CPSC 340 Assignment 1 (due 2021-05-17 at 9:25 am)

We are providing solutions because supervised learning is easier than unsupervised learning, and so we think having solutions available can help you learn. However, the solution file is meant for you alone and we do not give permission to share these solution files with anyone. Both distributing solution files to other people or using solution files provided to you by other people are considered academic misconduct. Please see UBC's policy on this topic if you are not familiar with it:
http://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,959
http://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,960

Commentary on Assignment 1: CPSC 340 is tough because it combines knowledge and skills across several disciplines. To succeed in the course, you will need to know or very quickly get up to speed on:

- Basic Python programming, including NumPy and plotting with matplotlib.
- Math to the level of the course prerequisites: linear algebra, multivariable calculus, some probability.
- Statistics, algorithms and data structures to the level of the course prerequisites.
- Some basic LaTeX skills so that you can typeset equations and submit your assignments.

This assignment will help you assess whether you are prepared for this course. We anticipate that each of you will have different strengths and weaknesses, so don't be worried if you struggle with *some* aspects of the assignment. **But if you find this assignment to be very difficult overall, that is a warning sign that you may not be prepared to take CPSC 340 at this time.** Future assignments will be more difficult than this one (and probably around the same length).

Questions 1-4 are on review material, that we expect you to know coming into the course. The rest is new CPSC 340 material from the first few lectures.

A note on the provided code: in the `code` directory we provide you with a file called `main.py`. This file, when run with different arguments, runs the code for different parts of the assignment. For example,

```
python main.py -q 6.2
```

runs the code for Question 6.2. At present, this should do nothing (throws a `NotImplementedError`), because the code for Question 6.2 still needs to be written (by you). But we do provide some of the bits and pieces to save you time, so that you can focus on the machine learning aspects. For example, you'll see that the provided code already loads the datasets for you. The file `utils.py` contains some helper functions. You don't need to read or modify the code in there. To complete your assignment, you will need to modify `grads.py`, `main.py`, `decision_stump.py` and `simple_decision.py` (which you'll need to create).

Instructions

Rubric: {points:5}

We use blue to highlight the deliverables that you must answer/do/submit with the assignment.

1 Linear Algebra Review

For these questions you may find it helpful to review these notes on linear algebra:
`http://www.cs.ubc.ca/~schmidtm/Documents/2009_Notes_LinearAlgebra.pdf`

1.1 Basic Operations

Rubric: {points:7}

Use the definitions below,

$$\alpha = 2, \quad x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}, \quad z = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix},$$

and use x_i to denote element i of vector x. Evaluate the following expressions (you do not need to show your work).

1. $\sum_{i=1}^{n} x_i y_i$ (inner product).

   Answer: $0 \cdot 3 + 1 \cdot 4 + 2 \cdot 5 = 14$.

2. $\sum_{i=1}^{n} x_i z_i$ (inner product between orthogonal vectors).

   Answer: $0 \cdot 1 + 1 \cdot 2 + 2 \cdot (-1) = 0$.

3. $\alpha(x + z)$ (vector addition and scalar multiplication)

   Answer: $2\left(\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}\right) = 2\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}$

4. $x^T z + \|x\|$ (inner product in matrix notation and Euclidean norm of x).

   Answer: $0 + \sqrt{0^2 + 1^2 + 2^2} = \sqrt{5}$.

5. Ax (matrix-vector multiplication).

   Answer: $\begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 0 + 2 \cdot 1 + 2 \cdot 2 \\ 1 \cdot 0 + 3 \cdot 1 + 1 \cdot 2 \\ 1 \cdot 0 + 1 \cdot 1 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 7 \end{bmatrix}.$

6. $x^T A x$ (quadratic form).

   Answer: $\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}\begin{bmatrix} 6 \\ 5 \\ 7 \end{bmatrix} = 0 \cdot 6 + 1 \cdot 5 + 2 \cdot 7 = 0 + 5 + 14 = 19.$

7. $A^T A$ (matrix tranpose and matrix multiplication).

   Answer: $\begin{bmatrix} 3 & 1 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{bmatrix}\begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 11 & 10 & 10 \\ 10 & 14 & 10 \\ 10 & 10 & 14 \end{bmatrix}.$

1.2 Matrix Algebra Rules

Rubric: {points:10}

Assume that $\{x, y, z\}$ are $n \times 1$ column vectors, $\{A, B, C\}$ are $n \times n$ real-valued matrices, 0 is the zero matrix of appropriate size, and I is the identity matrix of appropriate size. State whether each of the below is true in general (you do not need to show your work).

1. $x^T y = \sum_{i=1}^n x_i y_i$.

   Answer: True (by using definition of matrix transpose and matrix product).

2. $x^T x = \|x\|^2$.

   Answer: True ($\|x\|^2 = \left(\sqrt{\sum_{i=1}^n x_i^2}\right)^2 = \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i x_i = x^T x$).

3. $x^T x = x x^T$.

   Answer: False (the left side is a scalar and the right side is a matrix).

4. $(x-y)^T(x-y) = \|x\|^2 - 2x^T y + \|y\|^2$.

   Answer: True ($(x-y)^T(x-y) = x^T x - x^T y - y^T x + y^T y$, while $x^T y = y^T x$ and $x^T x = \|x\|^2$).

5. $AB = BA$.

   Answer: False (matrix-multiply is not commutative).

6. $A^T(B + IC) = A^T B + A^T C$.

   Answer: True (identity matrix can be removed, matrix-multiply is distributive across addition).

7. $(A + BC)^T = A^T + B^T C^T$.

   Answer: False (the second term should be $C^T B^T$).

8. $x^T A y = y^T A^T x$.

   Answer: True (dimension of result is 1×1, and scalar equals its transpose).

9. $A^T A = A A^T$ if A is a symmetric matrix.

   Answer: True (symmetry implies $A = A^T$ so $A^T A = AA = A A^T$).

10. $A^T A = 0$ if the columns of A are orthonormal.

    Answer: False (it should be $A^T A = I$).

2 Probability Review

For these questions you may find it helpful to review these notes on probability:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/probability.pdf`
And here are some slides giving visual representations of the ideas as well as some simple examples:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/probabilitySlides.pdf`

2.1 Rules of probability

Rubric: {points:6}

Answer the following questions. You do not need to show your work.

1. You are offered the opportunity to play the following game: your opponent rolls 2 regular 6-sided dice. If the difference between the two rolls is at least 3, you win \$15. Otherwise, you get nothing. What is a fair price for a ticket to play this game once? In other words, what is the expected value of playing the game?

   Answer: There are 12 ways this can happen (the sets $(3,6), (2,5), (2,6), (1,4), (1,5), (1,6)$ and their reverses) out of 36 possibilities, giving $\frac{12}{36}15 + \frac{24}{36}0 = 5$.

2. Consider two events A and B such that $\Pr(A, B) = 0$ (they are mutually exclusive). If $\Pr(A) = 0.4$ and $\Pr(A \cup B) = 0.95$, what is $\Pr(B)$? Note: $p(A, B)$ means "probability of A and B" while $p(A \cup B)$ means "probability of A or B". It may be helpful to draw a Venn diagram.

   Answer: $p(A \cup B) = p(A) + p(B) - p(A, B)$ so $0.95 = 0.4 + p(B)$ or $p(B) = 0.55$.

3. Instead of assuming that A and B are mutually exclusive ($\Pr(A, B) = 0$), what is the answer to the previous question if we assume that A and B are independent?

   Answer: Independence means that $p(A, B) = p(A)p(B)$. So we need to solve the linear equation $0.95 = 0.4 + p(B) - 0.4 \cdot p(B)$. This is equivalent to $(1 - .4)p(B) = 0.55$, or $p(B) = \frac{0.55}{0.6} = 0.91\overline{6}$ or $11/12$.

2.2 Bayes Rule and Conditional Probability

Rubric: {points:10}

Answer the following questions. You do not need to show your work.

Suppose a drug test produces a positive result with probability 0.97 for drug users, $P(T = 1 \mid D = 1) = 0.97$. It also produces a negative result with probability 0.99 for non-drug users, $P(T = 0 \mid D = 0) = 0.99$. The probability that a random person uses the drug is 0.0001, so $P(D = 1) = 0.0001$.

1. What is the probability that a random person would test positive, $P(T = 1)$?

   Answer:

$$\begin{aligned} p(T = 1) &= p(T = 1 \mid D = 1)p(D = 1) + p(T = 1 \mid D = 0)p(D = 0) \\ &= (0.97)(0.0001) + (1 - 0.99)(1 - 0.0001) \\ &\approx 0.000097 + 0.009999 \\ &\approx 0.010096. \end{aligned}$$

2. In the above, do most of these positive tests come from true positives or from false positives?

   Answer: Most come from false positives (≈ 0.01 vs. ≈ 0.0001).

3. What is the probability that a random person who tests positive is a user, $P(D = 1 \mid T = 1)$?

   Answer:

$$\begin{aligned} p(D = 1 \mid T = 1) &= \frac{p(T = 1 \mid D = 1)p(D = 1)}{p(T = 1)} \\ &= \frac{(0.97)(0.0001)}{(0.97)(0.0001) + (1 - 0.99)(1 - 0.0001)} \\ &\approx 0.0096. \end{aligned}$$

4. Suppose you have given this test to a random person and it came back positive, are they likely to be a drug user?

   Answer: No, only about 1% of positives tests are drug users. The vast majority of positive tests are not going to be drug users (false positives).

5. Suppose you are the designer of this drug test. You may change how the test is conducted, which may influence factors like false positive rate, false negative rate, and the number of samples collected. What is one factor you could change to make this a more useful test?

Answer: Increasing $p(T = 1 \mid D = 1)$ even to 1 doesn't help much. You would need to significantly decrease the probability of the test giving false positives, $p(T = 1 \mid D = 0)$. Alternately, it would be useful if you changed the sampling popluation to increase $p(D = 1)$ (maybe first giving another test with a much lower probability of false positives).

3 Calculus Review

3.1 One-variable derivatives

Rubric: {points:8}

Answer the following questions. You do not need to show your work.

1. Find the derivative of the function $f(x) = 3x^2 - 2x + 5$.

   Answer: $f'(x) = 6x - 2$.

2. Find the derivative of the function $f(x) = x(1 - x)$.

   Answer: $f'(x) = 1 - 2x$

3. Let $p(x) = \frac{1}{1+\exp(-x)}$ for $x \in \mathbb{R}$. Compute the derivative of the function $f(x) = x - \log(p(x))$ and simplify it by using the function $p(x)$.

   Answer: Using that $\log(1) = 0$ and the log rules for division we get $f(x) = x - \log(1) + \log(1+\exp(-x)) = x + \log(1 + \exp(-x))$. Applying the chain rule and the definition of of p gives

$$\begin{aligned} f'(x) &= 1 - \frac{\exp(-x)}{1 + \exp(-x)} \\ &= 1 - \frac{1}{1 + \exp(x)} \\ &= 1 - (1 - p(x)) \\ &= p(x). \end{aligned}$$

Remember that in this course we will $\log(x)$ to mean the "natural" logarithm of x, so that $\log(\exp(1)) = 1$. Also, obseve that $p(x) = 1 - p(-x)$ for the final part.

3.2 Multi-variable derivatives

Rubric: {points:5}

Compute the gradient vector $\nabla f(x)$ of each of the following functions. You do not need to show your work.

1. $f(x) = x_1^2 + \exp(x_1 + 2x_2)$ where $x \in \mathbb{R}^2$.

   Answer: $\nabla f(x) = \begin{bmatrix} 2x_1 + \exp(x_1 + 2x_2) \\ 2\exp(x_1 + 2x_2) \end{bmatrix}$.

2. $f(x) = \log\left(\sum_{i=1}^3 \exp(x_i)\right)$ where $x \in \mathbb{R}^3$ (simplify the gradient by defining $Z = \sum_{i=1}^3 \exp(x_i)$).

   Answer: $\nabla f(x) = \begin{bmatrix} \exp(x_1)/Z \\ \exp(x_2)/Z \\ \exp(x_3)/Z \end{bmatrix}$ or $\nabla f(x) = \frac{1}{Z}\begin{bmatrix} \exp(x_1) \\ \exp(x_2) \\ \exp(x_3) \end{bmatrix}$.

3. $f(x) = a^T x + b$ where $x \in \mathbb{R}^3$ and $a \in \mathbb{R}^3$ and $b \in \mathbb{R}$.

   Answer: $\nabla f(x) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$ or $\nabla f(x) = a$.

4. $f(x) = \frac{1}{2} x^\top A x$ where $A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ and $x \in \mathbb{R}^2$.

   Answer: $\nabla f(x) = \begin{bmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 \end{bmatrix}$ or $\nabla f(x) = Ax$.

5. $f(x) = \frac{1}{2}\|x\|^2$ where $x \in \mathbb{R}^d$.

   Answer: $\nabla f(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$ or $\nabla f(x) = x$.

Hint: it is helpful to write out the linear algebra expressions in terms of summations.

3.3 Optimization

Find the following quantities. You do not need to show your work. You can/should use your results from parts 3.1 and 3.2 as part of the process.

1. $\min 3x^2 - 2x + 5$, or, in words, the minimum value of the function $f(x) = 3x^2 - 2x + 5$ for $x \in \mathbb{R}$.

   Answer: $f'(x) = 6x - 2$. Setting this to zero and solving we get $x = 1/3$. The second derivative is $f''(x) = 6$ which is positive for all x so this is a minimizer. Plugging this back in gives $1/3 - 2/3 + 5 = 4\frac{2}{3}$.

2. $\max x(1 - x)$ for $x \in [0, 1]$.

   Answer: $f'(x) = 1 - 2x$ and $f''(x) = -2$, so ignoring the constraints the function is maximized at $x = 1/2$ with a value of $1/4$. But this satisfies the constraints so it's also the constrained maximum.

3. $\min x(1 - x)$ for $x \in [0, 1]$.

   Answer: We know that the function is curved downwards in both directions away from the maximum, and has no other stationary points, so the minimum has be at one of the end points of the interval, $x = 0$ or $x = 1$. Plugging in either of these values gives 0.

4. $\arg\max x(1 - x)$ for $x \in [0, 1]$.

   Answer: As shows above, the maximum occurs at $x = 1/2$.

5. $\min x_1^2 + \exp(x_2)$ where $x \in [0, 1]^2$, or in other words $x_1 \in [0, 1]$ and $x_2 \in [0, 1]$.

   Answer: Since one term involves x_1 only and the other involves x_2 only, we can minimize them separately. They are minimized at $x_1 = 0$ and $x_2 = 0$. So the minimum is $f(x) = 1$.

6. $\arg\min x_1^2 + \exp(x_2)$ where $x \in [0, 1]^2$.

   Answer: See explanation above. The minimum occurs at $x_1 = 0$ and $x_2 = 0$, or $x = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Note: the notation $x \in [0, 1]$ means "x is in the interval $[0, 1]$", or, also equivalently, $0 \leq x \leq 1$.

Note: the notation "$\max f(x)$" means "the value of $f(x)$ where $f(x)$ is maximized", whereas "$\arg\max f(x)$" means "the value of x such that $f(x)$ is maximized". Likewise for min and arg min. For example, the min of the function $f(x) = (x - 1)^2$ is 0 because the smallest possible value is $f(x) = 0$, whereas the arg min is

1 because this smallest value occurs at $x = 1$. The min is always a scalar but the arg min is a value of x, so it's a vector if x is vector-valued.

3.4 Derivatives of code

Rubric: {points:4}

Your repository contains a file named `grads.py` which defines several Python functions that take in an input variable x, which we assume to be a 1-d array (in math terms, a vector). It also includes (blank) functions that return the corresponding gradients. For each function, write code that computes the gradient of the function in Python. You should do this directly in `grads.py`; no need to make a fresh copy of the file. When finished, you can run `python main.py -q 3.4` to test out your code. Submit this code as a screenshot or using the `lstlisting` environment.

Hint: it's probably easiest to first understand on paper what the code is doing, then compute the gradient, and then translate this gradient back into code.

Note: do not worry about the distinction between row vectors and column vectors here. For example, if the correct answer is a vector of length 5, we'll accept numpy arrays of shape `(5,)` (a 1-d array) or `(5,1)` (a column vector) or `(1,5)` (a row vector). In future assignments we will start to be more careful about this.

Warning: Python uses whitespace instead of curly braces to delimit blocks of code. Some people use tabs and other people use spaces. My text editor (Atom) inserts 4 spaces (rather than tabs) when I press the Tab key, so the file `grads.py` is indented in this manner. If your text editor inserts tabs, Python will complain and you might get mysterious errors... this is one of the most annoying aspects of Python, especially when starting out. So, please be aware of this issue! And if in doubt you can just manually indent with 4 spaces, or convert everything to tabs. For more information see `https://www.youtube.com/watch?v=SsoOG6ZeyUI`.

4 Algorithms and Data Structures Review

4.1 Trees

Rubric: {points:2}

Answer the following questions. You do not need to show your work.

1. What is the minimum depth of a binary tree with 64 leaf nodes?

   Answer: $\log_2(64) = 6$.

2. What is the minimum depth of binary tree with 64 nodes (includes leaves and all other nodes)?

   Answer: Depth 5 would have a maximum of $(2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5) = (1 + 2 + 4 + 8 + 16 + 32) = 2^6 - 1 = 63$, so this would also be 6.

Note: we'll use the standard convention that the leaves are not included in the depth, so a tree with depth 1 has 3 nodes with 2 leaves.

4.2 Common Runtimes

Rubric: {points:4}

Answer the following questions using big-O notation You do not need to show your work.

1. What is the cost of finding the largest number in an unsorted list of n numbers?

   Answer: $O(n)$ by searching through the list and tracking the max.

2. What is the cost of finding the smallest element greater than 0 in a *sorted* list with n numbers.

   Answer: $O(\log n)$ with binary search.

3. What is the cost of finding the value associated with a key in a hash table with n numbers? (Assume the values and keys are both scalars.)

   Answer: $O(1)$ using standard implementations.

4. What is the cost of computing the inner product $a^T x$, where a is $d \times 1$ and x is $d \times 1$?

   Answer: $O(d)$ (d multiplications and $(d-1)$ additions).

5. What is the cost of computing the quadratic form $x^T Ax$ when A is $d \times d$ and x is $d \times 1$.

   Answer: $O(d^2)$. Computing Ax is a matrix-vector product involving d dot products, each one involving d multiplications/additions, hence d^2. The secon computation, $x^T(Ax)$ is just the dot proudct between two vectors, which costs $O(d)$ and is negligible.

4.3 Running times of code

Rubric: {points:4}

Your repository contains a file named `bigO.py`, which defines several functions that take an integer argument N. For each function, state the running time as a function of N, using big-O notation.

Answer: $\mathcal{O}(N), \mathcal{O}(N), \mathcal{O}(1), \mathcal{O}(N^2)$

5 Data Exploration

Your repository contains the file `fluTrends.csv`, which contains estimates of the influenza-like illness percentage over 52 weeks on 2005-06 by Google Flu Trends. Your `main.py` loads this data for you and stores it in a pandas DataFrame `X`, where each row corresponds to a week and each column corresponds to a different region. If desired, you can convert from a DataFrame to a raw numpy array with `X.values()`.

5.1 Summary Statistics

Rubric: {points:2}

Report the following statistics:

1. The minimum, maximum, mean, median, and mode of all values across the dataset.
2. The 5%, 25%, 50%, 75%, and 95% quantiles of all values across the dataset.
3. The names of the regions with the highest and lowest means, and the highest and lowest variances.

In light of the above, is the mode a reliable estimate of the most "common" value? Describe another way we could give a meaningful "mode" measurement for this (continuous) data. Note: the function `utils.mode()` will compute the mode value of an array for you.

Answer:

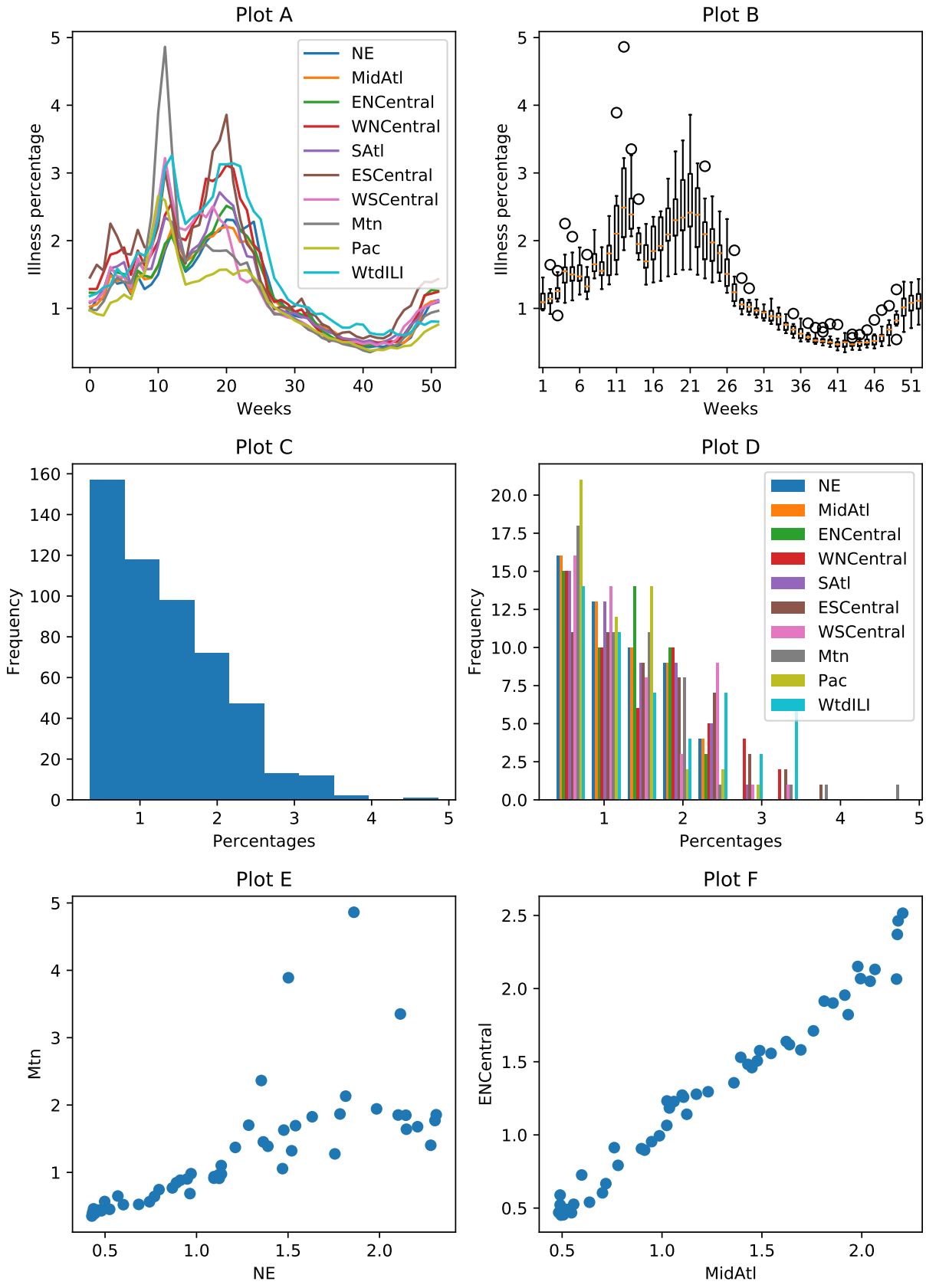
1. Minimum is 0.3520, maximum is 4.8620, mean is 1.3246, median is 1.1590, mode is 0.7700.
2. The quantiles are 0.46, 0.72, 1.16, 1.81, and 2.62.
3. The highest mean is *WtdILI* and highest variance is *Mtn*, the lowest man is *Pac* and it also has the lowest variance.

See the code directory for code. The mode is not necessarily reliable because this is a continuous quantity. It just happened by chance that 4 values of 0.7700 are present in the data, but chance could have led to a very different value (or no values occurring more than once). One way to get a better estimate of the most "common" value is to discretize, and then take the mode. For example if you discretize to 1 or 2 decimal places you would get a value of 0.5, if you discretize to integers you would get a mode of 1. Another approach could be to make a histogram, which also reveals that a mode somewhere in the range $[0.5, 1]$ is reasonable.

5.2 Data Visualization

Rubric: {points:3}

Consider the figure below.


Plot A
Illness percentage
Weeks
NE
MidAtl
ENCentral
WNCentral
SAtl
ESCentral
WSCentral
Mtn
Pac
WtdILI
Plot B
Illness percentage
Weeks
Plot C
Frequency
Percentages
Plot D
Frequency
Percentages
NE
MidAtl
ENCentral
WNCentral
SAtl
ESCentral
WSCentral
Mtn
Pac
WtdILI
Plot E
Mtn
NE
Plot F
ENCentral
MidAtl

The figure contains the following plots, in a shuffled order:

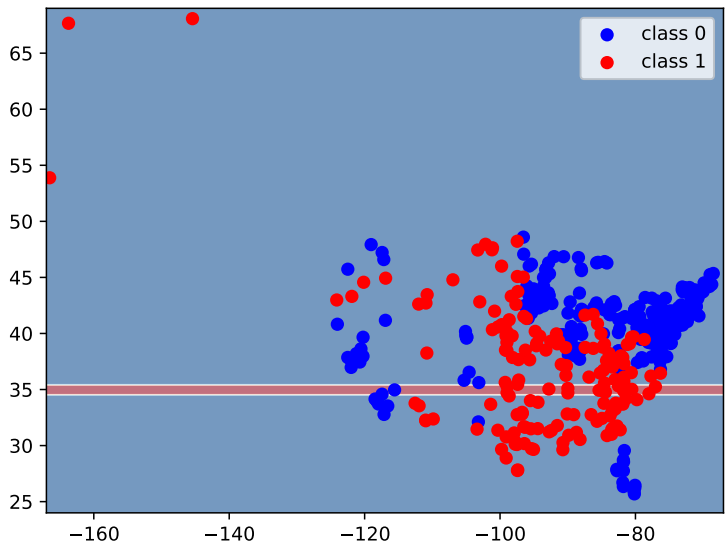
1. A single histogram showing the distribution of *each* column in X.
2. A histogram showing the distribution of each the values in the matrix X.
3. A boxplot grouping data by weeks, showing the distribution across regions for each week.
4. A plot showing the illness percentages over time.
5. A scatterplot between the two regions with highest correlation.
6. A scatterplot between the two regions with lowest correlation.

Match the plots (labeled A-F) with the descriptions above (labeled 1-6), with an extremely brief (a few words is fine) explanation for each decision.

Answer: 1:D, 2:C, 3:B, 4:A, 5:F 6:E

6 Decision Trees

If you run `python main.py -q 6`, it will load a dataset containing longitude and latitude data for 400 cities in the US, along with a class label indicating whether they were a "red" state or a "blue" state in the 2012 election.[1] Specifically, the first column of the variable X contains the longitude and the second variable contains the latitude, while the variable y is set to 0 for blue states and 1 for red states. After it loads the data, it plots the data and then fits two simple classifiers: a classifier that always predicts the most common label (0 in this case) and a decision stump that discretizes the features (by rounding to the nearest integer) and then finds the best equality-based rule (i.e., check if a feature is equal to some value). It reports the training error with these two classifiers, then plots the decision areas made by the decision stump. The plot is shown below:



[1]The cities data was sampled from `http://simplemaps.com/static/demos/resources/us-cities/cities.csv`. The election information was collected from Wikipedia.

As you can see, it is just checking whether the latitude equals 35 and, if so, predicting red (Republican). This is not a very good classifier.

6.1 Splitting rule

Rubric: {points:1}

Is there a particular type of features for which it makes sense to use an equality-based splitting rule rather than the threshold-based splits we discussed in class?

Answer: Categorical features.

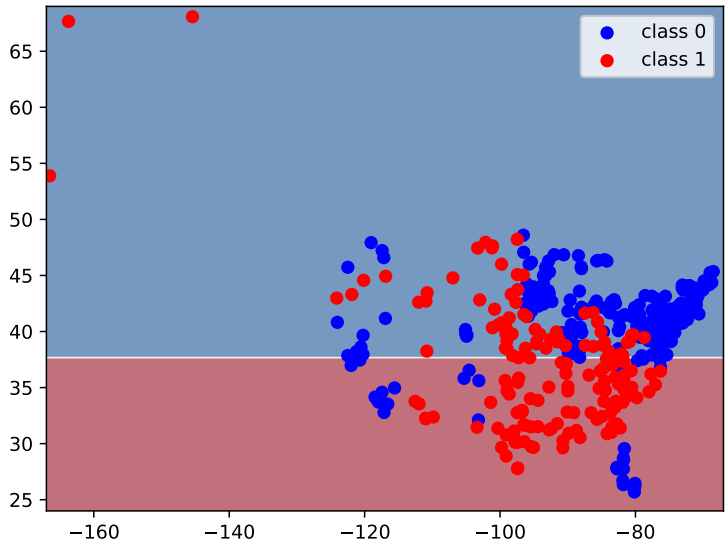
6.2 Decision Stump Implementation

Rubric: {points:3}

The file `decision_stump.py` contains the class `DecisionStumpEquality` which finds the best decision stump using the equality rule and then makes predictions using that rule. Instead of discretizing the data and using a rule based on testing an equality for a single feature, we want to check whether a feature is above or below a threshold and split the data accordingly (this is a more sane approach, which we discussed in class). Create a `DecisionStumpErrorRate` class to do this, and report the updated error you obtain by using inequalities instead of discretizing and testing equality. Submit your class definition code as a screenshot or using the `lstlisting` environment. Also submit the generated figure of the classification boundary.

Hint: you may want to start by copy/pasting the contents `DecisionStumpEquality` and then make modifications from there.

Answer: See code. The improvements reduce the error from 0.41 to 0.25, by splitting the country into a northern (blue) part and a southern (red) part:



6.3 Decision Stump Info Gain Implementation

Rubric: {points:3}

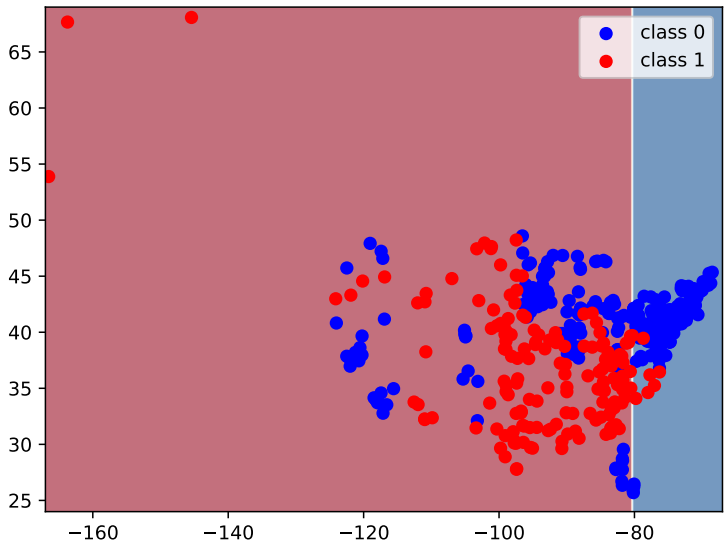
In `decision_stump.py`, create a `DecisionStumpInfoGain` class that fits using the information gain criterion discussed in lecture. Report the updated error you obtain. Submit your class definition code as a screenshot or using the `lstlisting` environment. Submit the classification boundary figure.

Notice how the error rate changed. Are you surprised? If so, hang on until the end of this question!

Note: even though this data set only has 2 classes (red and blue), your implementation should work for any number of classes, just like `DecisionStumpEquality` and `DecisionStumpErrorRate`.

Hint: take a look at the documentation for `np.bincount`, at `https://docs.scipy.org/doc/numpy/reference/generated/numpy.bincount.html`. The `minlength` argument comes in handy here to deal with a tricky corner case: when you consider a split, you might not have any cases of a certain class, like class 1, going to one side of the split. Thus, when you call `np.bincount`, you'll get a shorter array by default, which is not what you want. Setting `minlength` to the number of classes solves this problem.

Answer: Updated error: 0.325. See code.



6.4 Hard-coded Decision Trees

Rubric: {points:2}

Once your `DecisionStumpInfoGain` class is finished, running `python main.py -q 6.4` will fit a decision tree of depth 2 to the same dataset (which results in a lower training error). Look at how the decision tree is stored and how the (recursive) `predict` function works. Using the splits from the fitted depth-2 decision tree, write a hard-coded version of the `predict` function that classifies one example using simple if/else statements (see the Decision Trees lecture). Submit this code as a plain text, as a screenshot or using the `lstlisting` environment.

Note: this code should implement the specific, fixed decision tree which was learned after calling `fit` on this particular data set. It does not need to be a learnable model. You should just hard-code the split values directly into the code. Only the `predict` function is needed.

Hint: if you plot the decision boundary you can do a visual sanity check to see if your code is consistent with the plot.

6.5 Decision Tree Training Error

Rubric: {points:2}

Running `python main.py -q 6.5` fits decision trees of different depths using the following different implementations:

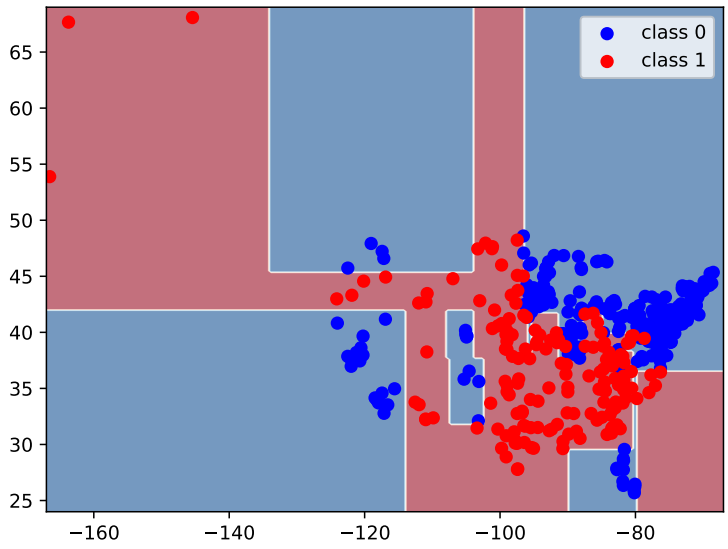
1. A decision tree using `DecisionStumpErrorRate`
2. A decision tree using `DecisionStumpInfoGain`
3. The `DecisionTreeClassifier` from the popular Python ML library *scikit-learn*

Run the code and look at the figure. Describe what you observe. Can you explain the results? Why is approach (1) so disappointing? Also, submit a classification boundary plot of the model with the lowest training error.

Note: we set the `random_state` because sklearn's `DecisionTreeClassifier` is non-deterministic. This is probably because it breaks ties randomly.

Note: the code also prints out the amount of time spent. You'll notice that sklearn's implementation is substantially faster. This is because our implementation is based on the $O(n^2d)$ decision stump learning algorithm and sklearn's implementation presumably uses the faster $O(nd \log n)$ decision stump learning algorithm that we discussed in lecture.

Answer: The training error plateaus in our implementation after a depth of 4, whereas for the sklearn implementation the training error goes all the way to zero. The training stops decreasing after depth 4 because there is no rule to split any of the leaves that increases the classification accuracy: just assigning the majority label gives the same or better accuracy than any split (e.g., because both sides of the split have the same mode of the labels). Here's the plot:



6.6 Comparing implementations

Rubric: {points:2}

In the previous section you compared different implementations of a machine learning algorithm. Let's say that two approaches produce the exact same curve of classification error rate vs. tree depth. Does this conclusively demonstrate that the two implementations are the same? If so, why? If not, what other experiment might you perform to build confidence that the implementations are probably equivalent?

Answer: No, they aren't necessarily the same just because they get the same error rate. They might be classifying examples differently but just happening to get the same number of errors. One simple test is to make sure they are actually making the same predictions for all the inputs in the data set. You could also try this for randomly generated inputs - they don't have to come form the data set, necessarily.

6.7 Cost of Fitting Decision Trees

Rubric: {points:3}

In class, we discussed how in general the decision stump minimizing the classification error can be found in $O(nd \log n)$ time. Using the greedy recursive splitting procedure, what is the total cost of fitting a decision tree of depth m in terms of n, d, and m?

Hint: even thought there could be $(2^m - 1)$ decision stumps, keep in mind not every stump will need to go through every example. Note also that we stop growing the decision tree if a node has no examples, so we may not even need to do anything for many of the $(2^m - 1)$ decision stumps.

Answer: The number of stumps that we need to fit for a decision tree of depth m will be $(2^m - 1)$, so a naive analysis would indicate that we need $O(2^m nd \log n)$ time. However, at each depth we have split the n examples across the decision stumps. This means that each depth will only need to look at n examples, and the cost for each layer is still $O(nd \log n)$. This gives a total cost $O(mnd \log n)$ to go through all m layers. [2]

[2] It's actually possible to go a little faster if you notice that you only need to sort once, and with some bookkeeping you maintain the examples in sorted order as you split them. This would reduce the cost slightly to $O(nd \log n + mnd)$.