for our course by not allowing one assignment's late submission to overlap with the next assignment's submission.

A homework score will not drop below 0 as a result of applying the late penalty point deduction.

[1] One day = one 24-hour period or fraction thereof.

Structuring and Organizing Your Code

We will use Python and PyTorch this semester. You may choose to use packages/toolboxes authored by others. If you use packages/toolboxes authored by others, you are expected to reference the packages/-toolboxes so we know what external code supported your completion of the assignment. You are also responsible for knowing how to use the packages/toolboxes to achieve what you are asked to do.

Installing PyTorch

If you have not used PyTorch before, I suggest you install it via Anaconda, following instructions in `https://pytorch.org/get-started/locally/`. We also recommend using jupyter notebook for easier debugging.

You are **free to use something like NumPy if you prefer**. However, we provide cleaned data in PyTorch format, so you will need to download and process the data yourself if you do not use PyTorch.

Data Preparation and Exploration

The cleaned Automobile Data Set [2] has been provided in two files: `cleaned_automobile_train_data.pt` and `cleaned_automobile_test_data.pt`. We are going to use this data to predict a car's price from its various features. For now, we are going to **restrict ourselves to the 13 continuous predictor variables** (in the order in which they appear): wheel-base, length, width, height, curb-weight, engine-size, bore, stroke, compression-ratio, horsepower, peak-rpm, city-mpg, and highway-mpg.

(5) 1. (a) Load cleaned version of the train set, available as `cleaned_automobile_train_data.pt`, using the function `load_dataset(filepath)`. A demo is provided in `HW1_Data_loading_function_and_usage.ipynb`. (If you prefer to not use a notebook, simply copy the content into a .py file. For the entirety of this question, only use the train set, do not use the test set.

(5) (b) In a future assignment, you will explore systematic feature (predictor variable) selection. For now, you are going to select features you believe are most promising via data exploration.

For each of the 13 continuous predictor variables (features), plot the target variable (price) as a function of the predictor variable (feature) using a scatter plot. This will produce 13 scatter plots, one for each feature.

(5) (c) Provide a brief summary (2 sentences) of overall trends from the scatter plots, highlighting promising and unpromising features. You may notice that for some features there appears to be a nonlinear relationship between the feature and the car's price. For example, $price = feature^2$, or $price = 1/feature$. Keep this in mind when you propose candidate models for predicting a car's price.

(5) (d) When performing regression, it is preferable to have features that are as independent as possible, as strongly related (correlated) features do not provide much additional information and may lead to computational challenges. For example, if there were two additional continuous features, "km per gallon city" and "km per gallon highway", these features would be highly (perfectly?) correlated with the existing features "city-mpg" and "highway-mpg," respectively, because 1 km = 0.6241 miles. For this reason, we would want to include only one of "km per gallon

[2] `https://archive.ics.uci.edu/ml/datasets/Automobile`

city" and "city-mpg" in our model, and only one of "km per gallon highway" and "highway-mpg" in our model.

Plot each pair-wise combination of features using scatter plots to aid in (visually) identifying features that are related (correlated).

This type of visualization is often referred to as a "scatterplot matrix" or "pairwise scatterplots." (Examples of this type of visualization are available in Figure 3.6 in *Introduction to Statistical Learning* and Figure 1.1 in *Elements of Statistical Learning*.) There will be **a lot** of subplots. Write code (or leverage a package) to do the repetitive heavy lifting for you! Since you are using these subplots to identify correlation trends the subplots do not need to be high-resolution; it is ok if the subplots are "small".

(5) (e) Summarize key correlated features in 2-3 sentences and list the features that you recommend excluding.

(5) (f) Submit a PDF print-out of your code.
(Submitting a URL for a cloud-based repository is insufficient.)

Ordinary Least Squares

Continuing with the **13 continuous predictor variables** from the Automobile Data Set from the UCI Machine Learning Repository to predict a car's price from its characteristics:

(5) 2. (a) Propose a linear model using all or a subset of the 13 continuous predictor variables to predict a car's price from its characteristics. You may propose a model that uses a transformed feature: $price = w^\top \phi(x)$, where x is the original 13-dimensional feature vector. A completely arbitrary example is $\phi(x) = \begin{bmatrix} x[0] \\ x[0]x[2] \\ \log(x[3]) \end{bmatrix}$.

(You may or may not want to utilize your findings from 1. Our solution did not.)

(30) (b) Under your choice of features and feature transformations, find the OLS solution. **Make sure to only use the training set when finding the optimal parameters!**

Ensure that your model achieves $R^2 \geq 0.84$ on the test data.

Tip 1: In our solution, we got about 86% with a relatively naive and brute-force choice of $\phi(x)$ whose dimension is slightly over 100.

Tip 2: When using the closed-form solution for OLS, you may run into numerical stability issues (we did) due to some features being correlated/large. To alleviate this:

1. Do not use `torch.linalg.pinv`. Instead, for matrix M, and vectors w, v, to solve for

$$Mw = v \Leftrightarrow w = M^\dagger v,$$

use `w=torch.linalg.solve(M,v)`

2. Sometimes, when M is ill conditioned, it helps to replace $M + \lambda I$, where λ is a small positive constant. In our case, we used
`w=torch.linalg.solve(M+(1e-10)*M.norm()*torch.eye(M.shape[0]),v)`

This technique of adding λI improves numerical stability. We will go over in lecture on 01/20. Strictly speaking, when you add λI, the setup is no longer called OLS, and is instead called **ridge regression**.

(10) (c) For the parameters you found, and for train set and test set separately, print the MSE error and R^2 score. Report your numbers in a 2x2 table.

The value test set MSE - train set MSE is sometimes called the **generalization gap**. Report this value as well.

(10) (d) Scatter plot the predicted price (on the vertical axis) as a function of the true price (on the horizontal axis). Also plot the line $\hat{price} = price$ (the line representing perfect prediction) as a reference.

This visualization aids in identification of trends in the estimation errors as a function of the target variable (true price).

Provide a short 1-sentence impression of the performance – what points had the largest deviation?

(5) (e) Submit a PDF print-out of your code.
(Submitting a URL for a cloud-based repository is insufficient.)

3. Now simply take $\phi(x) = x$, i.e. no feature transformation.

(10) (a) Find the OLS solution using the untransformed 13 features, again using just the training set. **Tip 2** from 2(b) may be helpful here as well.

(10) (b) For the parameters you found, and for train set and test set separately, print the MSE error and R^2 score. Report your numbers in a 2x2 table.

Again, compute and report the generalization gap: test set MSE - train set MSE

(10) (c) Compare the values presented in 2(c) versus 3(b). Which model would you pick for minimizing the R^2 error? Which model would you pick for minimizing the generalization gap?

Total weight = 120