CS5785 Assignment 1

The homework is generally split into programming exercises and written exercises.

This homework is due on **September 12, 2023 at 11:59 PM EST**. Upload your homework to Gradescope. There are two assignments for this homework in Gradescope. Please note a complete submission should include:

1. A write-up as a single `.pdf` file, which should be submitted to "Homework 1 (write-up)" This file should contain your answers to the written questions **and** exported pdf file / structured write-up of your answers to the coding questions (which should include core codes, plots, outputs, and any comments / explanations).
2. Source code for all of your experiments (AND figures) zipped into a single .zip file, in `.py` files if you use Python or `.ipynb` files if you use the IPython Notebook. If you use some other language, include all build scripts necessary to build and run your project along with instructions on how to compile and run your code. **If you use the IPython Notebook to create any graphs, please make sure you also include them in your write-up.** This should be submitted to "Homework 1 (code)".

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, homework number, and team member names. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and and "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. You are encouraged (but not required) to work in groups of 2.

Lastly, if you use generative AI (e.g., ChatGPT) as a tool for any of the problems, you are required to include a statement that describes how you used that tool. You are not allowed to copy-paste output from generative AI systems directly: you must always use your own words.

IF YOU NEED HELP

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on the Discussions section of Canvas. That way, your questions/solutions will be available to other students in the class.
- Your instructor and TAs will offer office hours, which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. in this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

PROGRAMMING EXERCISES

Part I. The Housing Prices

1. Join the House Prices - Advanced Regression Techniques competition on Kaggle. Download the training and test data.
2. Give 3 examples of continuous and categorical features in the dataset; choose one feature of each type and plot the histogram to illustrate the distribution.
3. Pre-process your data, explain your pre-processing steps, and the reasons why you need them. (Hint: data pre-processing steps can include but are not restricted to: dealing with missing values, normalizing numerical values, dealing with categorical values etc.)
4. One common method of pre-processing categorical features is to use a one-hot encoding (OHE).

   Suppose that we start with a categorical feature x_j, taking three possible values: $x_j \in \{R, G, B\}$. A one-hot encoding of this feature replaces x_j with three new features: x_{jR}, x_{jG}, x_{jB}. Each feature contains a binary value of 0 or 1, depending on the value taken by x_j. For example, if $x_j = G$, then $x_{jG} = 1$ and $x_{jR} = x_{jB} = 0$.

   Give some examples of features that you think should use a one-hot encoding and explain why. Convert at least one feature to a one-hot encoding (you can use your own implementation, or that in pandas or scikit-learn) and visualize the results by plotting feature histograms of the original feature and its new one-hot encoding.
5. Using ordinary least squares (OLS), try to predict house prices on this dataset. Choose the features (or combinations of features) you would like to use or ignore, provided you justify your choice. Evaluate your predictions on the training set using the MSE and the R^2 score. For this question, you need to implement OLS from scratch without using any external libraries or packages.
6. Train your model using all of the training data (all data points, but not necessarily all the features), and test it using the testing data. Submit your results to Kaggle.

Part II. The Titanic Disaster

1. Join the Titanic - Machine Learning from Disaster competition on Kaggle. Download and pre-process the data.
2. Implement logistic regression (it's ok to use sklearn or similar software packages), try to predict whether a passenger survived the disaster with your model. Choose the features (or combinations of features) you would like to use or ignore, provided you justify your choice.
3. Train your classifier using all of the training data, and test it using the testing data. Submit your results to Kaggle.

WRITTEN EXERCISES

1. Conceptual questions.

   (a) Identify one advantage and one disadvantage of using gradient descent as the optimizer of a supervised linear model, compared to using the analytical formula based on the normal equations (as is done in the Ordinary Least Squares algorithm).

   (b) Imagine doing a regression problem. You discretize the output space of y into a large number of small intervals and apply a multi-class classification algorithm (like softmax regression) to predict the interval containing the target output. Would this approach be sufficient to solve the regression problem? Describe why or why not.

   (c) Name one advantage and one disadvantage of performing multi-class classification by training multiple *one-vs-all classifiers* compared to directly using a multi-class algorithm like *softmax regression*.

   (d) The computational complexity of polynomial regression depends on the number of data points in the training set and on the degree of the polynomial that we are trying to fit.

      i. Assuming that each data point has d attributes and we want our model class to consist of all polynomials of d variables and degree at most p, state the dimension of the polynomial features that are needed to implement this model class. Specifying a big-Oh estimate is enough.

      ii. State the computational complexity of applying polynomial least squares with the above set of polynomial features. Again, a big-Oh estimate is sufficient.

      iii. Comment on how the above findings may influence your decision for when to apply polynomial regression in real-world settings.

2. Analytical solution for Ordinary Least Squares. Consider a simple dataset of n training instances $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$, with inputs $x^{(i)} \in \mathbb{R}$ and targets $y^{(i)} \in \mathbb{R}$. We are going to fit a simple linear model with parameters $\theta = (\theta_0, \theta_1)$ on this dataset:

$$f_\theta(x) = \theta_0 + \theta_1 x^{(i)}.$$

   We define the learning objective to be the residual sum of squares, parameterized by θ_0, θ_1:

$$J(\theta_0, \theta_1) = \sum_{i=1}^{n} (y^{(i)} - f_\theta(x)^{(i)})^2$$

   Instead of using gradient descent, which works in an iterative manner, we will derive a formula for the parameters that minimize the objective J.

   (a) Calculate the partial derivatives $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ and $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$.

   (b) Consider the fact that $J(\theta_0, \theta_1)$ has an unique optimum[1], which we denote as θ_0^*, θ_1^*. We can obtain the analytical solution for θ_0^*, θ_1^* by setting the gradient of J to zero, which yields the following normal equations:

[1] $J(\theta_0, \theta_1)$, the cost function for linear regression, has a unique optimum (i.e., a global minimum). The technical reason for this fact is that J is a convex function, which can be verified, for example, by showing that its Hessian is positive semidefinite.

$$\frac{\partial}{\partial \theta_0} J(\theta_0^*, \theta_1) = 0$$
$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1^*) = 0$$

Write out the above equations and use them to prove the following properties:

$$\theta_0^* = \overline{y} - \theta_1 \overline{x}$$

and

$$\theta_1^* = \frac{\sum_{i=1}^n x^{(i)}(y^{(i)} - \overline{y})}{\sum_{i=1}^n x^{(i)}(x^{(i)} - \overline{x})}$$

(Note: $\overline{x} = \frac{1}{n}\sum_{i=1}^n x^{(i)}$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^n y^{(i)}$.)

(c) For the optimal θ_0^*, θ_1^*, calculate the sum of the residuals $\sum_{i=1}^n e^{(i)} = \sum_{i=1}^n (y^{(i)} - (\theta_0^* + \theta_1^* x^{(i)}))$. What can you learn from the value of $\sum_{i=1}^n e^{(i)}$?

3. Maximum Likelihood Estimation

(a) You are conducting an experiment involving the tossing of a four-sided dice. You conducted *eight* dice throws and meticulously recorded the outcomes. Your objective is to determine from this dataset the true probabilities of the dice falling on each of its four sides (which we denote by 1, 2, 3, 4).

The dataset $D = \{3,2,3,4,4,4,2,4\}$, shown in Table 1, consists of the recorded outcomes of the eight dice throws. Each $x^{(i)} \in D$ refers to an individual throw. Each throw in Table 1 is represented by a throw number and a corresponding outcome (e.g., the dice landed on value '3' in throw number 1).

i. Probabilistic Model. Construct a probabilistic model that captures the underlying probabilities governing the dice outcomes. Define the model and its parameters, aiming to establish the probabilities associated with each outcome.

ii. Learning Paradigm. Classify the experiment as fitting into supervised learning, unsupervised learning, or reinforcement learning paradigms.

(b) Write a formula for the log-likelihood of the dataset under the probabilistic model. Provide an intuitive argument for why we would want to optimize this objective.

(c) Calculate the maximum likelihood estimate of the model parameters. Interpret the parameter values within the context of dice toss probabilities, considering how they relate to the observed outcomes.

(d) Recognize scenarios where this approach might yield inaccurate estimates of the true probabilities of the dice falling on each of its four sides. Provide at least one example and elucidate the reasoning behind potential inaccuracies.

Throw Number	Outcome
1	3
2	2
3	3
4	4
5	4
6	4
7	2
8	4

Table 1: Results of Eight Sample Throws of the 4-sided Dice