Homework #2

CSE 446/546: Machine Learning
Prof. Hunter Schafer & Prof. Matt Golub
Due: February 12, 2024 11:59pm
Points A: 104; B: 20

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

- All code must be written in Python and all written work must be typeset (e.g. LaTeX).
- 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 10% of the value of each question not properly linked. For instructions, see https://www.gradescope.com/get_started#student-submission.

Not adhering to these reminders may result in point deductions.

Important: By turning in this assignment (and all that follow), you acknowledge that you have read and understood the collaboration policy with humans and AI assistants alike: https://courses.cs.washington.edu/courses/cse446/24wi/assignments/. Any questions about the policy should be raised at least 24 hours before the assignment is due. There are no warnings or second chances. If we suspect you have violated the collaboration policy, we will report it to the college of engineering who will complete an investigation.
Not adhering to these reminders may result in point deductions.

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]* Explain why a L1 norm penalty is more likely to result in sparsity (a larger number of 0s) in the weight vector, as compared to the L2 norm.

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

c. *[2 points]* True or False: If the step-size for gradient descent is too large, it may not converge.

d. *[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.

e. *[2 points]* Why is it necessary to apply the gradient descent algorithm on logistic regression but not linear regression?

What to Submit:

- **Part c:** True or False.
- **Parts a-e:** Brief (2-3 sentence) explanation.

Solution:
a L2 norm adopts the quadratic form, and the L1 norm adopts the linear (absolute value) form. With a small coefficient (less than 1), L2 norm will put a even smaller penalty due to the quadratic from, whereas L1 norm will keep the penalty always be the same scale of the coefficient.
b Upside: this regularization put larger penalty for smaller coefficients, result in a very sparse solution if the coefficient is samll per se. Downside: it has smaller penalty for large coefficients (larger than 1), making it harder to penalize these coefficients.
c True. If the step-size is too large, the algorithm might jump over the minimum/maximum point and bounce around.
d SGD is less computationally complex in each step because it only uses one data point at each step; The variance of SGD gradient estimation is larger than GD at each step, resulting in more steps.
e Linear regression has close-form solution but the logistic regression does not.

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 $a, b \in \mathbb{R}$.)

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.)

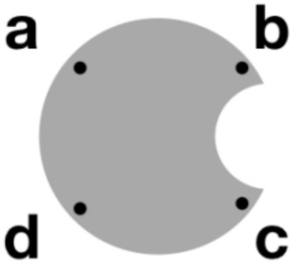
Context: norms are often used in regularization to encourage specific behaviors of solutions. If we define $\|x\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ then one can show that $\|x\|_p$ is a norm for all $p \geq 1$. The important cases of $p = 2$ and $p = 1$ correspond to the penalty for ridge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

What to Submit:

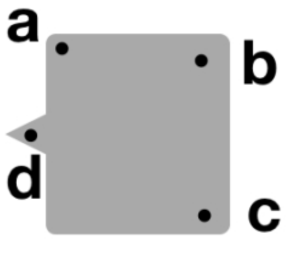
- **Parts a, b:** Proof.

Solution:
a
We first show the $|x+y| \leq |x| + |y|$ for all $x, y \in \mathbb{R}^n$. We can square both side, given that they are all positive, and get $(x+y)^2 \leq (|x|+|y|)^2$, meaning that $x^2 + y^2 + 2xy \leq x^2 + y^2 + 2|x||y|$. As it is always true that $2xy \leq 2|x||y|$. To show that $f(x) = (\sum_{i=1}^n |x_i|)$ is a norm, we show: (i) and (ii) Obviously, it is non-negativity and absolute scalability. Now, we show that (iii) $f(x+y) = (\sum_{i=1}^n |x_i + y_i|) = \sum_{i=1}^n |x_i + y_i| \leq \sum_{i=1}^n (|x_i| + |y_i|) = \sum_{i=1}^n |x_i| + \sum_{i=1}^n |y_i| = f(x) + f(y)$. Thus, $f(x)$ is a norm.
b
For $x, y \in \mathbb{R}^n$ and $n = 2$, we have $g(x+y) = \left(\sum_{i=1}^2 |x_i + y_i|^{1/2}\right)^2 = \left(|x_1 + y_1|^{1/2} + |x_2 + y_2|^{1/2}\right)^2 = |x_1 + y_1| + |x_2 + y_2| + 2 * (|x_1 + y_1||x_2 + y_2|)^{1/2}$
and $g(x) = \left(\sum_{i=1}^2 |x_i|^{1/2}\right)^2 = \left(|x_1|^{1/2} + |x_2|^{1/2}\right)^2 = |x_1| + |x_2| + 2(|x_1||x_2|)^{1/2}, g(y) = \left(\sum_{i=1}^2 |y_i|^{1/2}\right)^2 = \left(|y_1|^{1/2} + |y_2|^{1/2}\right)^2 = |y_1| + |y_2| + 2(|y_1||y_2|)^{1/2}$ Thus, we have $g(x) + g(y) = |x_1| + |x_2| + 2(|x_1||x_2|)^{1/2} + |y_1| + |y_2| + 2(|y_1||y_2|)^{1/2}$
When $x = (0,1), y = (1,0)$, we have $g(x+y) = 1 + 1 + 2 * 1 = 4$ and $g(x) + g(y) = 0 + 1 + 0 + 1 + 0 + 0 = 2$. Thus, $g(x+y) > g(x) + g(y)$ and $g(x)$ is not a norm.

A3. *[2 points]* A set $A \subseteq \mathbb{R}^n$ is *convex* if $\lambda x + (1-\lambda) y \in A$ for all $x, y \in A$ and $\lambda \in [0,1]$. For each of the grey-shaded sets below (I-II), state whether each one is convex, or state why it is not convex using any of the points a, b, c, d in your answer.



I



II

What to Submit:

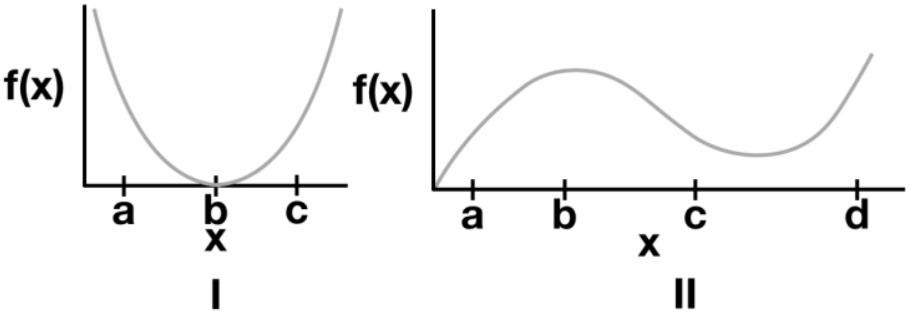
- **Parts I, II:** 1-2 sentence explanation of why the set is convex or not.

Solution:
a
The set I is not convex because the line between b and c is out of the set.
b
The set II is not convex because the line between a and d is out of the set.

A4. *[2 points]* We say a function $f : \mathbb{R}^d \to \mathbb{R}$ is convex on a set A if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ for all $x, y \in A$ and $\lambda \in [0,1]$. For each of the functions shown below (I-II), state whether each is convex on the specified interval, or state why not with a counterexample using any of the points a, b, c, d in your answer.

a. Function in panel I on $[a, c]$

b. Function in panel II on $[a, d]$



What to Submit:

- **Parts a, b:** 1-2 sentence explanation of why the function is convex or not.

Solution:

a

Function in panel I is convex as any line between two points on the function is over the function between the two points.

b

Function in panel II is not convex as the line between $f(x)$ where $x \in [a, c]$ is below the function $f(x)$.

B1. Use just the definitions above and let $\|\cdot\|$ be a norm.

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

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

c. *[2 points]* Draw a picture of the set $\{(x_1, x_2) \; : \; g(x_1, x_2) \leq 4\}$ 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?

Context: It is a fact that a function f defined over a set $A \subseteq \mathbb{R}^n$ is convex if and only if the set $\{(x, z) \in \mathbb{R}^{n+1} : z \geq f(x), x \in A\}$ is convex. Draw a picture of this for yourself to be sure you understand it.

What to Submit:

- **Parts a, b:** Proof.
- **Part c:** A picture of the set, and 1-2 sentence explanation.

Solution:

a

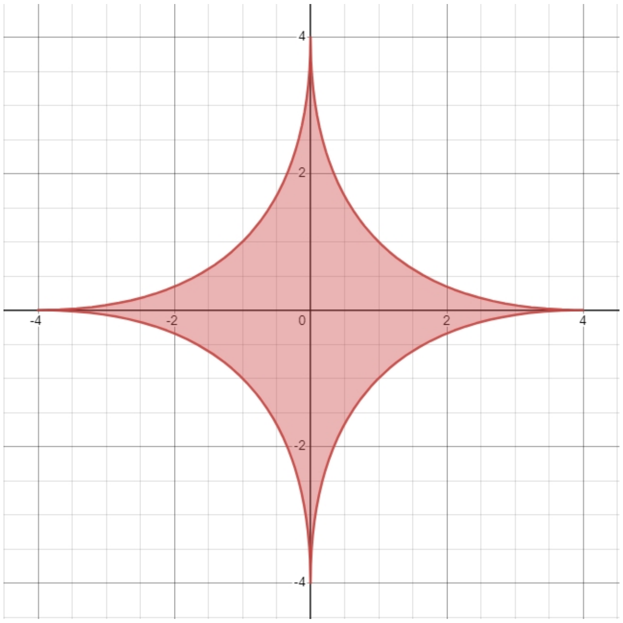
Given that $f(x)$ is a norm, we have $f(\lambda x + (1-\lambda)y) \leq |\lambda| f(x) + |1-\lambda| f(y)$. Given that $\lambda \in [0,1] \geq 0$, we have $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$, meaning that $f(x)$ is a convex function.

b

For any $x, y \in \mathbb{R}^n$ $\|x\| \leq 1, \|y\| \leq 1$, $\|\lambda x + (1-\lambda)y\| \leq \lambda\|x\| + (1-\lambda)\|y\| \leq 1$ Thus, $\|\lambda x + (1-\lambda)y\| \leq 1$ and the set is a convex set.

c

Please see the figure below. It is not a convex set as the line between (0,4) and (4,0) is out of the set.



Lasso on a Real Dataset

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, you will implement the iterative shrinkage thresholding algorithm shown in Algorithm 1 to solve the Lasso problem in `ISTA.py`. This is a variant of the subgradient descent method and a more detailed discussion can be found in these slides. You may use common computing packages (such as `numpy` or `scipy`), but do not use an existing Lasso solver (e.g., of `scikit-learn`).

Algorithm 1: Iterative Shrinkage Thresholding Algorithm for Lasso

```
Input: Step size $\eta$
while not converged do
    $b' \leftarrow b - 2\eta \sum_{i=1}^{n} (x_i^T w + b - y_i)$
    for $k \in \{1, 2, \cdots d\}$ do
        $w'_k \leftarrow w_k - 2\eta \sum_{i=1}^{n} x_{i,k}(x_i^T w + b - y_i)$
        $w'_k \leftarrow \begin{cases} w'_k + 2\eta\lambda & w'_k < -2\eta\lambda \\ 0 & w'_k \in [-2\eta\lambda, 2\eta\lambda] \\ w'_k - 2\eta\lambda & w'_k > 2\eta\lambda \end{cases}$
    end
    $b \leftarrow b'$, $w \leftarrow w'$
end
```

Before you get started, be sure to read the following:

- Wherever possible, use matrix libraries for matrix operations (not `for` loops). This especially applies to computing the updates for w. While we wrote the algorithm above with a `for` loop for clarity, you should be able to replace this loop using equivalent matrix/vector operations in your code (e.g., `numpy` functions).
- As a sanity check, ensure the objective value is nonincreasing with each step.
- It is up to you to decide on a suitable stopping condition. A common criteria is to stop when no element of w changes by more than some small δ during an iteration. If you need your algorithm to run faster, an easy place to start is to loosen this condition.
- You will need to solve the Lasso on the same dataset for many values of λ. This is called a regularization path. To do this efficiently you will start at a large λ, and then for each consecutive solution, initialize the algorithm with the previous solutions $\hat{w}$ and $\hat{b}$, 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.

A5. We will first try out your solver with some synthetic data. 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. Suppose that $x \in \mathbb{R}^d, y \in \mathbb{R}, k < d$, and data are generated independently according to the model $y_i = w^T x_i + \epsilon_i$ where

$$w_j = \begin{cases} j/k & \text{if } j \in \{1, \ldots, k\} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is noise (note that in the model above $b = 0$). We can see from Equation (2) that since $k < d$ and $w_j = 0$ for $j > k$, the features $k + 1$ through d are irrelevant for predicting y.
Generate a dataset using this model with $n = 500, d = 1000, k = 100$, and $\sigma = 1$. You should generate the dataset such that each $\epsilon_i \sim \mathcal{N}(0, 1)$, and y_i is generated as specified above. You are free to choose a distribution from which the x's are drawn, but make sure standardize the x's before running your experiments.

a. *[10 points]* With your synthetic data, solve multiple Lasso problems on a regularization path, starting at λ_{max} where no features are selected (see Equation (1)) and decreasing λ by a constant ratio (e.g., 2) until nearly all the features are chosen. In plot 1, plot the number of non-zeros as a function of λ on the x-axis (Tip: use `plt.xscale('log')`).

b. *[10 points]* For each value of λ tried, record values for false discovery rate (FDR) (number of incorrect nonzeros in $\widehat{w}$/total number of nonzeros in $\widehat{w}$) and true positive rate (TPR) (number of correct nonzeros in $\widehat{w}$/k). Note: for each j, $\widehat{w}_j$ is an incorrect nonzero if and only if $\widehat{w}_j \neq 0$ while $w_j = 0$. In plot 2, plot these values with the x-axis as FDR, and the y-axis as TPR.

   Note that in an ideal situation we would have an (FDR,TPR) pair in the upper left corner. We can always trivially achieve $(0, 0)$ and $(\frac{d-k}{d}, 1)$.

c. *[5 points]* Comment on the effect of λ in these two plots in 1-2 sentences.

What to Submit:

- **Part a:** Plot 1.
- **Part b:** Plot 2.
- **Part c:** 1-2 sentence explanation.
- **Code** on Gradescope through coding submission

Solution:
a

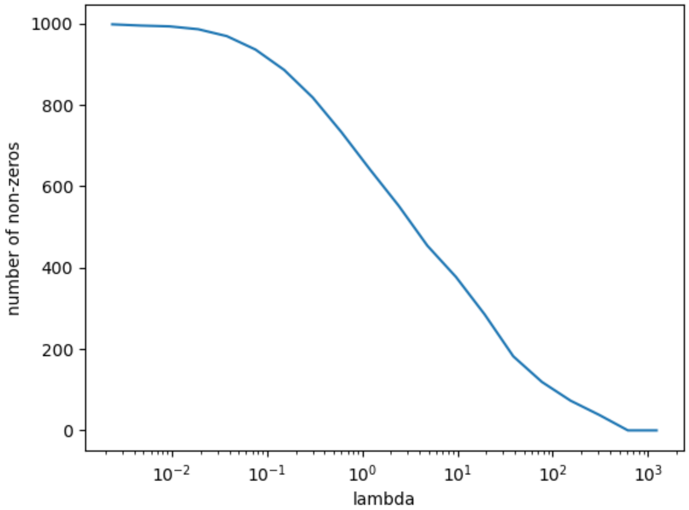


Figure 1: number of nonzeros and lambda

b

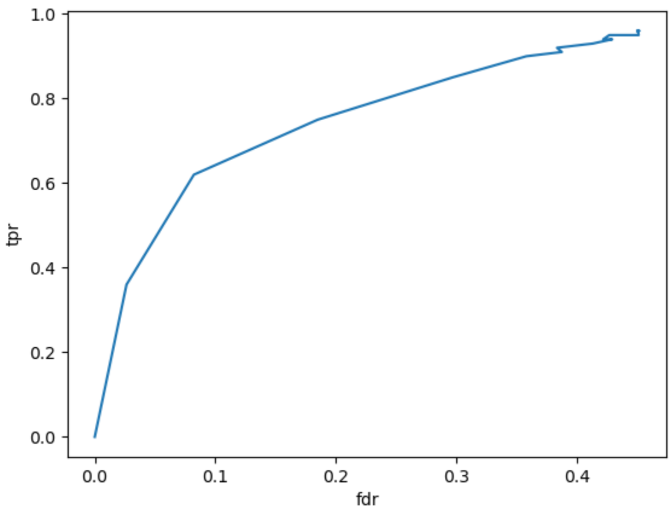


Figure 2: TPR and FDR

c As we can see from Figure 1, the larger lambda is. the more features will be zero. From Figure 2, we can see that with too small lambda, we have a high FDR, while with too large lambda, we have a low TPR. Thus, we need to find an appropriate lambda to get a sparse but accurate model.

A6. We'll now put the Lasso to work on some real data in `crime_data_lasso.py`. We have read in the data for you with the following:

```
df_train, df_test = load_dataset("crime")
```

This stores the data as Pandas `DataFrame` objects. `DataFrame`s are similar to Numpy `arrays` but more flexible; unlike `arrays`, `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 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.**

The dataset is split into a training and test set with 1,595 and 399 entries, respectively[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. In order to avoid this, use the ISTA Lasso algorithm implemented in the previous problem.

a. *[4 points]* Read the documentation for the originalcversion 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. Provide a short (1-3 sentence) explanation.

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. Provide a short (1-3 sentence) explanation.

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 $\hat{w}_0$ and $\hat{b}_0$ as the $\hat{w}$ and $\hat{b}$ found at the end

[1] The features have been standardized to have mean 0 and variance 1.

of your previous iteration. Continue the process 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.

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

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

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. Retrain and inspect the weights $\hat{w}$ for $\lambda = 30$ and for *all* input variables. Which feature had the largest (most positive) Lasso coefficient? What about the most negative? Discuss briefly.

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?)

What to Submit:

- **Parts a, b:** 1-2 sentence explanation.
- **Part c:** Plot 1.
- **Part d:** Plot 2.
- **Part e:** Plot 3.
- **Parts f, g:** Answers and 1-2 sentence explanation.
- **Code** on Gradescope through coding submission.

Solution:
a
The number of homeless people on the street or in the shelter (NumInShelters and NumStreet) is likely to be influenced by the policies, such as the support of shelters. The number of policemen (LemasSwornFT) is also likely to be influenced by policies, like financial support.
b
The family income may (medIncome) be a cause of higher levels of violent crime, because they may spend more money on security. The number of requests for policemen (LemasTotalReq and LemasTotReqPerPop) may be a result of higher levels of violent crime because the request is unlikely to cause the violent crime.

c

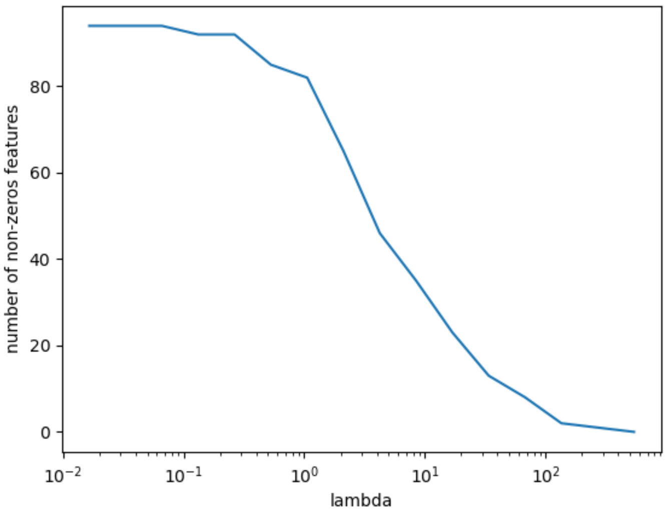


Figure 3: number of nonzeros and lambda

d

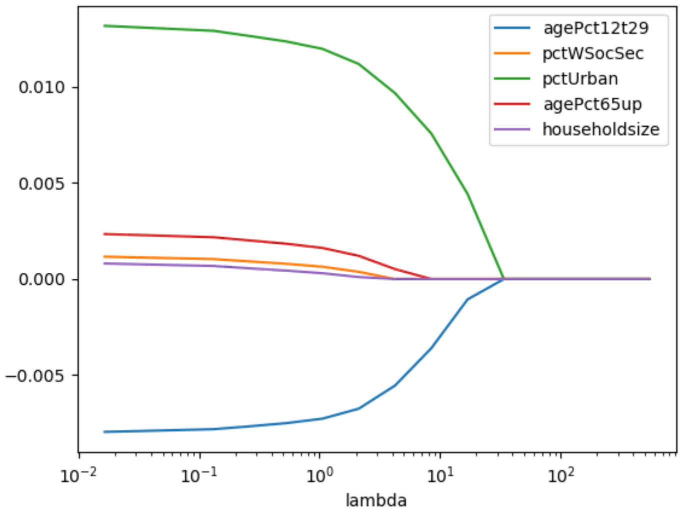


Figure 4: regularization path

e

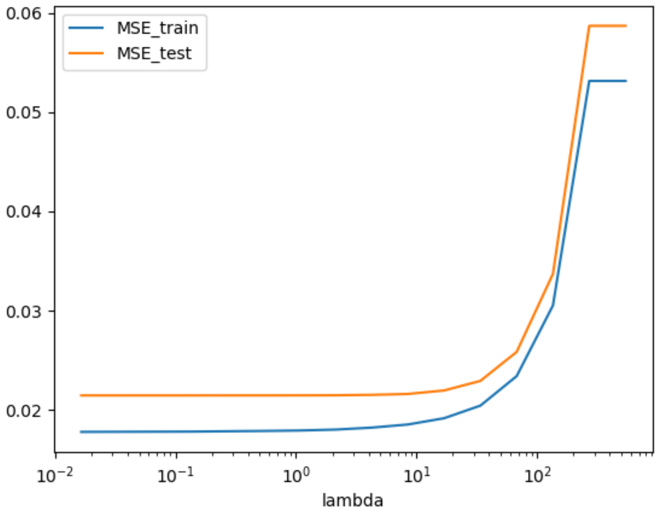


Figure 5: mean squared error

f
The most positive coefficient is PctIlleg (percentage of kids born to never married (numeric - decimal)) and its value is 0.0534562316558457. The most negative coefficient is PctKids2Par (percentage of kids in family housing with two parents (numeric - decimal)) and its value is -0.03355697788140728.

g
It is not that people over 65 decrease the crime rate but these people prefer to live in community with low crime rate.

Logistic Regression

A7. 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).

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.

   (i) 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).

   (ii) 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).

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 learning rate.

d. *[7 points]* Repeat (b) using mini-batch 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).

What to Submit

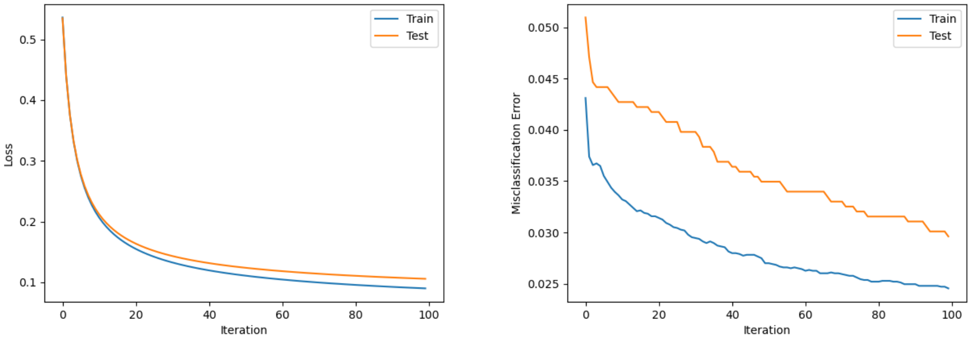
- **Part a:** Proof
- **Part b:** Separate plots for b(i) and b(ii).
- **Part c:** Separate plots for c which reproduce those from b(i) and b(ii) for this case.
- **Part d:** Separate plots for c which reproduce those from b(i) and b(ii) for this case.
- **Code** on Gradescope through coding submission.

Solution:
a

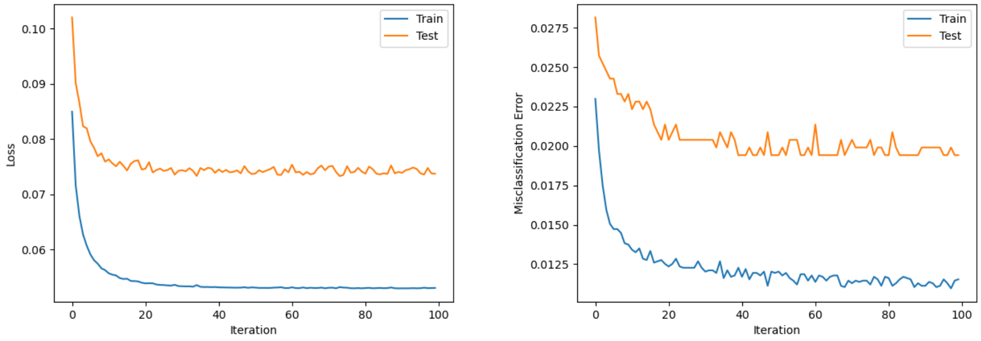
$$\begin{aligned}
\nabla_w J(w, b) &= \frac{1}{n} \sum_i \nabla_w \log(1 + \exp(-y_i(b + x_i^T w))) + \nabla_w \lambda \|w\|^2 \\
&= \frac{1}{n} \sum_i \mu_i(w, b)(\frac{1}{\mu_i(w, b)} - 1)(-y_i x_i) + 2\lambda w \\
&= \frac{1}{n} \sum_i (\mu_i(w, b) - 1) y_i x_i + 2\lambda w
\end{aligned}$$

$$\begin{aligned}\nabla_b J(w,b) &= \frac{1}{n}\sum_i \nabla_b \log(1+\exp(-y_i(b+x_i^T w))) + \nabla_b \lambda\|w\|^2 \\ &= \frac{1}{n}\sum_i \mu_i(w,b)(\frac{1}{\mu_i(w,b)}-1)(-y_i) \\ &= \frac{1}{n}\sum_i (\mu_i(w,b)-1)y_i\end{aligned}$$

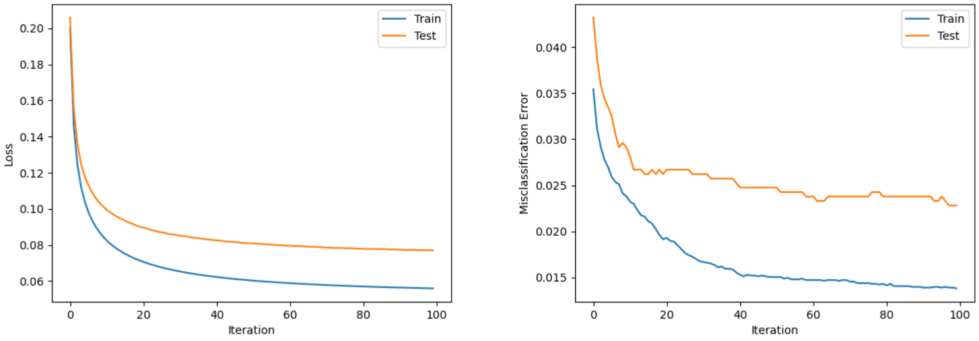
b For gradient descent, we have



c For stochastic gradient descent, we have



d For mini-batch gradient descent, we have



Bounding the Estimate

B2. 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.)*

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?

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 $j \in \{1, \ldots, d\}$. 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.*

What to Submit:

- **Parts a, b:** Proof.
- **Part b:** Answer.
- **Part c:** Plots of $\hat{\beta}$ and its confidence interval **on the same plot**.

Solution:
a

$$\widehat{\beta} = \min_\beta \|X\beta - Y\|_2^2 = (X^TX)^{-1}X^TY = (X^TX)^{-1}X^T(X\beta^* + \epsilon) = \beta^* + (X^TX)^{-1}X^T\epsilon$$

Given that $\epsilon \sim \mathcal{N}(0, I)$ and the proposition 1, we have

$$\widehat{\beta}) \sim \mathcal{N}(\beta^*, (X^TX)^{-1})$$

b
Given the proposition 2

$$P\left[|z^T(\widehat{\beta} - \beta^*)| \geq \sqrt{2\sigma^2 z^T(X^TX)^{-1}z \log(2/\delta)}\right] \leq \delta.$$

In our case, $\sigma = 1$ and let $z = 1$, we have

$$P\left[|\widehat{\beta} - \beta^*| \geq \sqrt{2(X^TX)^{-1} \log(2/\delta)}\right] \leq \delta.$$

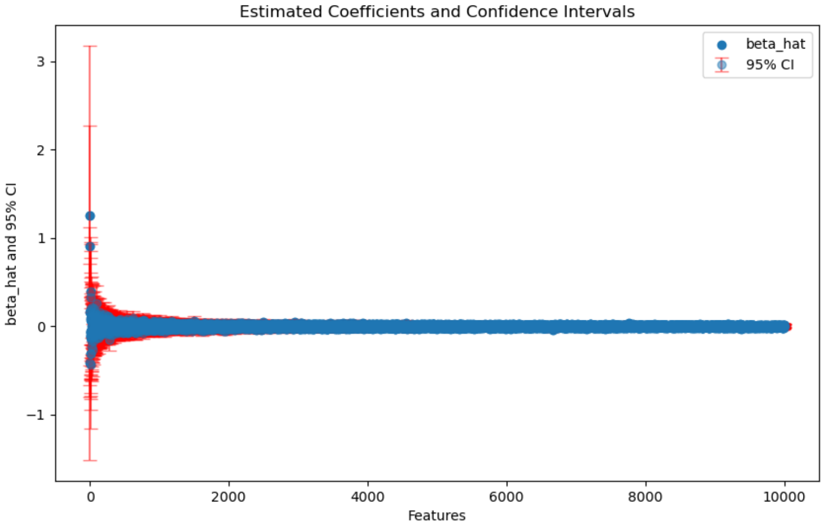
Thus,

$$P\left[|\widehat{\beta}-\beta^*| \leq \sqrt{2(X^TX)^{-1}\log(2/\delta)}\right] = 1 - P\left[|\widehat{\beta}-\beta^*| \geq \sqrt{2(X^TX)^{-1}\log(2/\delta)}\right] \geq 1-\delta.$$

We cannot conclude this for all $j \in [d]$ simultaneously, because that will also require us to account for d, meaning that

$$P\left[\bigcup_{i=1}^{d} |\widehat{\beta}_i - \beta_i^*| \geq \sqrt{2(X^TX)_{i,i}^{-1}\log(2d/\delta)}\right] \leq \delta.$$

c



Administrative

A8.

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

40 hours