CMPT 726 Machine Learning
Spring 2021 Assignment 1

Due: Thursday, February 25 at 11:59 pm Pacific Time

This assignment comes with a zip archive containing essential data and starter code that are required to complete the coding component of this assignment.

This assignment is designed to be substantially more challenging than the quizzes and requires thorough understanding of the course material and extensive thought, so **start early!** If you are stuck, come to office hours. Make sure to check the discussion board often for updates, clarifications, corrections and/or hints.

Requests for extensions will not be entertained – make sure you know how to submit your assignment on Canvas and properly account for time difference if you are not in Vancouver.

Warning: Copying others' solutions, seeking help from others not in this course or posting questions online are considered cheating. Consequences are severe and could lead to suspension or expulsion. If you become aware of such instances, you must report them here: https://forms.gle/mKgkKbujKtQojXCf6

Submission Instructions:

Carefully follow the instructions below when submitting your assignment.

1. Submit a **separate** PDF for each problem in the assignment to Canvas. You may typeset your assignment in LaTeX or Word (submit in PDF format, **not** .doc/.docx format) or submit **neatly** handwritten and scanned solutions. We will not be able to give credit to solutions that are not legible. If there are graphs, include those graphs in the correct sections. **Do not** put them in an appendix. We need each solution to be self-contained on pages of its own.

   - At the start of each problem, please state: (1) who you received help from on the problem, and (2) who you provided help to.
   - At the start of the first problem, please copy the following statement and sign your signature next to it. (Mac Preview and FoxIt PDF Reader, among others, have tools to let you sign a PDF file.) We want to make it *extra* clear so that no one inadvertently cheats. *"I certify that all solutions are entirely in my own words and that I have not looked at another student's solutions. I have given credit to all external sources I consulted."*

2. For all coding-related questions, we recommend working out the solution in the provided Jupyter notebook in Google Colab. Then take a screenshot of your solution and the output and include it in your PDF. In addition, you must (1) save the Jupyter notebook with your solutions and (2) copy your code into the Python script specific to each part of the question. You need to submit both the completed Jupyter notebook and the Python scripts as a zip archive named "`CMPT726-419_A1_⟨Last Name⟩_⟨Student ID⟩.zip`", whose directory structure

should be the same as that of the zip archive for the starter code. Do **NOT** include any data files we provided. Please include a short file named README listing your name and student ID. Please make sure that your code doesn't take up inordinate amounts of time or memory. If your code cannot be executed, your solution cannot be verified.

1 Python Configuration and Data Loading (Optional)

We recommend using Google Colab to complete the parts of this assignment that require coding. However, if you would like to set up your own Python environment on your computer, follow the instructions below.

Please follow the instructions below to ensure your Python environment is configured properly, and you are able to successfully load the data provided with this homework. No solution needs to be submitted for this question. For all coding questions, we recommend using Anaconda for Python 3.

(a) Either install Anaconda for Python 3, or ensure you're using Python 3. To ensure you're running Python 3, open a terminal in your operating system and execute the following command:

```
python --version
```

Do not proceed until you're running Python 3.

(b) Install the following dependencies required for this homework by executing the following command in your operating system's terminal:

```
pip install scikit-learn scipy numpy matplotlib
```

Please use Python 3 with the modules specified above to complete this homework.

To check whether your Python environment is configured properly for this homework, ensure the following Python script executes without error. Pay attention to errors raised when attempting to import any dependencies. Resolve such errors by manually installing the required dependency (e.g. execute `pip install numpy` for import errors relating to the numpy package).

```
import sys
if sys.version_info[0] < 3:
    raise Exception("Python 3 not detected.")

import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from scipy import io

for data_name in ["a", "b"]:
    data = io.loadmat("data/%s.mat" % data_name)
    print("\nloaded %s data!" % data_name)
```

2 Probabilities

Consider two discrete random variables A and B, where A can take on the values $\{2, 4\}$ and B can take on the values $\{1, 3, 5\}$. The pmf (probability mass function) of the joint probability distribution of A and B is given by the following table:

a	b	$\Pr(A = a, B = b)$
2	1	0.1
2	3	0.1
2	5	0.1
4	1	0.2
4	3	0.0
4	5	0.5

Using this information solve the following questions.

(a) **Calculate the marginal probability distribution of A and B. Use tables in the following format to summarize your answer. Show your work.**

a	$\Pr(A = a)$
2	
4	

b	$\Pr(B = b)$
1	
3	
5	

(b) **Compute $E[A]$, $E[B]$, and $E[AB]$.**

(c) **Calculate the conditional probability distribution of B given that $A = 4$. Use a table in the following format to summarize your answer. Show your work.**

b	$\Pr(B = b \mid A = 4)$
1	
3	
5	

(d) **Are A and B independent? Why?**

3 Entropy

Let A, B be discrete random variables with the following pmf for the joint probability distribution:

		B	
		0	1
A	0	1/2	0
	1	3/8	1/8

Calculate $H(A)$, $H(B)$, $H(A \mid B)$, $H(A, B)$, and $I(A; B)$ where all logs should be in base 2.

4 Short Answer Questions

(a) Let A, B and C be random variables and a, b and c be the realizations of A, B and C. Define p as the pmf (probability mass function) if A, B and C are discrete random variables, and the pdf (probability density function) if they are continuous random variables. **Prove that** $p(a, b \mid c) = p(a \mid b, c)p(b \mid c)$**.**
Hint: Use the definitions of conditional probabilities.

(b) **Using the fact proved in part (a), prove the conditional version of Bayes' rule, i.e.:**

$$p(b \mid a, c) = \frac{p(a \mid b, c)p(b \mid c)}{p(a \mid c)} \tag{1}$$

(c) **Give two examples of situations where it makes sense to add a regularizer to a model.**

(d) **As you increase the degree of polynomial features in ordinary least squares (OLS), which kind of loss usually goes down at first but then goes back up?**

(e) You have an ordinary least squares model for estimating the fair market value of cars (in dollars), and your training dataset consists of data on 10K cars. You used all the training data for training the model and obtained an average training loss / mean squared error (MSE) of 100. You have to submit your model to Kaggle (a website that runs machine learning competitions) for evaluation and it gave you an average testing loss of 1000000. You don't have access to the testing data that Kaggle uses and can only submit your code and receive the testing loss for a limited number of times. However, there are many choices of features you would like to try, and you will quickly use up all your submission budget if you submit each one to Kaggle. **How can you pick the best choice of features among all possible choices you have in mind without using up your submission budget?**

5 SVD and Eigendecomposition

(In this question, you can use a computer to help you compute the eigenvalues/vectors, singular values/vectors and the inverse of a matrix.)

(a) **Given a matrix** $A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$**, find an analytical expression of** A^{100}**. Show your work.** (You may leave powers in the expression and do not need to compute the numerical value.)

(b) Recall that given a linear model $\hat{y} = \vec{w}^\top \vec{x}$ where $\vec{w}$ is the parameter vector, and $X, \vec{y}$ are the data, the optimal parameters $\vec{w}^*$ under the square loss $L(\vec{w}) = \|\vec{y} - X\vec{w}\|_2^2$ is

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} L(\vec{w}) = (X^\top X)^{-1} X^\top \vec{y}$$

The matrix $X^\dagger = (X^\top X)^{-1} X^\top$ is known as the pseudoinverse.

Let $X = \begin{bmatrix} \sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & \sqrt{\frac{3}{2}} \end{bmatrix}$. **Compute the pseudoinverse matrix $X^\dagger = (X^\top X)^{-1}X^\top$ using the following approaches. Show your work.** (You may use a computer to help you compute the eigenvalues/vectors, singular values/vectors and the inverse of a matrix. You may round numbers to four decimal places of precision.)

(1) **Compute $X^\dagger$ using the brute force approach, i.e. by evaluating the expression $(X^\top X)^{-1}X^\top$ directly.**

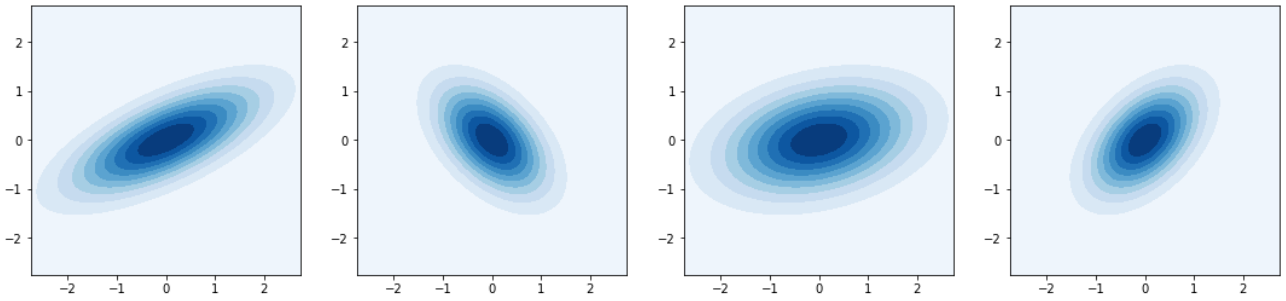
(2) **Compute $X^\dagger$ by applying eigendecomposition on $X^\top X$.**

(3) **Compute $X^\dagger$ by applying singular value decomposition (SVD) on X.**

(c) Below are four contour plots of 2D Gaussians $\mathcal{N}(\vec{0}, \Sigma_i)$ with different covariance matrices Σ_i:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 3 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 3 & 1.2 \\ 1.2 & 1 \end{bmatrix}$$

Match each covariance matrix to the corresponding plot and explain why.



(a) (b) (c) (d)

Figure 1

(d) Given two vectors $\vec{x}$ and $\vec{y}$, the Euclidean distance is defined as $d_E(\vec{x}, \vec{y}) = \|\vec{x}-\vec{y}\|_2 = \sqrt{(\vec{x}-\vec{y})^\top(\vec{x}-\vec{y})}$. Here we introduce a new distance metric called the *Mahalanobis distance*, defined as $d_M(\vec{x}, \vec{y}) = \sqrt{(\vec{x}-\vec{y})^\top S^{-1}(\vec{x}-\vec{y})}$, where S is a covariance matrix that is symmetric and positive semi-definite.

In the questions below, consider two points $\vec{a} = \begin{bmatrix} 10 \\ 5 \end{bmatrix}$ and $\vec{b} = \begin{bmatrix} 7 \\ 9 \end{bmatrix}$. Let $\|M\|_2$ be the induced 2-norm of a matrix.

(1) **Find a covariance matrix S where $\|S^{-1}\|_2 = 1$ and $d_M(\vec{a}, \vec{b}) = d_E(\vec{a}, \vec{b})$. Explain why.**

(2) **Find a covariance matrix S where $\|S^{-1}\|_2 = 1$ and $d_M(\vec{a}, \vec{b}) = 0.1 d_E(\vec{a}, \vec{b})$. Explain why.**

Hint: Think about the geometry of the squared Mahalanobis distance, which is a quadratic form $\vec{u}^\top S^{-1}\vec{u}$. Along which direction does it grow the fastest?

You are required to explain how you found the answer and why it works, though a formal proof is not required. (You may use a computer to verify your answer.)

6 Linear Regression

Making autonomous vehicles involves machine learning for different purposes. One of which is learning how cars actually behave based on their data.

(a) For $t \in \{1, 2, \ldots, T-1\}$, let $x_t, u_t \in \mathbb{R}$ be scalars that approximately observe the relationship $x_{t+1} \approx Ax_t + Bu_t$, where A and B are scalars. An expression of this form arises in control theory, where x_t represents the state and u_t represents the control input. Define the loss function $L(A, B) = \sum_{t=1}^{T-1}(x_{t+1} - Ax_t - Bu_t)^2$, and consider the problem of finding the optimal parameters A and B. **Formulate this as an ordinary least squares (OLS) problem, that is, write down what the data matrix X, label vector $\vec{y}$ and parameter vector $\vec{w}$ in OLS correspond to in the context of this problem. Then write down an expression for the optimal A and B that minimize L.**

(b) In the starter code zip archive associated with this assignment, the values of x_t and u_t are provided in `a.mat`. Write the code for computing the optimal A and B in the Jupyter notebook included in the zip archive. Then use it to compute the optimal A and B. **Take a screenshot of both your code and the output in Jupyter notebook and include it in your PDF submission.** Make sure the screenshot resolution is high enough for the code to be readable and that the optimal A and B are printed at a precision of at least eight decimal places in the screenshot. **Also save your code in the Jupyter notebook and copy your code to `A1_Q6_part_b.py` provided in the zip archive. Submit both the completed notebook and `A1_Q6_part_b.py`.**

(c) For $t \in \{1, 2, \ldots, T-1\}$, let $\vec{x}_t, \vec{u}_t \in \mathbb{R}^3$ be vectors that approximately observes the relationship, $\vec{x}_{t+1} \approx \mathbf{A}\vec{x}_t + \mathbf{B}\vec{u}_t$, where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times3}$ are matrices. Define the loss function $L(\mathbf{A}, \mathbf{B}) = \sum_{t=1}^{T-1}\|\vec{x}_{t+1} - \mathbf{A}\vec{x}_t - \mathbf{B}\vec{u}_t\|_2^2$, and consider the problem of finding the optimal parameters $\mathbf{A}$ and $\mathbf{B}$.

Formulate this as a multiple output linear regression problem, that is, write down what the data matrix X, label matrix Y and parameter matrix W in multiple output linear regression correspond to in the context of this problem. Then write down an expression for the optimal A and B that minimize L.

(d) In the starter code zip archive associated with this assignment, the values of $\vec{x}_t$ and $\vec{u}_t$ are stored in `b.mat`. Write the code for computing the optimal $\mathbf{A}$ and $\mathbf{B}$ using the approach in part (c) in the Jupyter notebook included in the zip archive. Then use it to compute the optimal $\mathbf{A}$ and $\mathbf{B}$. **Take a screenshot of both your code and the output in Jupyter notebook and include it in your PDF submission.** Make sure the screenshot resolution is high enough for the code to be readable and that the optimal $\mathbf{A}$ and $\mathbf{B}$ are printed at a precision of at least eight decimal places in the screenshot. **Also save your code in the Jupyter notebook and copy your code to `A1_Q6_part_d.py` provided in the zip archive. Submit both the completed notebook and `A1_Q6_part_d.py`.**

(e) Consider the same problem as in part (c), where we are given $\vec{x}_t, \vec{u}_t \in \mathbb{R}^3$ and would like to find the optimal $\mathbf{A}$ and $\mathbf{B}$ that minimize the loss function $L(\mathbf{A}, \mathbf{B}) = \sum_{t=1}^{T-1}\|\vec{x}_{t+1} - \mathbf{A}\vec{x}_t - \mathbf{B}\vec{u}_t\|_2^2$. Whereas part (c) asked you to formulate as a multiple output linear regression problem, now formulate it as an ordinary least squares problem, where the model parameters form a vector

rather than a matrix. **Devise a formulation of the problem as an OLS problem, that is, write down what the data matrix X, label vector $\vec{y}$ and parameter vector $\vec{w}$ in OLS correspond to in the context of this problem. Then write down an expression for the optimal A and B that minimize L.**

Hint: Consider each component of $\vec{x}_{t+1}$ separately and express it in terms of the entries of the matrices **A** and **B**.

(f) Write the code for computing the optimal **A** and **B** using the approach in part (e) in the Jupyter notebook included in the starter code zip archive. Then use it to compute the optimal **A** and **B**. Check that it produces the same result as the code in part (d). **Take a screenshot of both your code and the output in Jupyter notebook and include it in your PDF submission.** Make sure the screenshot resolution is high enough for the code to be readable and that the optimal **A** and **B** are printed at a precision of at least eight decimal places in the screenshot. **Also save your code in the Jupyter notebook and copy your code to `A1_Q6_part_f.py` provided in the zip archive. Submit both the completed notebook and `A1_Q6_part_f.py`.**

(g) Consider the general form of multiple output linear regression: we are given a data matrix $\mathbf{X} \in \mathbb{R}^{N\times(n+1)}$ and a label matrix $\mathbf{Y} \in \mathbb{R}^{N\times m}$, where N is the number of observations, n is the number of input dimensions and m is the number of output dimensions. The goal is to find $\mathbf{W} \in \mathbb{R}^{m\times(n+1)}$ that minimizes the loss $L(\mathbf{W}) = \|\mathbf{Y} - \mathbf{X}\mathbf{W}^\top\|_F^2$. **Prove that optimal parameter matrix $\mathbf{W}^*$ is $\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}$.**

Hint: Use the fact that $\|\mathbf{Y} - \mathbf{X}\mathbf{W}^\top\|_F^2 = \sum_{i=1}^{m} \|\vec{y}_i - \mathbf{X}\vec{w}_i\|_2^2$, where $\vec{y}_i$ is the ith column of **Y** and $\vec{w}_i$ is the ith row of **W** (represented as a column vector).

(h) Recall the problem in part (c), where we are given $\vec{x}_t, \vec{u}_t \in \mathbb{R}^3$ and would like to find the optimal **A** and **B** that minimize the loss function $L(\mathbf{A}, \mathbf{B}) = \sum_{t=1}^{T-1} \|\vec{x}_{t+1} - \mathbf{A}\vec{x}_t - \mathbf{B}\vec{u}_t\|_2^2$. Suppose we now add a regularizer to encourage the optimal **A** to be close to the identity matrix. It takes the form $\lambda\|\mathbf{A} - \mathbf{I}\|_F^2$, where $\lambda > 0$ is a hyperparameter, so the new loss function becomes $L(\mathbf{A}, \mathbf{B}) = \left(\sum_{t=1}^{T-1} \|\vec{x}_{t+1} - \mathbf{A}\vec{x}_t - \mathbf{B}\vec{u}_t\|_2^2\right) + \lambda\|\mathbf{A} - \mathbf{I}\|_F^2$. **Derive an expression for the optimal A and B that minimize L.**

