Homework #2

CSE 446/546: Machine Learning
Profs. Jamie Morgenstern and Simon Du
Due: **Wednesday** November 3, 2021 11:59pm
A: 96 points, **B:** 29 points

Please review all homework guidance posted on the website before submitting to GradeScope. Reminders:

- Make sure to read the "What to Submit" section following each question and include all items.
- Please provide succinct answers and supporting reasoning for each question. Similarly, when discussing experimental results, concisely create tables and/or figures when appropriate to organize the experimental results. All explanations, tables, and figures for any particular part of a question must be grouped together.
- For every problem involving generating plots, please include the plots as part of your PDF submission.
- When submitting to Gradescope, please link each question from the homework in Gradescope to the location of its answer in your homework PDF. Failure to do so may result in deductions of up to *[5 points]* .
- Please recall that B problems, indicated in $\boxed{\text{boxed text}}$, are only graded for 546 students, and that they will be weighted at most 0.2 of your final GPA (see the course website for details). In Gradescope, there is a place to submit solutions to A and B problems separately. You are welcome to create a single PDF that contains answers to both and submit the same PDF twice, but associate the answers with the individual questions in Gradescope.
- If you collaborate on this homework with others, you must indicate who you worked with on your homework. Failure to do so may result in accusations of plagiarism.
- For every problem involving code, please include the code as part of your PDF for the PDF submission *in addition to* submitting your code to the separate assignment on Gradescope created for code. Not submitting all code files will lead to a deduction of *[1 point]* .
- Please indicate your final answer to each question by placing a box around the main result(s). To do this in LaTeX, one option is using the `boxed` command.

Not adhering to these reminders may result in point deductions.

Short Answer and "True or False" Conceptual questions

A1. The answers to these questions should be answerable without referring to external materials. Briefly justify your answers with a few words.

a. *[2 points]* Suppose that your estimated model for predicting house prices has a large positive weight on the feature `number of bathrooms`. If we remove this feature and refit the model, will the new model have a strictly higher error than before? Why?

Solution: If we remove the feature `number of bathrooms` the new model **may not** have a strictly higher error. To see why this is true, we have to think about related features. In particular, we have to think about perfectly correlated features such as `number of sinks` as mentioned in the lecture notes. We have to consider whether they are included in the model. If, for example, `number of sinks` is still included in the model and these two features are perfectly correlated, then the weight on the feature `number of sinks` will increase while having no effect on the error. The omission of the feature `number of bathrooms` may not have any effect on error if there is a collinear feature which can absorb the effect in terms of a higher weight.

---

b. *[2 points]* Compared to L2 norm penalty, explain why a L1 norm penalty is more likely to result in sparsity (a larger number of 0s) in the weight vector.

Solution:

This has to do with the geometry of the norm balls of L1 and L2. Recall: the L1 norm ball in 2D is shaped like a rhombus with spikes at points where one of the co-ordinates is zero. The L2 norm ball in 2D is a circle. Then, notice that we can write our problem as a constrained optimization problem:

$$\min_{w} RSS(w) \text{ such that } \|w\|_1 \leq \lambda$$

where λ is an upper bound on the L1 norm of the weights. When we look at the contours of the RSS objective function and the contours of the L1 and L2 norms, the point where the lowest level set of the objective function intersects the L1/L2 norm balls is the optimal solution. With the L1 ball the spikes of the ball are more likely to intersect the elliptical level set than one of the sides. These spikes correspond to sparse solutions, which lie on the co-ordinate axes. With the L2 norm, the intersection can happen at any point and since there are no spikes we there is not a high chance of solutions that lie on the co-ordinate axes. In higher dimensions, we will have more of these spikes and more intersections can occur at these spikes.

c. *[2 points]* In at most one sentence each, state one possible upside and one possible downside of using the following regularizer: $\left(\sum_i |w_i|^{0.5}\right)$.

Solution:

Upside: This regularizer is more likely to result in **sparsity** than the L1 and L2 norm.

Downside: The function is **not convex** by observing Figure 3.12 in HTF.

---

d. *[1 point]* True or False: If the step-size for gradient descent is too large, it may not converge.

Solution: $\boxed{\textbf{True.}}$. If step-size for gradient descent is too large, we may never converge. Let us consider a convex function. We know that we move in the right direction with gradient descent, but it is possible that we move too much by overshooting and constantly miss the minimum point on either side of it. It is possible that we just keep bouncing around on either side of the minimum with a constant step size.

---

e. *[2 points]* In your own words, describe why stochastic gradient descent (SGD) works, even though only a small portion of the data is considered at each update.

Solution:

In expectation, the effect of an iteration of stochastic gradient descent is exactly the same as the deterministic effect of one iteration of gradient descent. Thus, even though we only use a small portion of the data we should expect to move towards the minimum over a large number of iterations.

---

f. *[2 points]* In at most one sentence each, state one possible advantage of SGD over GD (gradient descent), and one possible disadvantage of SGD relative to GD.

Solution:

Advantage: An iteration of SGD is less computationally intensive than GD as each iteration of SGD takes $O(n)$ time compared to $O(mn)$ time of GD.

Disadvantage: There is high variance in SGD so in a given iteration, a poor choice of a data point can lead in the completely different direction which does not happen with GD.

Convexity and Norms

A2. A *norm* $\|\cdot\|$ over $\mathbb{R}^n$ is defined by the properties: (i) non-negativity: $\|x\| \geq 0$ for all $x \in \mathbb{R}^n$ with equality if and only if $x = 0$, (ii) absolute scalability: $\|a\,x\| = |a|\,\|x\|$ for all $a \in \mathbb{R}$ and $x \in \mathbb{R}^n$, (iii) triangle inequality: $\|x+y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$.

a. *[3 points]* Show that $f(x) = (\sum_{i=1}^n |x_i|)$ is a norm. (Hint: for (iii), begin by showing that $|a+b| \leq |a| + |b|$ for all $a, b \in \mathbb{R}$.)

Solution:

We will show that $f(x) = (\sum_{i=1}^n |x_i|)$ is a norm. We will first show (i) non negativity:

(i) Let $x \in \mathbb{R}^n$. Since $|x_i| \geq 0$, by definition, we should have that:

$$\sum_{i=1}^n |x_i| \geq 0.$$

Thus, $\boxed{f(x) \geq 0 \;\forall\; x \in \mathbb{R}^n}$. Now, we have to show that the above holds with equality $\iff$ $x = 0$. Suppose $x = 0$:

$$f(x) = \sum_{i=1}^n |x_i| = \sum_{i=1}^n 0 = 0$$

Now, suppose that $f(x) = 0$. For a contradiction suppose that $x \neq 0$. Then, there exists x_j such that $|x_j| > 0$. Hence,

$$f(x) = \sum_{i=1}^n |x_i| \geq |x_j| > 0$$

which is a contradiction of the assumption that $f(x) = 0$. Thus, $\boxed{f(x) = 0 \iff x = 0}$.

Now, we will show (ii) absolute scalability.

(ii) Let $x \in \mathbb{R}^n$ and $a \in \mathbb{R}$. Then,

$$\begin{aligned} f(ax) &= \sum_{i=1}^n |ax_i| \\ &= \sum_{i=1}^n |a||x_i| \\ &= |a| \sum_{i=1}^n |x_i| \\ &= |a| f(x) \end{aligned}$$

Hence, we have shown that $\boxed{f(ax) = |a| f(x) \;\forall\; x \in \mathbb{R}^n, a \in \mathbb{R}}$.

Finally, we will show the (iii) triangle inequality, that is, $f(a+b) \leq f(x) + f(y)$ for all $x, y \in \mathbb{R}^n$. We will begin by showing that $|a+b| \leq |a| + |b|$ for all $a, b \in \mathbb{R}$.

Let $a, b \in \mathbb{R}$. Notice that: $-|a| \leq a \leq |a|$ and $-|b| \leq b \leq |b|$. Adding the two inequalities yield:

$$-(|a| + |b|) \leq a + b \leq |a| + |b|$$

Recall: $|x| \leq y \iff -y \leq x \leq y$ for $y \geq 0$. Thus, letting $x = a + b$ in our example and using the above fact we get the following:

$$-(|a| + |b|) \leq a + b \leq |a| + |b| \implies |a + b| \leq |a| + |b|.$$

Now, let $x, y \in \mathbb{R}^n$. Then.

$$f(x + y) = \sum_{i=1}^{n} |x_i + y_i|$$

Since, $|x_i + y_i| \leq |x_i| + |y_i|$ for all i:

$$\begin{aligned} f(x + y) &= \sum_{i=1}^{n} |x_i + y_i| \\ &\leq \sum_{i=1}^{n} |x_i| + |y_i| \\ &= \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |y_i| \\ &= f(x) + f(y) \text{ (by definition)} \end{aligned}$$

Thus,

$$\boxed{f(x + y) \leq f(x) + f(y)}$$

and so we have shown that $f(x)$ is a norm as it satisfies (i) non-negativity, (ii) absolute scalability and (iii) triangle inequality.

---

b. *[2 points]* Show that $g(x) = \left(\sum_{i=1}^n |x_i|^{1/2}\right)^2$ is not a norm. (Hint: it suffices to find two points in $n = 2$ dimensions such that the triangle inequality does not hold.)

Context: norms are often used in regularization to encourage specific behaviors of solutions. If we define $\|x\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ then one can show that $\|x\|_p$ is a norm for all $p \geq 1$. The important cases of $p = 2$ and $p = 1$ correspond to the penalty for ridge regression and the lasso, respectively.

Solution
Let $n = 2$. We will find two points in $\mathbb{R}^2$ such that the triangle inequality does not hold. This means that we will find x and y in $\mathbb{R}^2$ such that $g(x + y) > g(x) + g(y)$. Notice:

$$\begin{aligned} g(x+y) &= \left(\sum_{i=1}^n |x_i + y_i|^{\frac{1}{2}}\right)^2 \\ &= \left(|x_1 + y_1|^{\frac{1}{2}} + |x_2 + y_2|^{\frac{1}{2}}\right)^2 \end{aligned}$$

and,

$$\begin{aligned} g(x) + g(y) &= \left(\sum_{i=1}^n |x_i|^{\frac{1}{2}}\right)^2 + \left(\sum_{i=1}^n |y_i|^{\frac{1}{2}}\right)^2 \\ &= \left(|x_1|^{\frac{1}{2}} + |x_2|^{\frac{1}{2}}\right)^2 + \left(|y_1|^{\frac{1}{2}} + |y_2|^{\frac{1}{2}}\right)^2 \end{aligned}$$

Consider the following $x, y \in \mathbb{R}^2$:

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then:

$$\begin{aligned} g(x+y) &= \left(|x_1 + y_1|^{\frac{1}{2}} + |x_2 + y_2|^{\frac{1}{2}}\right)^2 \\ &= \left(1 + 1\right)^2 \\ &= 4 \end{aligned}$$

and,

$$\begin{aligned} g(x) + g(y) &= \left(|x_1|^{\frac{1}{2}} + |x_2|^{\frac{1}{2}}\right)^2 + \left(|y_1|^{\frac{1}{2}} + |y_2|^{\frac{1}{2}}\right)^2 \\ &= 1^2 + 1^2 \\ &= 2 < 4 = g(x+y) \end{aligned}$$

Thus, $g(x) = (\sum_{i=1}^n |x_i|^{\frac{1}{2}})^2$ is not a norm.

B1. A set $A \subseteq \mathbb{R}^n$ is *convex* if $\lambda x + (1-\lambda)y \in A$ for all $x, y \in A$ and $\lambda \in [0,1]$. Let $\|\cdot\|$ be a norm.

a. *[3 points]* Show that $f(x) = \|x\|$ is a convex function.

Solution:

We will show that $f(x) = \|x\|$ is a convex function, where $\|x\|$ is a norm. Let $x, y \in A$ and let $\lambda \in [0,1]$. We will show that $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$. Notice:

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= \|\lambda x + (1-\lambda)y\| \\ &\leq \|\lambda x\| + \|(1-\lambda)y\| \qquad \text{(by triangle inequality)} \\ &= |\lambda|\|x\| + |(1-\lambda)|\|y\| \qquad \text{(by absolute scalability)} \\ &= \lambda\|x\| + (1-\lambda)\|y\| \qquad \text{(since } \lambda \in [0,1]\text{)} \\ &= \lambda f(x) + (1-\lambda)f(y) \qquad \text{(by the definition of } f(\cdot)\text{)} \end{aligned}$$

Thus,

$$\boxed{f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)}$$

and hence, $f(x) = \|x\|$ is a convex function.

---

b. *[3 points]* Show that $\{x \in \mathbb{R}^n : \|x\| \leq 1\}$ is a convex set.

Solution:

Define $A = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$. Let $a, b \in A$ and $\lambda \in [0,1]$. We aim to show that $\lambda a + (1-\lambda)b \in A$. Since $a \in A$ and $b \in A$ we have that $\|a\| \leq 1$ and $\|b\| \leq 1$. Recall: $\|\cdot\|$ is a norm. Then:

$$\begin{aligned}
\|\lambda a + (1-\lambda)b\| &\leq \|\lambda a\| + \|(1-\lambda)b\| \quad \text{(by the triangle inequality)} \\
&= |\lambda|\|a\| + |1-\lambda|\|b\| \quad \text{(by absolute scalability)} \\
&= \lambda\|a\| + (1-\lambda)\|b\| \quad \text{(since } \lambda \in [0,1]) \\
&\leq \lambda + (1-\lambda) \text{ (since } a, b \in A) \\
&= 1
\end{aligned}$$

Hence,

$$\boxed{\|\lambda a + (1-\lambda)b\| \leq 1}.$$

This implies $\lambda a + (1-\lambda)b \in A$. Thus, $A = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$ is a convex set.

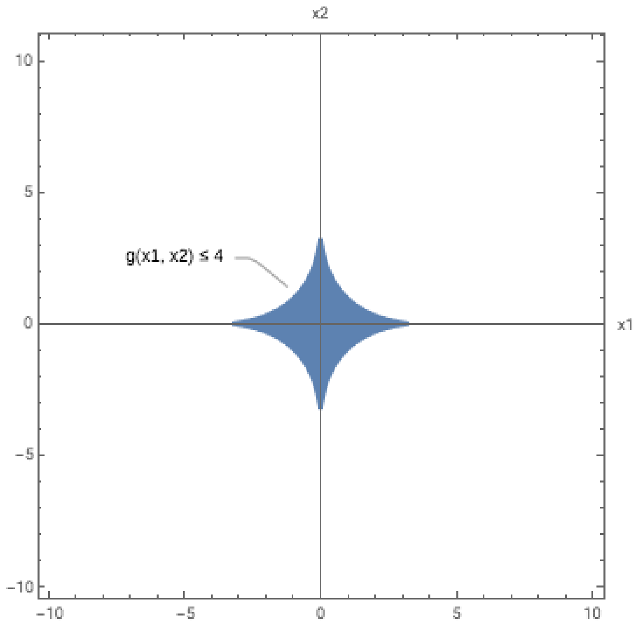
---

c. *[2 points]* Draw a picture of the set $\{(x_1, x_2) : g(x_1, x_2) \le 4\}$ where $g(x_1, x_2) = \left(|x_1|^{1/2} + |x_2|^{1/2}\right)^2$. (This is the function considered in 1b above specialized to $n = 2$.) We know g is not a norm. Is the defined set convex? Why not?

Solution:

The following figure was generated using `Mathematica` and the following line of code:

```
RegionPlot[Callout[(Abs[x]^0.5+Abs[y]^0.5)^2 <= 4,"g(x1, x2) <= 4",{-2.5,2.5}], {x, -10, 10},
{y, -10, 10},Axes ->True, AxesLabel->{x1, x2}, PlotStyle ->DarkBlue]
```



The defined set is **not convex**. To see this, let consider the following points:

$$x = \begin{bmatrix} -4 \\ 0 \end{bmatrix}, \; y = \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

For any $\lambda \in (0, 1)$, the line segment joining the two points $\lambda x + (1 - \lambda y)$ will not be in the set. For example, let $\lambda = \frac{1}{4}$. Then:

$$\lambda x + (1 - \lambda) y = \begin{bmatrix} -1 \\ -3 \end{bmatrix}$$

and this will not be in the set $\{(x_1, x_2) : g(x_1, x_2) \le 4\}$. We can also see this by doing the following calculation:

$$\begin{aligned} g(-1, 3) &= \left(1 + \sqrt{3}\right)^2 \\ &= 7.46 > 4 \end{aligned}$$

Thus, $\lambda x + (1 - \lambda) y$ is not in the given set and the set is not convex.

B2. For $i = 1, \dots, n$ let $\ell_i(w)$ be convex functions over $w \in \mathbb{R}^d$ (e.g., $\ell_i(w) = (y_i - w^\top x_i)^2$), $\|\cdot\|$ is any norm, and $\lambda > 0$.

a. *[3 points]* Show that

$$\sum_{i=1}^{n} \ell_i(w) + \lambda \|w\|$$

is convex over $w \in \mathbb{R}^d$ (Hint: Show that if f, g are convex functions, then $f(x) + g(x)$ is also convex.)

Solution: We will show a stronger version of the hint. We will show that for convex functions $f_i(x)$ for $i = 1, \dots, k$,

$$\sum_{i=1}^{k} f_i(x)$$

is a convex function. We will prove this by induction.

Base case: $(k = 2)$ Suppose $f_1(x)$ and $f_2(x)$ are convex. We will show that $(f_1 + f_2)(x) = f_1(x) + f_2(x)$ is convex. Let $\lambda \in [0, 1]$. Then:

$$\begin{aligned}(f_1 + f_2)(\lambda x + (1 - \lambda y)) &= f_1(\lambda x + (1 - \lambda)y) + f_2(\lambda x + (1 - \lambda y) \\ &\leq \lambda f_1(x) + (1 - \lambda) f_1(y) + \lambda f_2(x) + (1 - \lambda) f_2(y) \\ &= \lambda(f_1(x) + f_2(x)) + (1 - \lambda) f_2(y) \\ &= \lambda(f_1 + f_2)(x) + (1 - \lambda)(f_1 + f_2)(y)\end{aligned}$$

Thus, $f_1(x) + f_2(x)$ is a convex function. Now, suppose that this statement holds true for k functions. We will show that it will be true for $k + 1$. Consider:

$$\sum_{i=1}^{k+1} f_i(x) = \sum_{i=1}^{k} f_i(x) + f_{k+1}(x)$$

By the inductive step,

$$\sum_{i=1}^{k} f_i(x)$$

is convex and by the base case

$$\sum_{i=1}^{k} f_i(x) + f_{k+1}(x) = \sum_{i=1}^{k+1} f_i(x)$$

is convex. Hence, we have shown that a finite sum of convex functions is convex.

We can now refer back to our problem. We have shown that $\|w\|$ is a convex function in B1 a). Clearly, $\lambda \|w\|$ will still be convex. We are also given that $l_i(w)$ for $i = 1, \dots, n$ is a convex function. Therefore,

$$\sum_{i=1}^{n} l_i(w) + \lambda \|w\|$$

is simply a sum of $n+1$ convex functions. Using the result that a finite sum of convex functions is convex, we have that:

$$\sum_{i=1}^{n} l_i(w) + \lambda \|w\|$$

is convex over $w \in \mathbb{R}^d$.

---

b. *[1 point]* Explain in one sentence why we prefer to use loss functions and regularized loss functions that are convex.

Solution:
We prefer to use loss functions and regularized loss functions that are convex because all local minima of convex functions are global minima. Hence, if we find a minimum of a convex function we are guaranteed that it is the global minimum. We do not have to worry about the function attaining a lower value somewhere else on its domain. Convex functions are also efficient to optimize.

Lasso on a Real Dataset

A Lasso Algorithm

Given $\lambda > 0$ and data $\Big(x_i, y_i\Big)_{i=1}^n$, the Lasso is the problem of solving

$$\arg\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \sum_{i=1}^n (x_i^T w + b - y_i)^2 + \lambda \sum_{j=1}^d |w_j|$$

where λ is a regularization parameter. For the programming part of this homework, we have implemented the coordinate descent method shown in Algorithm 1 to solve the Lasso problem for you.

Algorithm 1: Coordinate Descent Algorithm for Lasso

```
while not converged do
    $b \leftarrow \frac{1}{n}\sum_{i=1}^n \Big(y_i - \sum_{j=1}^d w_j x_{i,j}\Big)$
    for $k \in \{1, 2, \cdots d\}$ do
        $a_k \leftarrow 2\sum_{i=1}^n x_{i,k}^2$
        $c_k \leftarrow 2\sum_{i=1}^n x_{i,k}\Big(y_i - (b + \sum_{j\neq k} w_j x_{i,j})\Big)$
        $w_k \leftarrow \begin{cases} (c_k+\lambda)/a_k & c_k < -\lambda \\ 0 & c_k \in [-\lambda, \lambda] \\ (c_k-\lambda)/a_k & c_k > \lambda \end{cases}$
    end
end
```

You will often apply Lasso on the same dataset for many values of λ. This is called a regularization path. One way to do this efficiently is to start at a large λ, and then for each consecutive solution, initialize the algorithm with the previous solution, decreasing λ by a constant ratio (e.g., by a factor of 2).
The smallest value of λ for which the solution $\widehat{w}$ is entirely zero is given by

$$\lambda_{max} = \max_{k=1,\ldots,d} 2\left|\sum_{i=1}^n x_{i,k}\left(y_i - \left(\frac{1}{n}\sum_{j=1}^n y_j\right)\right)\right| \tag{1}$$

This is helpful for choosing the first λ in a regularization path.
A benefit of the Lasso is that if we believe many features are irrelevant for predicting y, the Lasso can be used to enforce a sparse solution, effectively differentiating between the relevant and irrelevant features.

Dataset

Download the training data set "crime-train.txt" and the test data set "crime-test.txt" from the course website. Store your data in your working directory, ensure you have `pandas` installed, and read in the files with the following Python code:

```
import pandas as pd
df_train = pd.read_table("crime-train.txt")
df_test = pd.read_table("crime-test.txt")
```

This stores the data as Pandas `DataFrame` objects. `DataFrame`s are similar to Numpy `array`s but more flexible; unlike `array`s, `DataFrame`s store row and column indices along with the values of the data. Each column of a `DataFrame` can also store data of a different type (here, all data are floats).
Here are a few commands that will get you working with Pandas for this assignment:

```
df.head()                    # Print the first few lines of DataFrame df.
df.index                     # Get the row indices for df.
df.columns                   # Get the column indices.
df[``foo'']                 # Return the column named ``foo''.
df.drop(``foo'', axis = 1)   # Return all columns except ``foo''.
df.values                    # Return the values as a Numpy array.
df[``foo''].values          # Grab column foo and convert to Numpy array.
df.iloc[:3,:3]               # Use numerical indices (like Numpy) to get 3 rows and cols.
```

The data consist of local crime statistics for 1,994 US communities. The response y is the rate of violent crimes reported per capita in a community. The name of the response variable is `ViolentCrimesPerPop`, and it is held in the first column of `df_train` and `df_test`. There are 95 features. These features include many variables. Some features are the consequence of complex political processes, such as the size of the police force and other systemic and historical factors. Others are demographic characteristics of the community, including self-reported statistics about race, age, education, and employment drawn from Census reports.

The dataset is split into a training and test set with 1,595 and 399 entries, respectively. The features have been standardized to have mean 0 and variance 1. We will use this training set to fit a model to predict the crime rate in new communities and evaluate model performance on the test set. As there are a considerable number of input variables and fairly few training observations, overfitting is a serious issue, and the coordinate descent Lasso algorithm may mitigate this problem during training.

The goals of this problem are threefold: (*i*) to encourage you to think about how data collection processes affect the resulting model trained from that data; (*ii*) to encourage you to think deeply about models you might train and how they might be misused; and (*iii*) to see how Lasso encourages sparsity of linear models in settings where d is large relative to n. **We emphasize that training a model on this dataset can suggest a degree of correlation between a community's demographics and the rate at which a community experiences and reports violent crime. We strongly encourage students to consider why these correlations may or may not hold more generally, whether correlations might result from a common cause, and what issues can result in misinterpreting what a model can explain.**

Applying Lasso

A3.

a. *[4 points]* Read the documentation for the original version of this dataset: `http://archive.ics.uci.edu/ml/datasets/communities+and+crime`. Report 3 features included in this dataset for which historical *policy* choices in the US would lead to variability in these features. As an example, the *number of police* in a community is often the consequence of decisions made by governing bodies, elections, and amount of tax revenue available to decision makers.

Solution:

1. `PctImmigRecent`: The percentage of immigrants who immigrated within last 3 years. The number of immigrants that can immigrate is going to be a consequence of policy choices. Immigration policies change across administrations and directly affect the number of people who can enter the US. On a more local level, states can have immigration/refugee friendly policies. For example, some states may have funding for shelters which makes it more likely that the percentage of immigrants will be higher in those streets.
2. `PolicBudgPerPop`: police operating budget. This is again decided by policy makers and is dependent on the tax revenue available to decision makers.
3. `NumStreet`: number of homeless people counted in the street. This can be affected due to policies regarding funding on shelters, housing projects etc. If the policy is to increase funding on shelters

or housing projects, we should expect to see a decrease in the number of homeless people counted in the street.

b. *[4 points]* Before you train a model, describe 3 features in the dataset which might, if found to have nonzero weight in model, be interpreted as *reasons* for higher levels of violent crime, but which might actually be a *result* rather than (or in addition to being) the cause of this violence.

Solution:

1. `racepctblack`: percentage of population that is African American.

   It is possible that this feature has a nonzero weight in the model but this can be thought of as a result of violence. For example, it is possible that white population were able to relocate from high violence areas whereas the African American population could not. This could be due to the fact that the median income and wealth of a white household is higher so they could afford to move. Redlining could also make it harder for African American households to able to get the funds to be able to relocate.

2. `LemasTotalReqPerPop`: total requests for police per 100K population.

   This could also have a positive, non zero weight. However, we should expect higher requests in areas with higher crime. This does not mean that higher number of requests should lead to more crime.

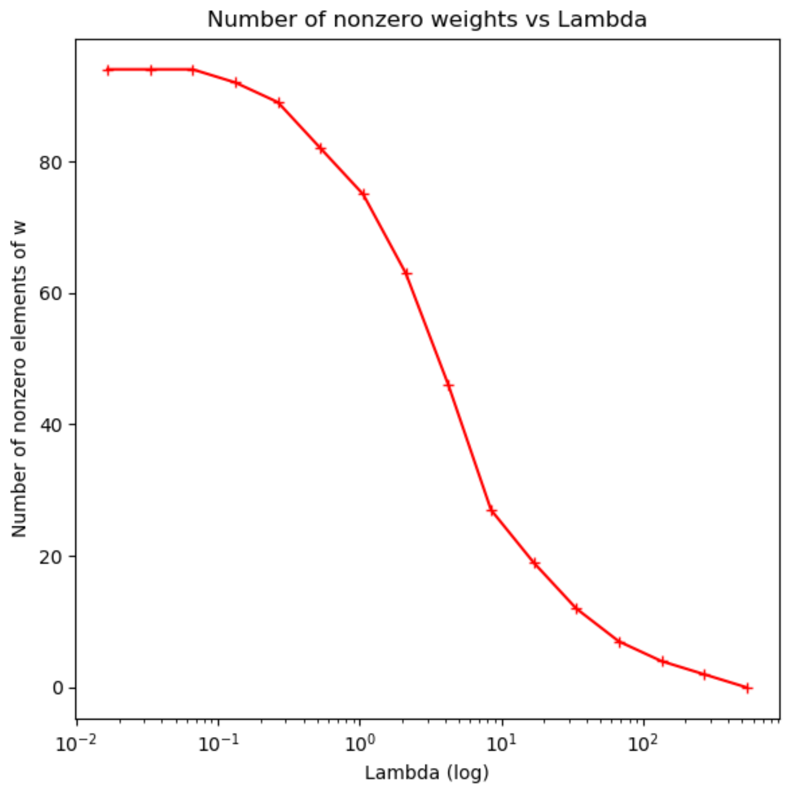
3. `medIncome:` median household income.

   It is possible that this feature has a nonzero, negative weight. We can apply a similar reasoning to the first feature in this list. People, in general, would want to leave areas associated with higher violence. Families with higher income have a higher ability to afford to move out crime infested areas leaving the median income in crime infested areas to be lower lower.

Now, we will run the Lasso solver. Begin with $\lambda = \lambda_{\max}$ defined in Equation (1). Initialize all weights to 0. Then, reduce λ by a factor of 2 and run again, but this time initialize $\hat{w}$ from your $\lambda = \lambda_{\max}$ solution as your initial weights, as described above. Continue the process of reducing λ by a factor of 2 until $\lambda < 0.01$. For all plots use a log-scale for the λ dimension (Tip: use `plt.xscale('log')`).

c. *[4 points]* Plot the number of nonzero weights of each solution as a function of λ.

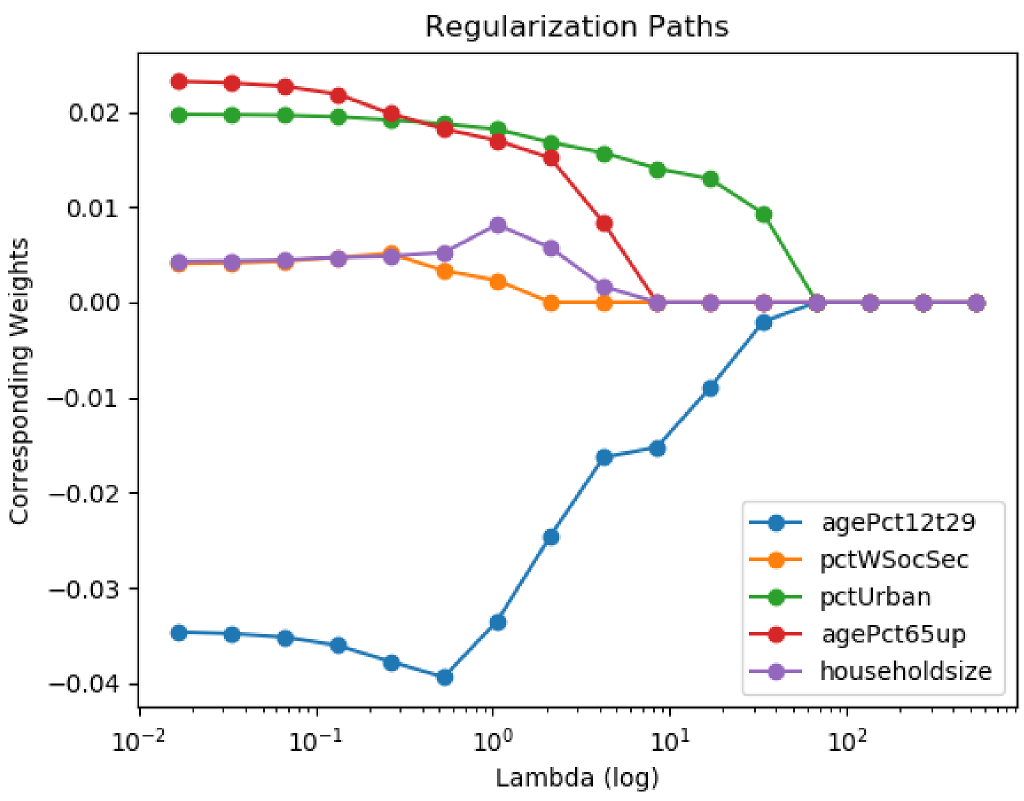
Solution:



---

d. *[4 points]* Plot the regularization paths (in one plot) for the coefficients for input variables `agePct12t29`, `pctWSocSec`, `pctUrban`, `agePct65up`, and `householdsize`.

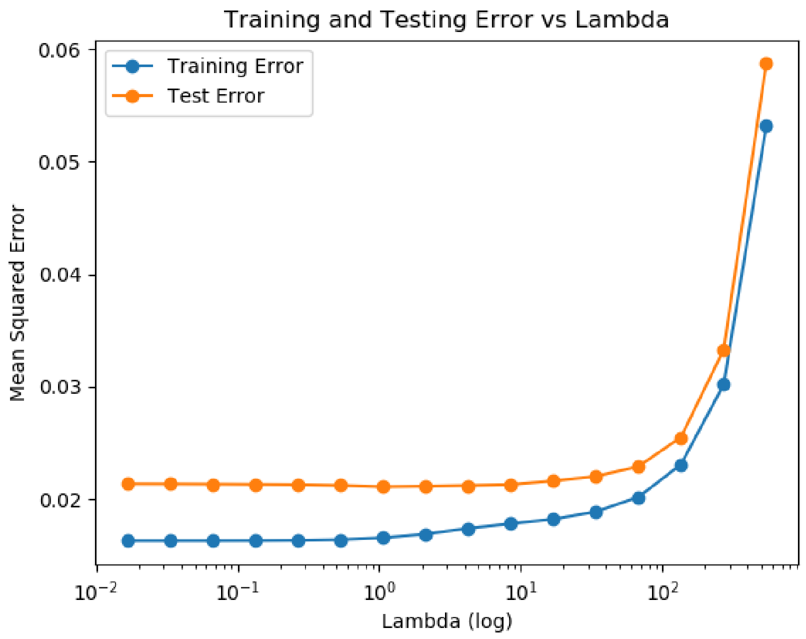
Solution:



---

e. *[4 points]* On one plot, plot the squared error on the training and test data as a function of λ.

Solution:



f. *[4 points]* Sometimes a larger value of λ performs nearly as well as a smaller value, but a larger value will select fewer variables and perhaps be more interpretable. Inspect the weights $\hat{w}$ for $\lambda = 30$. Which feature had the largest (most positive) Lasso coefficient? What about the most negative? Discuss briefly.

Solution:

As reported in the code, $\boxed{\texttt{PctIlleg}}$ has the largest (most positive) Lasso coefficient (0.069) and $\boxed{\texttt{PctKids2Par}}$ has the most negative Lasso coefficient (-0.069) .

This means that percentage of kids born to never married has the highest positive correlation with violent crimes per population. The result is trying to imply that this feature has the highest impact on violent crimes. On the other hand, percentage of kids in family housing with two parents is implied to be the best in reducing violent crimes. We should be careful when interpreting these results, however, and that is why these are simply mentioned as implications of the model and not reasons for violent crimes. One could argue about the stability two parents in a family housing provide and how that is a reason for reduced violence. This would be a causal argument but there could also be an alternative narrative suggesting a feature that impacts violent crimes and these two features. In that case, we would not be able to make a causal argument.

---

g. *[4 points]* Suppose there was a large negative weight on `agePct65up` and upon seeing this result, a politician suggests policies that encourage people over the age of 65 to move to high crime areas in an effort to reduce crime. What is the (statistical) flaw in this line of reasoning? (Hint: fire trucks are often seen around burning buildings, do fire trucks cause fire?)

Solution: The flaw in this line of reasoning is fundamental: correlation $\neq$ causation. It is possible that the having a higher percentage of population that is 65 and over is correlated with lower violence but this does not mean that this causes the lower violence. We should not expect that moving older people into high crime areas will lower crime because we do not expect that a lot of the crimes are committed by people in this age group anyways. The correlation that neighborhoods with a larger percentage of people over 65 and above have lower crime rates can be attributed to the fact that older people may have moved to lower crime neighborhoods.

Logistic Regression

Binary Logistic Regression

A4. Here we consider the MNIST dataset, but for binary classification. Specifically, the task is to determine whether a digit is a 2 or 7. Here, let $Y = 1$ for all the "7" digits in the dataset, and use $Y = -1$ for "2". We will use regularized logistic regression. Given a binary classification dataset $\{(x_i, y_i)\}_{i=1}^n$ for $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ we showed in class that the regularized negative log likelihood objective function can be written as

$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(b + x_i^T w))) + \lambda ||w||_2^2$$

Note that the offset term b is not regularized. For all experiments, use $\lambda = 10^{-1}$. Let $\mu_i(w,b) = \frac{1}{1+\exp(-y_i(b+x_i^T w))}$.

a. *[8 points]* Derive the gradients $\nabla_w J(w,b)$, $\nabla_b J(w,b)$ and give your answers in terms of $\mu_i(w,b)$ (your answers should not contain exponentials).

Solution: Notice:

$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(b + \sum_{j=1}^{d}(x_i)_j w_j))) + \lambda ||w||_2^2$$

Then:

$$\begin{aligned}
\Big(\nabla_w J(w,b)\Big)_j &= \frac{dJ(w,b)}{dw_j} \\
&= \frac{d}{dw_j}\Big(\frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(b + \sum_{j=1}^{d}(x_i)_j w_j))) + \lambda ||w||_2^2\Big) \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{d}{dw_j}\Big(\log(1 + \exp(-y_i(b + \sum_{j=1}^{d}(x_i)_j w_j)))\Big) + \lambda \frac{d}{dw_j}(w_1^2 + \cdots + w_d^2) \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\exp(-y_i(b + x_i^T w))}{1 + \exp(-y_i(b + x_i^T w))} \times (-y_i (x_i)_j) + 2\lambda w_j
\end{aligned}$$

Now, notice: $\exp(-y_i(b + x_i^T w)) = \frac{1}{\mu_i}(w,b) - 1$. Thus,

$$\begin{aligned}
\Big(\nabla_w J(w,b)\Big)_j &= \frac{1}{n}\sum_{i=1}^{n} \frac{\exp(-y_i(b + x_i^T w))}{1 + \exp(-y_i(b + x_i^T w))} \times (-y_i (x_i)_j) + 2\lambda w_j \\
&= \frac{1}{n}\sum_{i=1}^{n} \mu_i(w,b) \times (\frac{1}{\mu_i(w,b)} - 1) \times (-y_i (x_i)_j) + 2\lambda w_j \\
&= \frac{1}{n}\sum_{i=1}^{n} (\mu_i(w,b) - 1)(y_i (x_i)_j) + 2\lambda w_j
\end{aligned}$$

Finally:

$$\begin{aligned}
\nabla_w J(w,b) &= \begin{bmatrix} \Big(\nabla_w J(w,b)\Big)_1 \\ \vdots \\ \Big(\nabla_w J(w,b)\Big)_d \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_1) + 2\lambda w_1 \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_d) + 2\lambda w_d \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_1) \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_d) \end{bmatrix} + \begin{bmatrix} 2\lambda w_1 \\ \vdots \\ 2\lambda w_d \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_1) \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)(y_i(x_i)_d) \end{bmatrix} + 2\lambda w \\
&= \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)y_i x_i + 2\lambda w
\end{aligned}$$

Thus,

$$\boxed{\nabla_w J(w,b) = \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)y_i x_i + 2\lambda w.}$$

Now, we will calculate $\nabla_b J(w,b)$:

$$\begin{aligned}
\nabla_b J(w,b) &= \frac{dJ(w,b)}{db} \\
&= \frac{d}{db}\Big(\frac{1}{n}\sum_{i=1}^n \log(1+\exp(-y_i(b+\sum_{j=1}^d (x_i)_j w_j))) + \lambda||w||_2^2\Big) \\
&= \frac{1}{n}\sum_{i=1}^n \frac{d}{db}\Big(\log(1+\exp(-y_i(b+x_i^T w)))\Big) \\
&= \frac{1}{n}\sum_{i=1}^n \frac{\exp(-y_i(b+x_i^T w))}{1+\exp(-y_i(b+x_i^T w))} \times (-y_i) \\
&= \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)y_i \\
&= \frac{1}{n}(\mu(w,b)-\mathbf{1})^\top y
\end{aligned}$$

Thus:

$$\boxed{\nabla_b J(w,b) = \frac{1}{n}\sum_{i=1}^n (\mu_i(w,b)-1)y_i = \frac{1}{n}(\mu(w,b)-\mathbf{1})^\top y.}$$

---

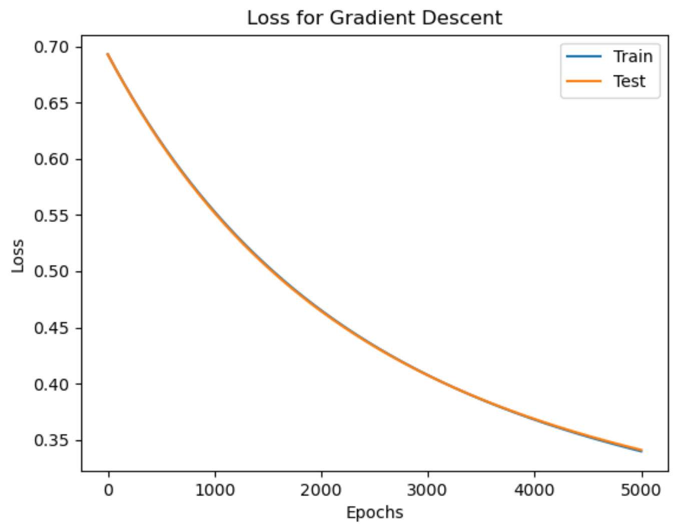
b. *[8 points]* Implement gradient descent with an initial iterate of all zeros. Try several values of step sizes to find one that appears to make convergence on the training set as fast as possible. Run until you feel you are near to convergence.

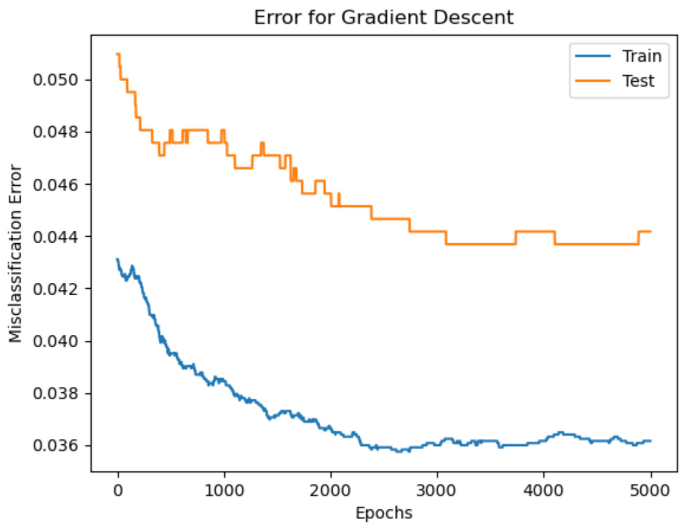
(a) For both the training set and the test, plot $J(w, b)$ as a function of the iteration number (and show both curves on the same plot).

(b) For both the training set and the test, classify the points according to the rule $\text{sign}(b + x_i^T w)$ and plot the misclassification error as a function of the iteration number (and show both curves on the same plot).

Reminder: Make sure you are only using the test set for evaluation (not for training).

Solution:

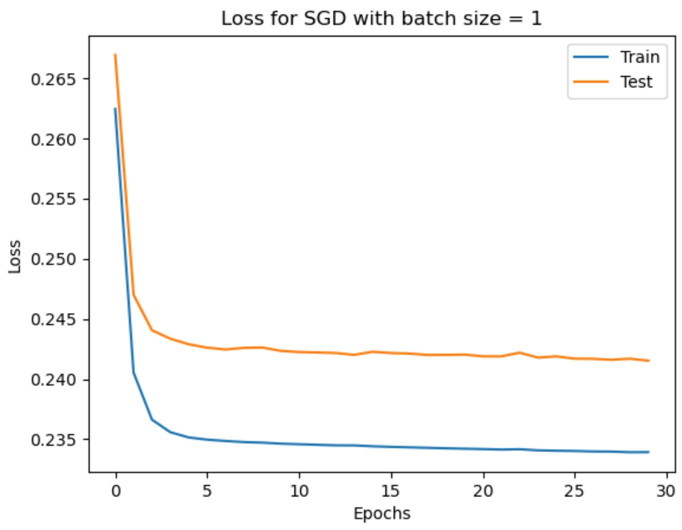


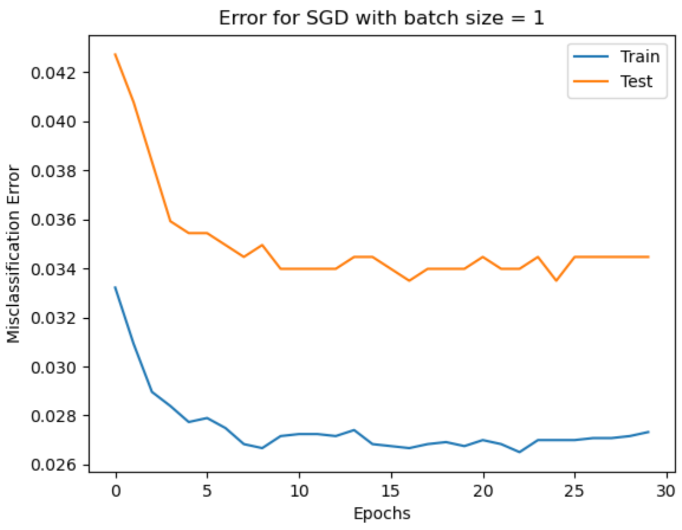


---

c. *[7 points]* Repeat (b) using stochastic gradient descent with a batch size of 1. Note, the expected gradient with respect to the random selection should be equal to the gradient found in part (a). Show both plots described in (b) when using batch size 1. Take careful note of how to scale the regularizer.

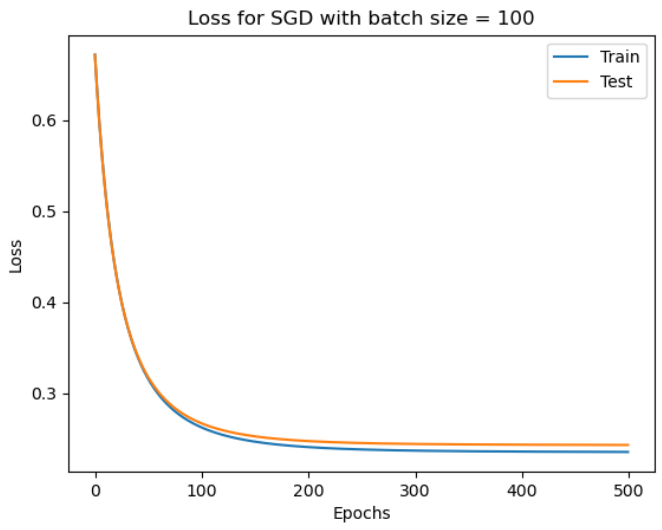
Solution:

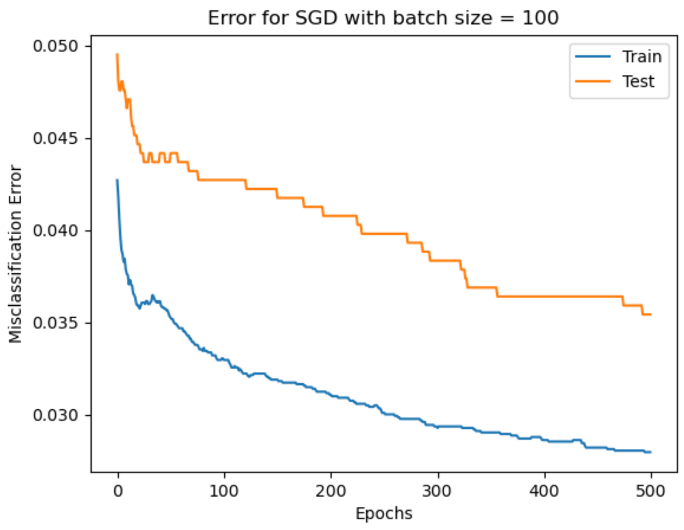




d. *[7 points]* Repeat (b) using stochastic gradient descent with batch size of 100. That is, instead of approximating the gradient with a single example, use 100. Note, the expected gradient with respect to the random selection should be equal to the gradient found in part (a).

Solution:





Ridge Regression on MNIST

These problems were moved from HW1 and are reproduced identically here. If you already started these, you may wish to reuse your work from HW1.

A5. In this problem we will implement a regularized least squares classifier for the MNIST data set. The task is to classify handwritten images of numbers between 0 to 9.

You are **NOT** allowed to use any of the pre-built classifiers in `sklearn`. Feel free to use any method from `numpy` or `scipy`. **Remember:** if you are inverting a matrix in your code, you are probably doing something wrong (Hint: look at `scipy.linalg.solve`).

Each example has features $x_i \in \mathbb{R}^d$ (with $d = 28*28 = 784$) and label $z_j \in \{0, \dots, 9\}$. You can visualize a single example x_i with `imshow` after reshaping it to its original 28×28 image shape (and noting that the label z_j is accurate). We wish to learn a predictor $\widehat{f}$ that takes as input a vector in $\mathbb{R}^d$ and outputs an index in $\{0, \dots, 9\}$. We define our training and testing classification error on a predictor f as

$$\widehat{\epsilon}_{\text{train}}(f) = \frac{1}{N_{\text{train}}} \sum_{(x,z)\in\text{Training Set}} \mathbf{1}\{f(x) \neq z\}$$

$$\widehat{\epsilon}_{\text{test}}(f) = \frac{1}{N_{\text{test}}} \sum_{(x,z)\in\text{Test Set}} \mathbf{1}\{f(x) \neq z\}$$

We will use one-hot encoding of the labels: for each observation (x, z), the original label $z \in \{0, \dots, 9\}$ is mapped to the standard basis vector e_{z+1} where e_i is a vector of size k containing all zeros except for a 1 in the i^{th} position (positions in these vectors are indexed starting at one, hence the $z + 1$ offset for the digit labels). We adopt the notation where we have n data points in our training objective with features $x_i \in \mathbb{R}^d$ and label one-hot encoded as $y_i \in \{0, 1\}^k$. Here, $k = 10$ since there are 10 digits.

---

a. *[10 points]* In this problem we will choose a linear classifier to minimize the regularized least squares objective:

$$\widehat{W} = \text{argmin}_{W\in\mathbb{R}^{d\times k}} \sum_{i=1}^{n} \|W^T x_i - y_i\|_2^2 + \lambda \|W\|_F^2$$

Note that $\|W\|_F$ corresponds to the Frobenius norm of W, i.e. $\|W\|_F^2 = \sum_{i=1}^d \sum_{j=1}^k W_{i,j}^2$. To classify a point x_i we will use the rule $\arg\max_{j=0,\dots,9} e_{j+1}^T \widehat{W}^T x_i$. Note that if $W = \begin{bmatrix} w_1 & \dots & w_k \end{bmatrix}$ then

$$\begin{aligned}
\sum_{i=1}^{n} \|W^T x_i - y_i\|_2^2 + \lambda \|W\|_F^2 &= \sum_{j=1}^{k} \left[\sum_{i=1}^{n} (e_j^T W^T x_i - e_j^T y_i)^2 + \lambda \|W e_j\|^2 \right] \\
&= \sum_{j=1}^{k} \left[\sum_{i=1}^{n} (w_j^T x_i - e_j^T y_i)^2 + \lambda \|w_j\|^2 \right] \\
&= \sum_{j=1}^{k} \left[\|X w_j - Y e_j\|^2 + \lambda \|w_j\|^2 \right]
\end{aligned}$$

where $X = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^\top \in \mathbb{R}^{n\times d}$ and $Y = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^\top \in \mathbb{R}^{n\times k}$. Show that

$$\widehat{W} = (X^T X + \lambda I)^{-1} X^T Y$$

Solution: Using the hint, we can rewrite the objective function as the following:

$$
\begin{aligned}
\sum_{i=1}^{n} \|W^T x_i - y_i\|_2^2 + \lambda \|W\|_F^2 &= \sum_{j=1}^{k} \left[\|Xw_j - Ye_j\|^2 + \lambda \|w_j\|^2 \right] \\
&= (Xw_j - Ye_j)^\top (Xw_j - Ye_j) + \lambda w_j^\top w_j
\end{aligned}
$$

Then:

$$
\frac{d}{dw_j} \Big(\sum_{i=1}^{n} \|W^T x_i - y_i\|_2^2 + \lambda \|W\|_F^2 \Big) = 2X^\top (Xw_j - Ye_j) + 2\lambda w_j
$$

Equating the above to zero:

$$
\begin{aligned}
& X^\top (Xw_j - Ye_j) + \lambda w_j = 0 \\
\implies & X^\top X w_j + \lambda w_j = X^\top Y e_j \\
\implies & (X^\top X + \lambda I_d) w_j = X^\top Y e_j \\
\implies & w_j = (X^\top X + \lambda I_d)^{-1} X^\top Y e_j
\end{aligned}
$$

Notice: Ye_j is simply the j-th column of Y. Therefore,

$$
\begin{aligned}
\widehat{W} &= \begin{bmatrix} w_1 & \dots & w_k \end{bmatrix} \\
&= \begin{bmatrix} (X^\top X + \lambda I_d)^{-1} X^\top Y e_1 & \dots & (X^\top X + \lambda I_d)^{-1} X^\top Y e_k \end{bmatrix} \\
&= (X^\top X + \lambda I_d)^{-1} X^\top \begin{bmatrix} Y e_1 & \dots & Y e_k \end{bmatrix} \\
&= \boxed{(\mathrm{X}^T X + \lambda I_d)^{-1} X^\top Y}
\end{aligned}
$$

b. *[10 points]*

- Implement a function `train` that takes as input $X \in \mathbb{R}^{n \times d}$, $Y \in \{0,1\}^{n \times k}$, $\lambda > 0$ and returns $\widehat{W} \in \mathbb{R}^{d \times k}$.
- Implement a function `one_hot` that takes as input $Y \in \{0, ..., k-1\}^n$, and returns $Y \in \{0,1\}^{n \times k}$.
- Implement a function `predict` that takes as input $W \in \mathbb{R}^{d \times k}$, $X' \in \mathbb{R}^{m \times d}$ and returns an m-length vector with the ith entry equal to $\arg\max_{j=0,\ldots,9} e_j^T W^T x_i'$ where $x_i' \in \mathbb{R}^d$ is a column vector representing the ith example from X'.
- Using the functions you coded above, train a model to estimate $\widehat{W}$ on the MNIST training data with $\lambda = 10^{-4}$, and make label predictions on the test data. This behavior is implemented in `main` function provided in zip file. **What is the training and testing error?** Note that they should both be about 15%.

Solution:
From our code, we get the following:

$$\boxed{\text{Train Error: 15\% and Test Error: 15\%}}$$

We can remove the formatting to get the actual training and testing error:

$$\boxed{\text{Train Error: 14.804999999999998\% and Test Error: 14.66\%}}$$

B3.

a. *[5 points]* Instead of reporting just the test error, which is an unbiased estimate of the *true* error, we would like to report a *confidence interval* around the test error that contains the true error.

Lemma 1. *(Hoeffding's inequality) Fix $\delta \in (0,1)$. If for all $i = 1, \dots, m$ we have that X_i are i.i.d. random variables with $X_i \in [a,b]$ and $\mathbb{E}[X_i] = \mu$ then*

$$\mathbb{P}\left(\left| \left(\frac{1}{m} \sum_{i=1}^{m} X_i \right) - \mu \right| \geq \sqrt{\frac{(b-a)^2 \log(2/\delta)}{2m}} \right) \leq \delta$$

We will use the above equation to construct a confidence interval around the true classification error $\epsilon(\widehat{f}) = \mathbb{E}_{\text{test}}[\widehat{\epsilon}_{\text{test}}(\widehat{f})]$ since the test error $\widehat{\epsilon}_{\text{test}}(\widehat{f})$ is just the average of indicator variables taking values in $\{0,1\}$ corresponding to the ith test example being classified correctly or not, respectively, where an error happens with probability $\mu = \epsilon(\widehat{f}) = \mathbb{E}_{\text{test}}[\widehat{\epsilon}_{\text{test}}(\widehat{f})]$, the *true* classification error.

Let $\widehat{p}$ be the value of p that approximately minimizes the validation error on the plot you just made and use $\widehat{f}(x) = \arg\max_j x^T \widehat{W}^{\widehat{p}} e_j$ to compute the classification test error $\widehat{\epsilon}_{\text{test}}(\widehat{f})$. Use Hoeffding's inequality, of above, to compute a confidence interval that contains $\mathbb{E}_{\text{test}}[\widehat{\epsilon}_{\text{test}}(\widehat{f})]$ (i.e., the *true* error) with probability at least 0.95 (i.e., $\delta = 0.05$). Report $\widehat{\epsilon}_{\text{test}}(\widehat{f})$ and the confidence interval.

Solution: We will use Hoeffding's inequality for this part. First, we have fixed $\delta = 0.05$. We will also define the following:

$$\widehat{\epsilon}_{\text{test}}(\widehat{f}) = \frac{1}{m} \sum_{i=1}^{m} X_i$$

The test error $\widehat{\epsilon}_{\text{test}}(\widehat{f})$ is just the average of indicator variables taking values in $\{0,1\}$, corresponding to the ith test example being classified correctly or not. Therefore, we can rewrite:

$$\widehat{\epsilon}_{\text{test}}(\widehat{f}) = \frac{1}{m} \sum_{i=1}^{m} 1\{\widehat{f}(x_i) = y_i\}$$

Thus, the i.i.d random variables X_i in this case is equal to $1\{\widehat{f}(x_i) = y_i\}$. Since, they only take values $\{0,1\}$ we have that $b = 1$ and $a = 0$. Thus, using Hoeffding's inequality we have:

$$\mathbb{P}\left(\left| \widehat{\epsilon}_{\text{test}}(\widehat{f}) - \mu \right| \geq \sqrt{\frac{(b-a)^2 \log(2/\delta)}{2m}} \right) \leq \delta \tag{2}$$

$$\implies \mathbb{P}\left(\left| \widehat{\epsilon}_{\text{test}}(\widehat{f}) - \mu \right| \geq \sqrt{\frac{\log(2/0.05)}{2m}} \right) \leq \delta \tag{3}$$

Thus, a confidence interval that contains $\epsilon(\widehat{f})$ with probability at least 0.95 is given by the following:

$$\epsilon(\widehat{f}) \in \widehat{\epsilon}_{\text{test}}(\widehat{f}) \pm v$$

where

$$\boxed{v = \sqrt{\frac{\log(2/0.05)}{2m}}}.$$

This is because:

$$\begin{aligned}\mathbb{P}\left(\widehat{\epsilon}_{\text{test}}(\widehat{f})-v\le\epsilon(\widehat{f})\le\widehat{\epsilon}_{\text{test}}(\widehat{f})+v\right)&=\mathbb{P}\left(-v\le\epsilon(\widehat{f})-\widehat{\epsilon}_{\text{test}}(\widehat{f})\le v\right)\\&=\mathbb{P}\left(\left|\epsilon(\widehat{f})-\widehat{\epsilon}_{\text{test}}(\widehat{f})\right|\le v\right)\\&=\mathbb{P}\left(\left|\widehat{\epsilon}_{\text{test}}(\widehat{f})-\epsilon(\widehat{f})\right|\le v\right)\\&=1-\mathbb{P}\left(\left|\widehat{\epsilon}_{\text{test}}(\widehat{f})-\epsilon(\widehat{f})\right|\ge v\right)\\&\ge 1-\delta=0.95\end{aligned}$$

Confidence Interval of Least Squares Estimation

Bounding the Estimate

B4. Let us consider the setting, where we have n inputs, $X_1, ..., X_n \in \mathbb{R}^d$, and n observations $Y_i = \langle X_i, \beta^* \rangle + \epsilon_i$, for $i = 1, ..., n$. Here, β^* is a ground truth vector in $\mathbb{R}^d$ that we are trying to estimate, the noise $\epsilon_i \sim \mathcal{N}(0, 1)$, and the n examples piled up — $X \in R^{n \times d}$. To estimate, we use the least squares estimator $\widehat{\beta} = \min_\beta \|X\beta - Y\|_2^2$. Moreover, we will use $n = 20000$ and $d = 10000$ in this problem.

a. *[3 points]* Show that $\widehat{\beta}_j \sim \mathcal{N}(\beta_j^*, (X^TX)^{-1}_{j,j})$ for each $j = 1, ..., d$. *(Hint: see notes on confidence intervals from lecture.)*

Solution:

From class, we have that:

$$\begin{aligned}
\widehat{\beta} &= (X^TX)^{-1}X^TY \\
&= (X^TX)^{-1}X^T(X\beta^* + \epsilon) \\
&= \beta^* + (X^TX)^{-1}X^T\epsilon \\
\implies \widehat{\beta} - \beta^* &= (X^TX)^{-1}X^T\epsilon \\
\widehat{\beta} &= \beta^* + (X^TX)^{-1}X^T\epsilon
\end{aligned}$$

Using Proposition 1, X constant and the fact that $\epsilon \sim \mathcal{N}(0, I)$:

$$\begin{aligned}
\widehat{\beta} \sim &\mathcal{N}\Big(0 * (X^TX)^{-1}X^T\epsilon + \beta^*, (X^TX)^{-1}X^TI((X^TX)^{-1}X^T)^T\Big) \\
= &\mathcal{N}\Big(\beta^*, (X^TX)^{-1}X^T((X^TX)^{-1}X^T)^T\Big) \\
= &\mathcal{N}\Big(\beta^*, (X^TX)^{-1}X^TX(X^{-1}X^{-T})^T\Big) \\
= &\mathcal{N}\Big(\beta^*, X^{-1}X^{-T}\Big) \\
= &\mathcal{N}\Big(\beta^*, (X^TX)^{-1}\Big)
\end{aligned}$$

$(X^TX)^{-1}$ is the covariance matrix of $\widehat{\beta}$. For a given element $\widehat{\beta}_j$, the variance is then going to be the j-th diagonal element of $(X^TX)^{-1}$. Since $\widehat{\beta}$ is normally distributed, all its components are going to be normally distributed as well. Hence,

$$\boxed{\widehat{\beta}_j \sim \mathcal{N}\Big(\beta_j^*, (X^TX)^{-1}_{j,j}\Big)}$$

If X was not considered to be fixed, but rather a random variable then we would require further assumptions. We would require $\mathbb{E}\big[X^T\epsilon\big] = 0$ to show that, in fact, $\mathbb{E}\Big[\widehat{\beta}\Big] = \beta^*$. We can assume that $\mathbb{E}[\epsilon|X] = 0$ and then use the Law of Iterated Expectations to get the desired result.

b. *[4 points]* Fix $\delta \in (0,1)$ suppose $\beta^* = 0$. Applying the proposition from the notes, conclude that for each $j \in [d]$, with probability at least $1-\delta$, $|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(2/\delta)}$. Can we conclude that with probability at least $1-\delta$, $|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(2/\delta)}$ for all $j \in [d]$ simultaneously? Why or why not?

Solution:

Let us fix $\delta \in (0,1)$ and let $j \in [d]$. We have shown above that:

$$\begin{aligned} &\widehat{\beta}_j \sim \mathcal{N}\Big(\beta_j^*, (X^TX)^{-1}_{j,j}\Big) \\ \implies &(\widehat{\beta}_j - \beta_j^*) \sim \mathcal{N}\Big(0, (X^TX)^{-1}_{j,j}\Big) \\ \implies &\frac{(\widehat{\beta}_j - \beta_j^*)}{(X^TX)^{-1}_{j,j}} \sim \mathcal{N}\Big(0, 1\Big) \\ \implies &\frac{\widehat{\beta}_j}{(X^TX)^{-1}_{j,j}} \sim \mathcal{N}\Big(0, 1\Big) \end{aligned}$$

Then, by Proposition 2, we have that with probability at least $1-\delta$, $\widehat{\beta}_j$ will be contained within the confidence interval:

$$\Big[\beta^* - \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}, \beta^* + \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Big] = \Big[-\sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}, \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Big]$$

Therefore,

$$\mathbb{P}\Bigg(-\sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})} \leq \widehat{\beta}_j \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Bigg) \geq 1-\delta$$

$$\boxed{\implies \mathbb{P}\Bigg(|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Bigg) \geq 1-\delta.}$$

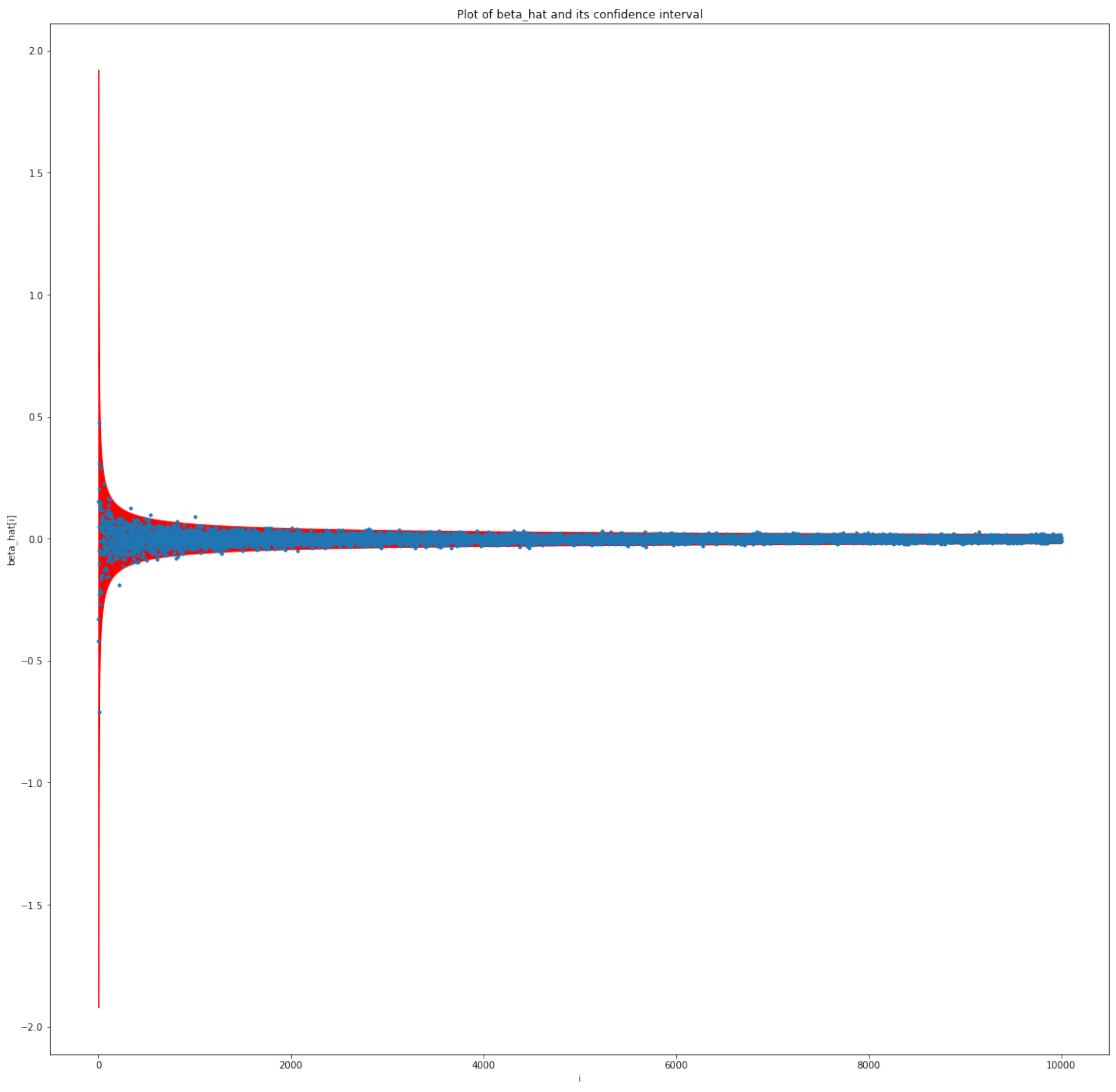
No, we cannot conclude that with probability at least $1-\delta$, $|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}$ for all $j \in [d]$ simultaneously. Each of these events have probability at least $1-\delta$ but this does not guarantee that the probability of all of these events happening simultaneously is at least $1-\delta$. If we consider the events $|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}$ to be independent, then:

$$\begin{aligned} &\mathbb{P}\Bigg(|\widehat{\beta}_j| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})} \text{ for all } j \in [d]\Bigg) \\ &= \mathbb{P}\Bigg(|\widehat{\beta}_1| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Bigg) \times \cdots \times \mathbb{P}\Bigg(|\widehat{\beta}_d| \leq \sqrt{2(X^TX)^{-1}_{j,j}\log(\frac{2}{\delta})}\Bigg) \\ &\geq (1-\delta)^d \end{aligned}$$

Since $(1-\delta) \in (0,1)$ this lower bound will now be smaller.

c. *[5 points]* Let's explore this question empirically. Assume data is generated as $x_i = \sqrt{(i \mod d) + 1} \cdot e_{(i \mod d)+1}$ where e_i is the ith canonical vector and $i \mod d$ is the remainder of i when divided by d. Generate each y_i according to the model above. Compute $\widehat{\beta}$ and plot each $\widehat{\beta}_j$ as a scatter plot with the x-axis as $j \in \{1, \dots, d\}$. Plot $\pm\sqrt{2(X^TX)^{-1}_{j,j} \log(2/\delta)}$ as the upper and lower confidence intervals with $1 - \delta = 0.95$. How many $\widehat{\beta}_j$'s are outside the confidence interval? *Hint: Due to the special structure of how we generated x_i, we can compute $(X^TX)^{-1}$ analytically without computing an inverse explicitly.*

Solution:



We have 56 $\widehat{\beta}_j$'s outside the confidence interval. The percentage of $\widehat{\beta}_j$'s outside the confidence interval is $\frac{56}{10{,}000} \times 100\% = 0.56\%$.

Administrative

A6.

a. *[2 points]* About how many hours did you spend on this homework? There is no right or wrong answer :)

12 hours