Homework #2

CSE 446/546: Machine Learning
Prof. Kevin Jamieson
Due: November 6, 2023 11:59pm
Points A: 104; B: 20

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.

Answers

a. Since L1 norm penalty leads to many coefficient to become zeros. While L2 norm penalty with small but non-zero coefficient.

b. Upsie: provide zero sparsity like L1 norm. Downside: this regularizer changes faster and faster when close to zero, may complicate garadient computation.

c. True. The algorithm may overshot the optimal solution if the step-size is too large. It may bounce back and forth across the minimal solution.

d. SGD converge faster than GD. Randomness in SGD may have issue like inconsistent result.

e. Logistic regression is non-linear non-convex function thus does not have a closed form. It needs GD to iterate find optimal sulution.

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 $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ 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.

Answers

a. $f(x) = \sum_{i=1}^n |x_i| = |x_1| + |x_2| + \cdots + |x_n|$. Since each term is a non negative value, we conclude that $f(x) \geq 0$.

$$\begin{aligned} f(ax) &= \sum_{i=1}^n |ax_i| \\ &= |ax_1| + |ax_2| + \cdots + |ax_n| \\ &= a|x_1| + a|x_2| + \cdots + a|x_n| \\ &= a(|x_1| + a|x_2| + \cdots + a|x_n|) \\ &= af(x) \end{aligned}$$

b. We'll try to prove by finding a counterexample. First, let $n = 2$, assume we have two positive vector x, y.

$$\begin{aligned} g(x) &= (\sum_{i=1}^n |x_i|^{1/2})^2 \\ &= (\sqrt{|x_1|} + \sqrt{|x_2|})^2 \\ &= x_1 + x_2 + 2\sqrt{x_1 x_2} \end{aligned}$$

Similarly, $g(y) = y_1 + y_2 + 2\sqrt{y_1 y_2}$

$$\begin{aligned} g(x) + g(y) &= x_1 + x_2 + 2\sqrt{x_1 x_2} + y_1 + y_2 + 2\sqrt{y_1 y_2} \\ &= (x_1 + y_1) + (x_2 + y_2) + 2(\sqrt{x_1 x_2} + \sqrt{y_1 y_2}) \end{aligned}$$

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

If $g(x)$ is a norm, the triangle inequality should hold.

$$\begin{aligned}
g(x) + g(y) &\geq g(x+y) \\
(x_1 + y_1) + (x_2 + y_2) + 2(\sqrt{x_1 x_2} + \sqrt{y_1 y_2}) &\geq (x_1 + y_1) + (x_2 + y_2) + 2\sqrt{x_1 x_2 + y_1 y_2 + x_1 y_2 + x_2 y_1} \\
\sqrt{x_1 x_2} + \sqrt{y_1 y_2} &\geq \sqrt{x_1 x_2 + y_1 y_2 + x_1 y_2 + x_2 y_1} \\
x_1 x_2 + y_1 y_2 + 2\sqrt{x_1 x_2 y_1 y_2} &\geq x_1 x_2 + y_1 y_2 + x_1 y_2 + x_2 y_1 \\
2\sqrt{x_1 x_2 y_1 y_2} &\geq x_1 y_2 + x_2 y_1 \\
0 &\geq x_1 y_2 + x_2 y_1 - 2\sqrt{x_1 x_2 y_1 y_2} \\
0 &\geq (\sqrt{x_1 y_2} + \sqrt{x_2 y_1})^2
\end{aligned}$$

However, $(\sqrt{x_1 y_2} + \sqrt{x_2 y_1})^2 > 0$ is always true. This violates the trianglar inequality. So that $g(x)$ 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.

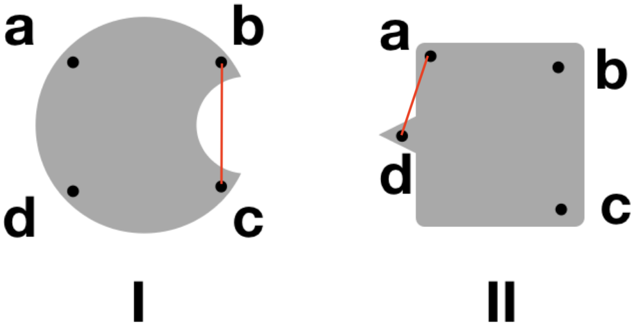


Figure 1: Convex Shapes

What to Submit:

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

Answers

- Both set **I, II** are not convex. For set **I**, part of the line between point b and c lies outside of the set. While for set II, draw a line between point a and d, we can see not all points on the line are within the domain of the set. See figure 1

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.

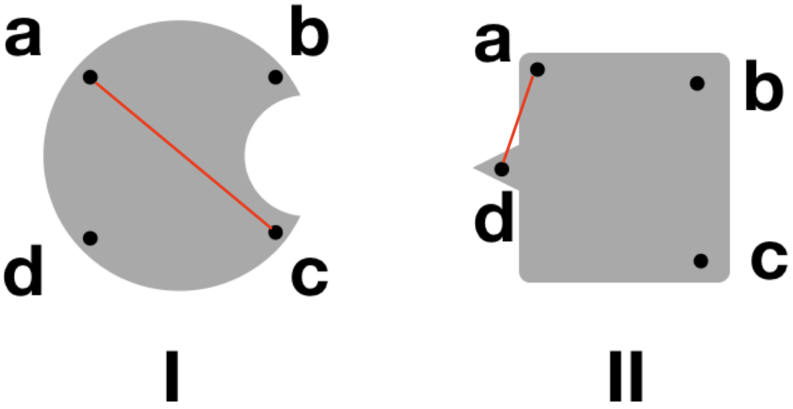


Figure 2: Convex Functions

Answers

- In panel I on $[a, c]$, the line segment lies within the set (Figure 2). Which means $\lambda x + (1 - \lambda)y$ in the set, so that f is convex in set I.
- In panel II on $[a, d]$, we find part of the line segment lies outside of the set ((Figure 2). We can find points that let $\lambda x + (1 - \lambda)y$ not in the set, so f is not convex in this set.

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

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

and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is noise (note that in the model above $b = 0$). We can see from Equation (1) 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 (**??**)) 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')`).

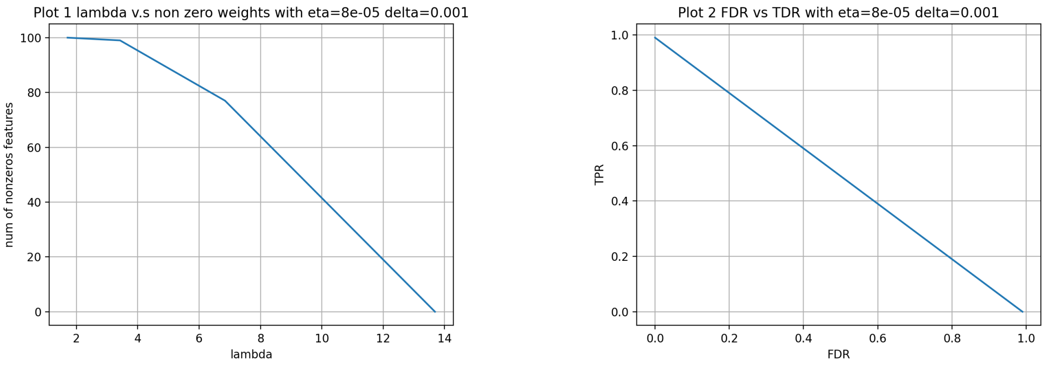
b. *[10 points]* For each value of λ tried, record values for false discovery rate (FDR) (number of incorrect nonzeros in $\widehat{w}$/total number of nonzeros in $\widehat{w}$) and true positive rate (TPR) (number of correct nonzeros in $\widehat{w}$/k). Note: for each j, $\widehat{w}_j$ is an incorrect 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
- **All code you wrote** in the write-up, with correct page mapping.



(a) Plot 1

(b) Plot 2

Figure 3: A5

a. As λ increases, the penalty for non-zero coefficients becomes more severe, pushing more coefficients to become zero, thereby reducing the number of features that contribute to the model. When $\lambda$is small enough, the penalty effect is minimal and the Lasso model behaves almost like a regular linear regression, typically with many non-zero coefficients.

b. Figure 3b shows as the of number of correct nonzeros increases, the incorrect nonzeros in $\widehat{w}$/total number decreasing. The two are negatively correlated.

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. `DataFrames` are similar to Numpy `arrays` but more flexible; unlike `arrays`, `DataFrames` 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.

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

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 (**??**). 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.

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.
- **All code you wrote** in the write-up, with correct page mapping.

0.1 Answers

a. `Percent of Police that are African American`: Government efforts to increase diversity in law enforcement can affect this percentage. `Percent of Population Who Have Immigrated within the Last 3 Years`: Immigration policies and reforms, such as the Immigration and Nationality Act amendments, impact the number and demographics of new immigrants. `Percentage of Males Who Have Never Married`: The benefits and disadvantages that may arise from the enactment and revision of the provisions of marriage laws, such as the cost of divorce, may affect whether people get married.

b. `Takeotal requests for police per police officer`, `percent of officers assigned to drug units`, and `police operating budget`. These factors may be both causes and results of violence; for instance, higher violence rates may lead to more police requests and a larger portion of officers in drug units, while effective use of police resources might reduce violence.

c.

d.

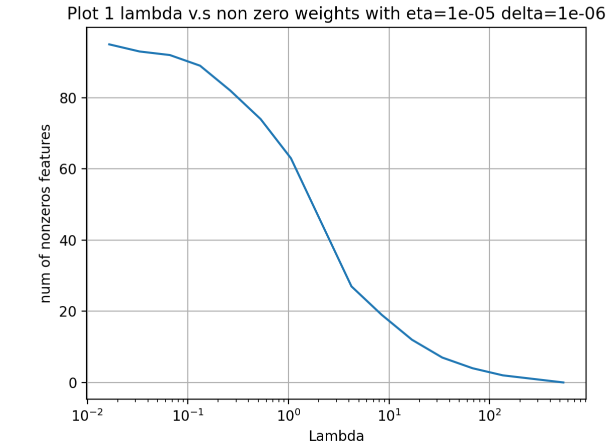


Figure 4: The number of nonzero weights of each solution as a function of λ

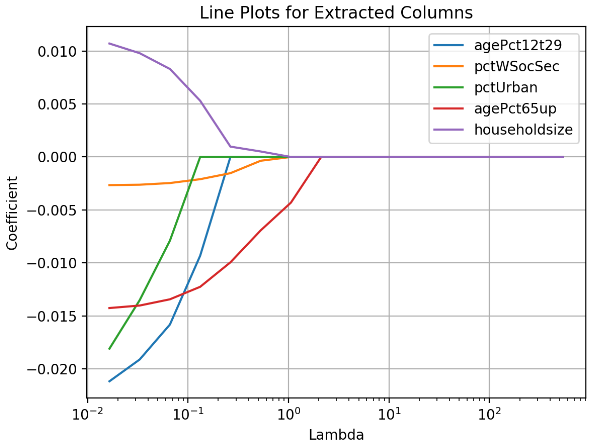


Figure 5: Plot the regularization paths

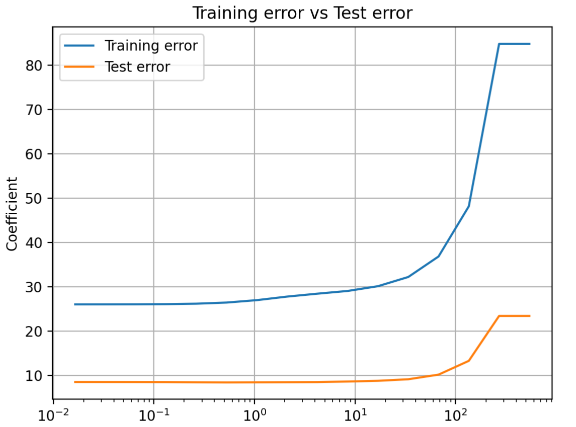


Figure 6: mean squared error on the training and test data as a function of λ

e.

f. PersPerOccupHous (mean persons per household) suggests that there is a positive correlation between the number of people living in a household and the crime rate, This could potentially be explained by socioeconomic factors; for example, higher household occupancy might be correlated with lower income or socioeconomic status, which in turn could be related to higher crime rates in some areas. However, a negative coefficient for PctKids2Par (percentage of kids in family housing with two parents) indicates a negative correlation with the crime rate. This could be interpreted to mean that family stability or parental supervision, which is more likely in two-parent households, may be associated with lower crime rates.

g. The negative weight on agePct65up in the Lasso regression model indicates that there is a correlation between higher percentages of the population being over the age of 65 and lower crime rates in the data used to train the model. The presence of people over 65 and the crime rate, it does not imply that one causes the other. It's possible that lower crime rates cause more people over 65 to move to or remain in an area, rather than the other way around.

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 regularizer.

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

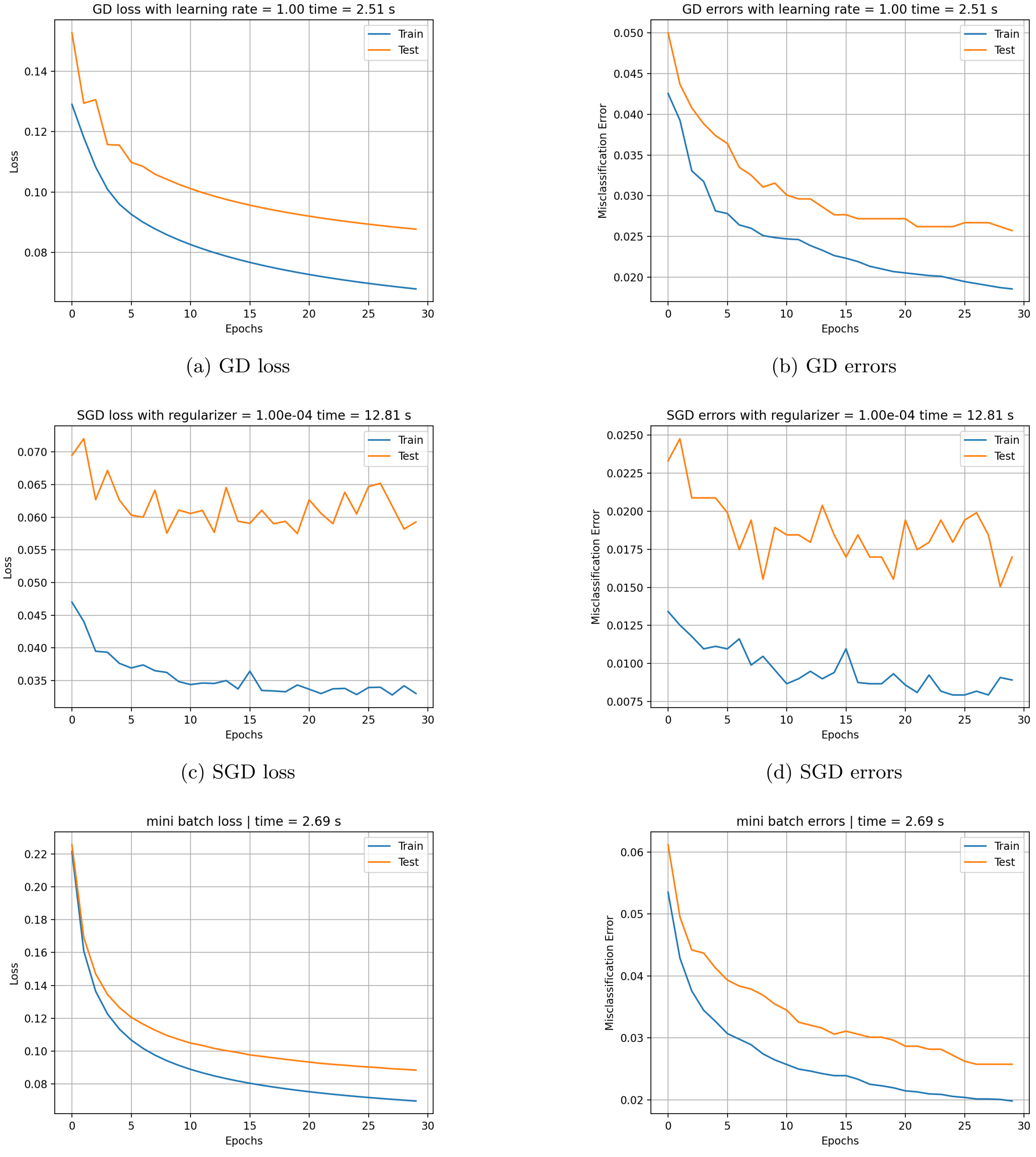
- **Part a:** Proof
- **Part b:** Separate plots for b(i) and b(ii).
- **Part c:** Separate plots for c which reproduce those from b(i) and b(ii) for this case.
- **Part d:** Separate plots for c which reproduce those from b(i) and b(ii) for this case.
- **Code** on Gradescope through coding submission.
- **All code you wrote in the write-up, with correct page mapping.**

0.2 Answers

a.

$$\begin{aligned}
J(w,b) &= \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i(b+x_i^Tw))) + \lambda||w||_2^2 \\
\nabla_w J(w,b) &= \nabla_w\Big(\frac{1}{n}\sum_{i=1}^{n}\log(\exp(-y_i(b+x_i^Tw))) + \lambda||w||_2^2\Big) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\nabla_w\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))} + 2\lambda w \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))}\nabla_w(-y_i(b+x_i^Tw)) + 2\lambda w \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))}(-y_ix_i) + 2\lambda w \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1+\exp(-y_i(b+x_i^Tw))-1}{1+\exp(-y_i(b+x_i^Tw))}(-y_ix_i) + 2\lambda w \\
&= \frac{1}{n}\sum_{i=1}^{n}(1-\frac{1}{1+\exp(-y_i(b+x_i^Tw))})(-y_ix_i) + 2\lambda w \\
\nabla_w J(w,b) &= \frac{1}{n}\sum_{i=1}^{n}(1-\mu_i(w,b))(-y_ix_i) + 2\lambda w
\end{aligned}$$

$$\begin{aligned}
\nabla_b J(w,b) &= \nabla_b\Big(\frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i(b+x_i^Tw))) + \lambda||w||_2^2\Big) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\nabla_b\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))} + 0 \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))}\nabla_b(-y_i(b+x_i^Tw)) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i(b+x_i^Tw))}{1+\exp(-y_i(b+x_i^Tw))}(-y_i) \\
\nabla_b J(w,b) &= \frac{1}{n}\sum_{i=1}^{n}(1-\mu_i(w,b))(-y_i)
\end{aligned}$$



(a) GD loss

(b) GD errors

(c) SGD loss

(d) SGD errors

(e) mini-batch loss

(f) mini-batch errors

Figure 7: A7

Administrative

A8.

a. *[2 points]* About how many hours did you spend on this homework? There is no right or wrong answer :)
About 40 hours