CSCI567 Homework 1

Mengxiao Zhang
2024.2.9

Problem 1

Assume there exists an optimal hyperplane $\mathbf{w}_{\text{opt}}, \|\mathbf{w}_{\text{opt}}\|_2 = 1$ and some $\gamma > 0$ such that $y_i \cdot (\mathbf{w}_{\text{opt}}^T \mathbf{x}_i) \geq \gamma, \forall i \in \{1, 2, \ldots, n\}$. Additionally, assume $\|\mathbf{x}_i\|_2 \leq R, \forall i \in \{1, 2, ..., n\}$ for some $R > 0$. Following the steps below, show that the perceptron algorithm makes at most $\frac{R^2}{\gamma^2}$ errors, and therefore the algorithm must converge.

1.1 (7pts) Show that if the algorithm makes a mistake, the update rule moves the parameter $\mathbf{w}$ towards the direction of the optimal weights $\mathbf{w}_{\text{opt}}$. Specifically, denoting explicitly the updating iteration index by k, the current weight vector by $\mathbf{w}_k$, and the updated weight vector by $\mathbf{w}_{k+1}$, show that, if $y_i(\mathbf{w}_k^T \mathbf{x}_i) < 0$ where $(\mathbf{x}_i, y_i)$ is the instance selected in this iteration, we have

$$\mathbf{w}_{k+1}^T \mathbf{w}_{\text{opt}} \geq \mathbf{w}_k^T \mathbf{w}_{\text{opt}} + \gamma \|\mathbf{w}_{\text{opt}}\|_2. \tag{1}$$

$w_{k+1} = w_k + y_i \cdot x_i$ ← data at k-th iter.

$w_{opt}^T w_{k+1} = w_{opt}^T w_k + y_i \cdot x_i^T w_{opt}$ (with $y_i \cdot x_i^T w_{opt} \geq \gamma$)

$\geq w_{opt}^T w_k + \gamma$

$= w_{opt}^T w_k + \gamma \|w_{opt}\|_2$

Problem 1

1.2 (5pts) Show that the length of updated weights does not increase by a large amount. In particular, show that if $y_i(\mathbf{w}_k^T \mathbf{x}_i) < 0$, then

$$\|\mathbf{w}_{k+1}\|_2^2 \leq \|\mathbf{w}_k\|_2^2 + R^2. \tag{2}$$

$$\|w_{k+1}\|_2^2 = w_{k+1}^T w_{k+1}$$

$$= (w_k + y_i x_i)^T (w_k + y_i x_i)$$

$$= \|w_k\|_2^2 + \underbrace{2 y_i \cdot x_i^T w_k}_{<0} + \underbrace{y_i^2 \cdot x_i^T x_i}_{1 \quad R^2}$$

$$\leq \|w_k\|_2^2 + R^2$$

Problem 1

1.3 (6pts) Assume that the initial weight vector $\mathbf{w}_0 = \mathbf{0}$ (an all-zero vector). Using results from the previous two parts, show that for any iteration $k+1$, with M being the total number of mistakes the algorithm has made for the first $k+1$ iterations, we have

$$\gamma M \leq \|\mathbf{w}_{k+1}\|_2 \leq R\sqrt{M}. \tag{3}$$

1.4 (2pts) Use **1.3** to conclude that $M \leq R^2/\gamma^2$. Therefore the algorithm can make at most R^2/γ^2 mistakes (note that there is no direct dependence on the dimensionality of the data points).

Problem 2

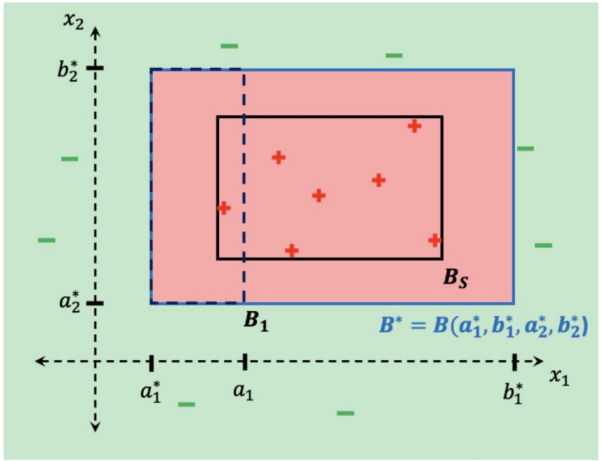


Figure 1: Learning axis-aligned rectangles in two dimensions, here $+$ (red) denotes datapoints in training set S with label 1 and $-$ (green) denotes datapoints with label -1. The true labels are given by rectangle B^* (solid blue line), with everything outside B^* being labelled negative and inside being labelled positive. B_S (solid black line) is the rectangle learned by the algorithm in Part (a). B_1 (dashed black line) is defined in Part (c).

Problem 2

2.1 (4pts) We will follow the general supervised learning framework from class. For a function $f_{(a_1,b_1,a_2,b_2)}$, define the empirical risk with respect to 0-1 loss as $\widehat{\mathcal{R}}(f_{(a_1,b_1,a_2,b_2)}) = \sum_{i=1}^{n} \mathbb{I}\{f_{(a_1,b_1,a_2,b_2)}(\mathbf{x}_i) \neq y_i\}$. Given the 0-1 loss, and the function class of axis-aligned rectangles, we want to find an empirical risk minimizer. Consider the algorithm which returns the smallest rectangle enclosing all positive examples in the training set. Prove that this algorithm is an empirical risk minimizer, meaning that it minimize $\widehat{\mathcal{R}}(f)$ over all f in $\mathcal{F}^2_{\text{rec}}$. (*Hint: use the realizability assumption.*)

X: the set of all points.

S: training set.

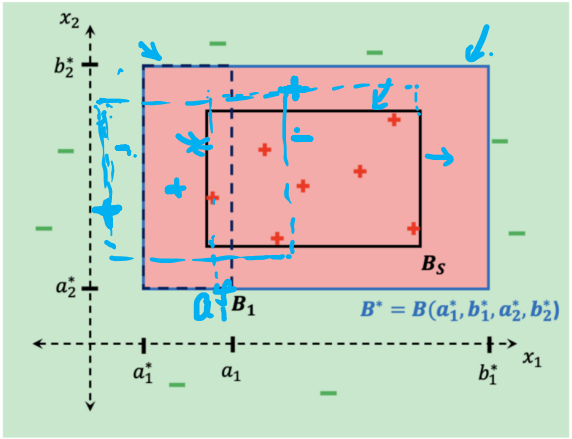
$f = A(S)$: B^* $\quad R(B^*) = 0 \Rightarrow \hat{R}(B^*) = 0$

① no training data with + will be classified as −

if $\exists\ x$ s.t. $A(S)(x) = -$

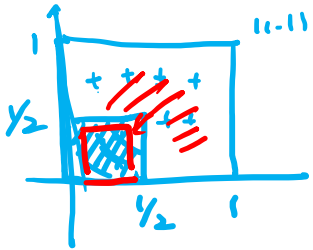
$x_1 < a_1^f$, $a_1^f \le x_1$

② no true data with − will be classified as +



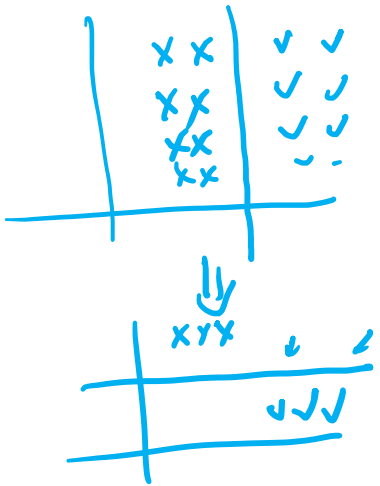
Problem 2

2.2 (4pts) Our next task is to show that the algorithm from the previous part not only does well on the training data, but also gets small classification error with respect to the distribution D. Let B_S be the rectangle returned by the algorithm in **2.1** on training set S, and let f_S^{ERM} be the corresponding hypothesis. First, we will convince ourselves that generalization is inherently a probabilistic statement. Let a *bad* training set S' be a training set such that $R(f_{S'}^{ERM}) \geq 0.5$. Pick some simple distribution D and ground-truth rectangle B^*, and give a short explanation for why there is a *non-zero* probability of seeing a bad training set.



$\geq \frac{3}{4} > \frac{1}{2}$

$\left[\left(\frac{1}{2}\right) \times \left(\frac{1}{2}\right)\right]^n = \frac{1}{4^n} > 0$



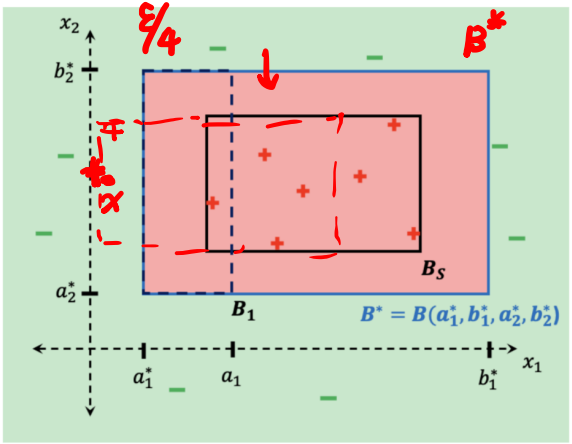
Problem 2

2.3 (8pts) Though there is non-zero probability seeing bad training set, we will now show that *with high probability* over the training dataset S, f_S^{ERM} *does* get small error. Show that if $n \geq \frac{4\log(4/\delta)}{\epsilon}$ then with probability at least $1-\delta$, $R(f_S^{ERM}) \leq \epsilon$.

To prove this follow the following steps. Let $a_1 \geq a_1^*$ be such that the probability mass (with respect to D) of the rectangle $B_1 = B(a_1^*, a_1, a_2^*, b_2^*)$ is exactly $\epsilon/4$. Similarly, let $b_1 \leq b_1^*, a_2 \geq a_2^*, b_2 \leq b_2^*$ be numbers such that the probability mass (with respect to D) of the rectangles $B_2 = B(b_1, b_1^*, a_2^*, b_2^*), B_3 = B(a_1^*, b_1^*, a_2^*, a_2), B_4 = B(a_1^*, b_1^*, b_2, b_2^*)$ are all exactly $\epsilon/4$.

- Show that $B_S \subseteq B^*$.

$\exists x \in B_S \quad x \notin B^*$



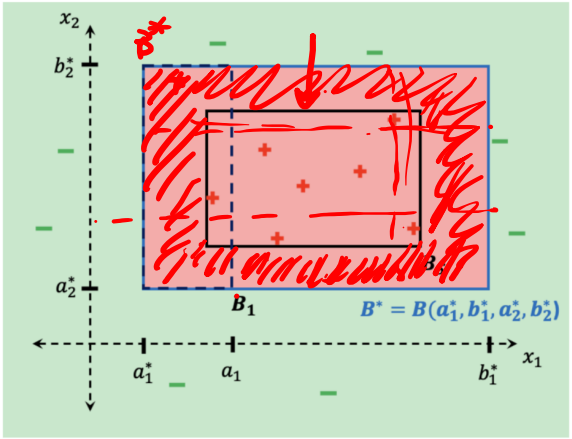
Problem 2

2.3 (8pts) Though there is non-zero probability seeing bad training set, we will now show that *with high probability* over the training dataset S, f_S^{ERM} *does* get small error. Show that if $n \geq \frac{4\log(4/\delta)}{\epsilon}$ then with probability at least $1-\delta$, $R(f_S^{ERM}) \leq \epsilon$.

To prove this follow the following steps. Let $a_1 \geq a_1^*$ be such that the probability mass (with respect to D) of the rectangle $B_1 = B(a_1^*, a_1, a_2^*, b_2^*)$ is exactly $\epsilon/4$. Similarly, let $b_1 \leq b_1^*, a_2 \geq a_2^*, b_2 \leq b_2^*$ be numbers such that the probability mass (with respect to D) of the rectangles $B_2 = B(b_1, b_1^*, a_2^*, b_2^*), B_3 = B(a_1^*, b_1^*, a_2^*, a_2), B_4 = B(a_1^*, b_1^*, b_2, b_2^*)$ are all exactly $\epsilon/4$.

- Show that if S contains (positive) examples in all of the rectangles B_1, B_2, B_3, B_4 then $R(f_S^{ERM}) \leq \epsilon$. (Hint: think about the geometric relationship between B_i and B_S, $i \in \{1,2,3,4\}$)

$$R(f_S^{ERM}) = P_x(B_S(x) \neq B^*(x)) = P_x(x \in B^* \setminus B_S) \leq \sum_{i=1}^{4} P_x(x \in B_i) = \frac{\epsilon}{4} \cdot 4 = \epsilon.$$



Problem 2

2.3 (8pts) Though there is non-zero probability seeing bad training set, we will now show that *with high probability* over the training dataset S, f_S^{ERM} *does* get small error. Show that if $n \geq \frac{4\log(4/\delta)}{\epsilon}$ then with probability at least $1-\delta$, $R(f_S^{ERM}) \leq \epsilon$.

To prove this follow the following steps. Let $a_1 \geq a_1^*$ be such that the probability mass (with respect to D) of the rectangle $B_1 = B(a_1^*, a_1, a_2^*, b_2^*)$ is exactly $\epsilon/4$. Similarly, let $b_1 \leq b_1^*, a_2 \geq a_2^*, b_2 \leq b_2^*$ be numbers such that the probability mass (with respect to D) of the rectangles $B_2 = B(b_1, b_1^*, a_2^*, b_2^*), B_3 = B(a_1^*, b_1^*, a_2^*, a_2), B_4 = B(a_1^*, b_1^*, b_2, b_2^*)$ are all exactly $\epsilon/4$.

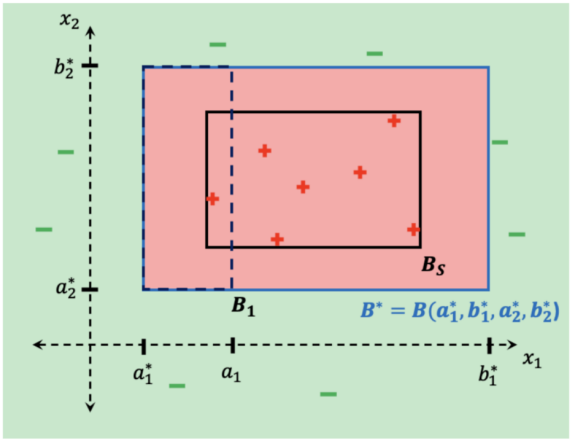
- For each $i \in \{1, \ldots, 4\}$ upper bound the probability that S does not contain an example from B_i.

$P_r(x \in B_1) = 1 - \epsilon/4$

$Pr(\text{not points in } S \in B_1)$

$\leq \prod_{i=1}^{n} Pr(x_i \notin B_1) = (1-\epsilon/4)^n \leq e^{-\frac{n\epsilon}{4}}$

$(1 - x \leq e^{-x})$



Problem 2

2.3 (8pts) Though there is non-zero probability seeing bad training set, we will now show that *with high probability* over the training dataset S, f_S^{ERM} *does* get small error. Show that if $n \geq \frac{4\log(4/\delta)}{\epsilon}$ then with probability at least $1-\delta$, $R(f_S^{ERM}) \leq \epsilon$.

To prove this follow the following steps. Let $a_1 \geq a_1^*$ be such that the probability mass (with respect to D) of the rectangle $B_1 = B(a_1^*, a_1, a_2^*, b_2^*)$ is exactly $\epsilon/4$. Similarly, let $b_1 \leq b_1^*, a_2 \geq a_2^*, b_2 \leq b_2^*$ be numbers such that the probability mass (with respect to D) of the rectangles $B_2 = B(b_1, b_1^*, a_2^*, b_2^*), B_3 = B(a_1^*, b_1^*, a_2^*, a_2), B_4 = B(a_1^*, b_1^*, b_2, b_2^*)$ are all exactly $\epsilon/4$.

- Use the union bound to conclude the argument.

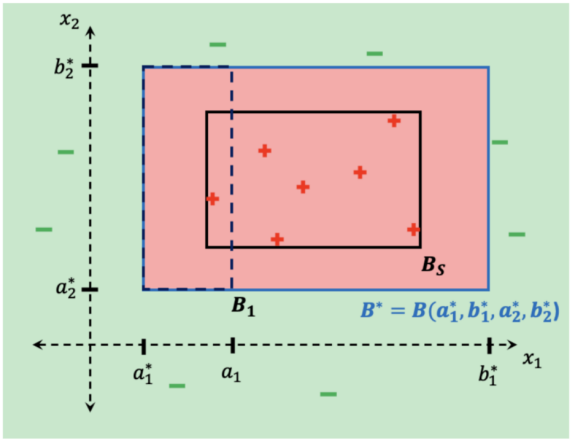
$P(R(f_S^{ERM}) > \epsilon)$

$= P(f_S^{ERM} \text{ tricked by } S)$

$\leq \sum_{i=1}^{4} P(\text{all points in } S \text{ not in } B_i)$

$\leq 4(1-\epsilon/4)^n \leq 4e^{-\epsilon n/4} \leq \delta$

$-\frac{\epsilon n}{4} \leq \log\frac{\delta}{4} \Rightarrow n \geq \frac{4}{\epsilon}\log\frac{4}{\delta}$



Problem 2

(Bonus) 2.4 (8pts) Repeat the previous question for the function class of axis-aligned rectangles in $\mathbb{R}^d$. Specifically, the current function class is $\mathcal{F}_{\text{d-rec}} = \{f_{(a_1,b_1,a_2,b_2,\ldots,a_d,b_d)}(x_1, x_2, \ldots, x_d) : a_i \leq b_i, i \in \{1, 2, \ldots, d\}\}$ and a datapoint $\mathbf{x} \in \mathbb{R}^d$ is predicted to be 1 by $f_{(a_1,b_1,a_2,b_2,\ldots,a_d,b_d)}(x_1, x_2, \ldots, x_d)$ if $a_i \leq x_i \leq b_i$ for all $i \in \{1, 2, \ldots, d\}$ and -1 otherwise. We will still assume realizability. To show this, try to generalize all the steps in the above question to higher dimensions, and find the number of training samples n required to guarantee that $R(f_S^{ERM}) \leq \epsilon$ with probability at least $1 - \delta$.

a) $B_S \subseteq B^*$. $x \in B_S$ $x \notin B^*$

$x_i < a_i^*$ or $x_i > b_i^*$

$\exists$ training data x' $x_i' < a_i^*$ +

b) $B_1 \cdots B_{2d}$

$B_1 = (a_1^*, a_1, a_2^* b_2^* \cdots a_d^* b_d^*)$

$B_2 = (b_1, b_1^*, \cdots\cdots\cdots)$

$B_3 \cdots B_{2d}$

$\varepsilon/2d$

$R(f_S^{ERM}) = P(x \in B^* \setminus B_S) \leq \sum_{i=1}^{2d} P(x \in B_i) = \varepsilon.$

(c) prob no points in B_i

$\prod_{i=1}^{n} (1 - \varepsilon/2d)^n \leq e^{-\frac{n\varepsilon}{2d}}$

(d) $P(f_S^{ERM} \geq \varepsilon)$

$\leq \sum_{i=1}^{2d} P(\text{no points in } S \in B_i)$

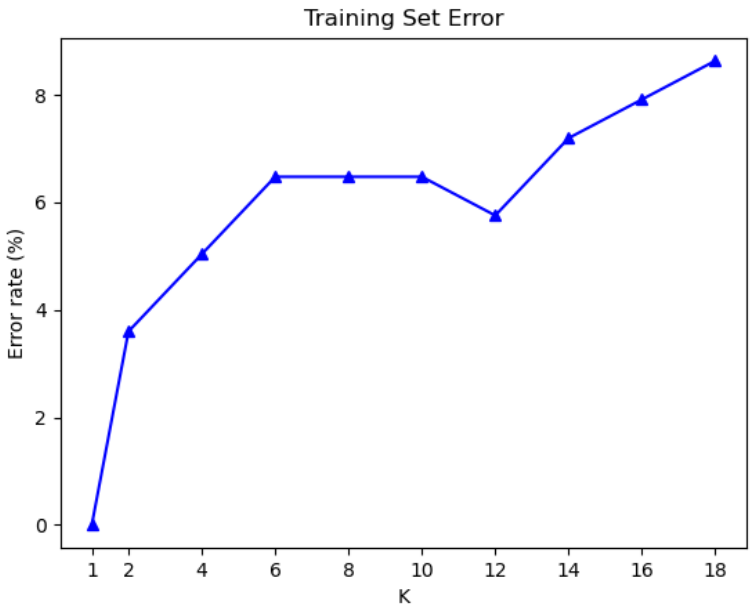
$\leq 2d e^{-\frac{n\varepsilon}{2d}} \leq \delta$

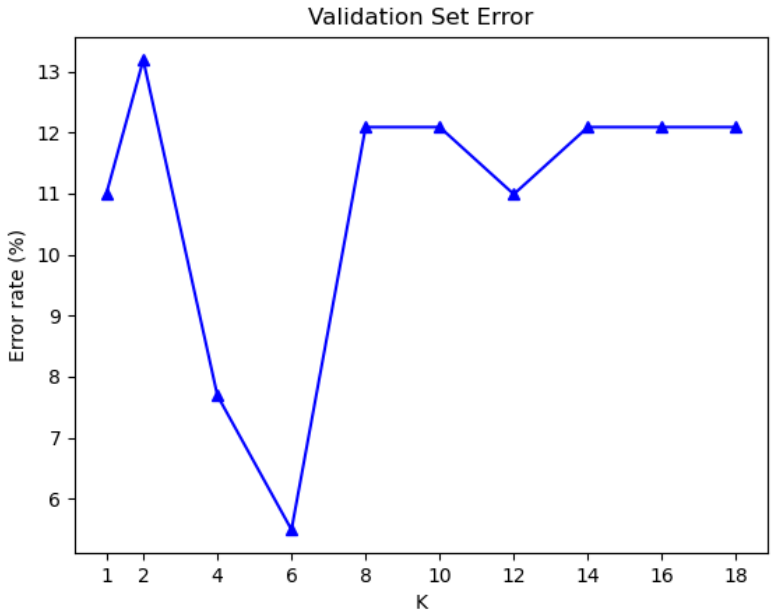
$e^{\frac{n\varepsilon}{2d}} \geq \frac{2d}{\delta}$

$\Rightarrow n \geq 2d/\varepsilon \cdot \ln \frac{2d}{\delta}$

Problem 3: KNN

3.4





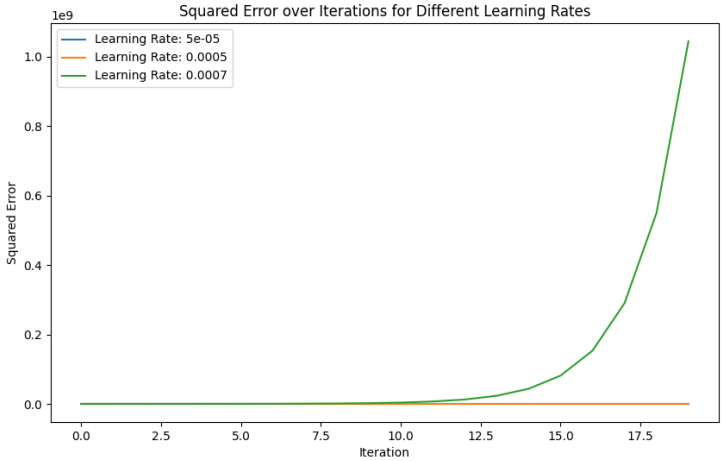
Problem 4: Gradient Descent

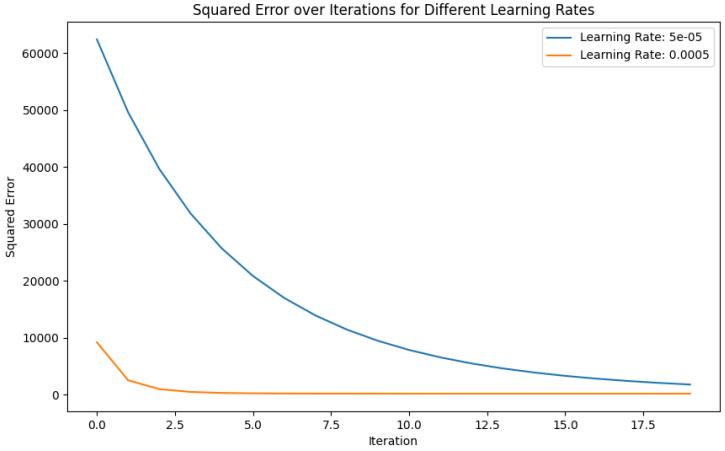
4.1

- F(w_LS) = 217.49 on training data
- F(w_0) = 78885.83 on training data
- F(w_LS) = 294.07 on testing data

Problem 4

4.2


Squared Error over Iterations for Different Learning Rates
1e9
Learning Rate: 5e-05
Learning Rate: 0.0005
Learning Rate: 0.0007
Squared Error
Iteration


Squared Error over Iterations for Different Learning Rates
Learning Rate: 5e-05
Learning Rate: 0.0005
Squared Error
Iteration

Problem 4

4.3

- SGD training loss: 439.93 (vs GD training loss 217.49)