Optimisation Project1:
Ridge Regression with Linear Model and Perceptrons

Zhe HUANG

January 2024

1 Question 1: Boston Housing Dataset

We load the Boston Housing Dataset. This dataset comprises a total of 506 samples, where each sample represents a different neighborhood in Boston. For each neighborhood, there are 13 features or attributes that provide various pieces of information relevant to housing. These features include:

- **crim**: Crime rate in the neighborhood.
- **zn**: Proportion of residential land zoned for large lots.
- **indus**: Proportion of non-retail business acres per town.
- **chas**: Presence of the Charles River (binary: 1 if the neighborhood borders the river, 0 otherwise).
- **nox**: Nitrogen oxide concentration (parts per 10 million).
- **rm**: Average number of rooms per dwelling.
- **age**: Proportion of owner-occupied units built before 1940.
- **dis**: Weighted distance to employment centers.
- **rad**: Accessibility to radial highways.
- **tax**: Property tax rate.
- **ptratio**: Pupil-teacher ratio in public schools.
- **b**: Proportion of residents of African American descent.
- **lstat**: Percentage of lower status population.

To better understand the dataset, we computed the rank of the feature matrix X using `np.linalg.matrix_rank`. The result indicates that the rank of X is 13, which means that there are no linearly dependent features, and as a result, the dimension of the kernel of X is 0.

Next, we divided the dataset into training and test sets. The training set consists of 406 samples, while the test set contains 100 samples. Consequently, for our subsequent questions, when referring to the training set, we will have $N = 406$ and $d = 13$.

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat
crim	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.379670	0.625505	0.582764	0.289946	-0.385064	0.455621
zn	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995
indus	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800
chas	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929
nox	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879
rm	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808
age	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339
dis	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996
rad	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676
tax	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993
ptratio	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044
b	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087
lstat	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000

Figure 1: The correlation matrix of X

To gain further insights into the dataset, we generated the correlation matrix Figure1, which provides information about the relationships between different features. Looking at the correlation matrix, we can observe significant correlations among certain variables, for example:

- **nox** has a strong positive correlation with **indus** (0.763651), suggesting that as the industrial concentration increases, the nitric oxides pollution might also increase.
- **dis** has a strong negative correlation with **indus** (-0.708027), indicating that locations with higher industrial concentration might be closer to employment centers.
- **age** has a high positive correlation with **nox** (0.731470), which could mean that older buildings are more likely to be found in areas with higher nitric oxides pollution.

2 Question2

Here we implement a linear regression model with the loss function calculated as

$$E(w) := \|Xw - y\|_2^2 = \sum_{i=1}^{N} \left(\langle w, x_i \rangle - y_i\right)^2 .$$

In this formulation:

- $X = (x_1, \ldots, x_N)^T \in R^{N \times d}$ is the feature matrix, with $x_i \in R^d$.
- $y \in R^N$ is the vector of labels.
- $w \in R^d$ is the vector of weights.

We then implement the gradient descent as:

$$w_{k+1} = w_k - \tau \nabla E(w_k).$$

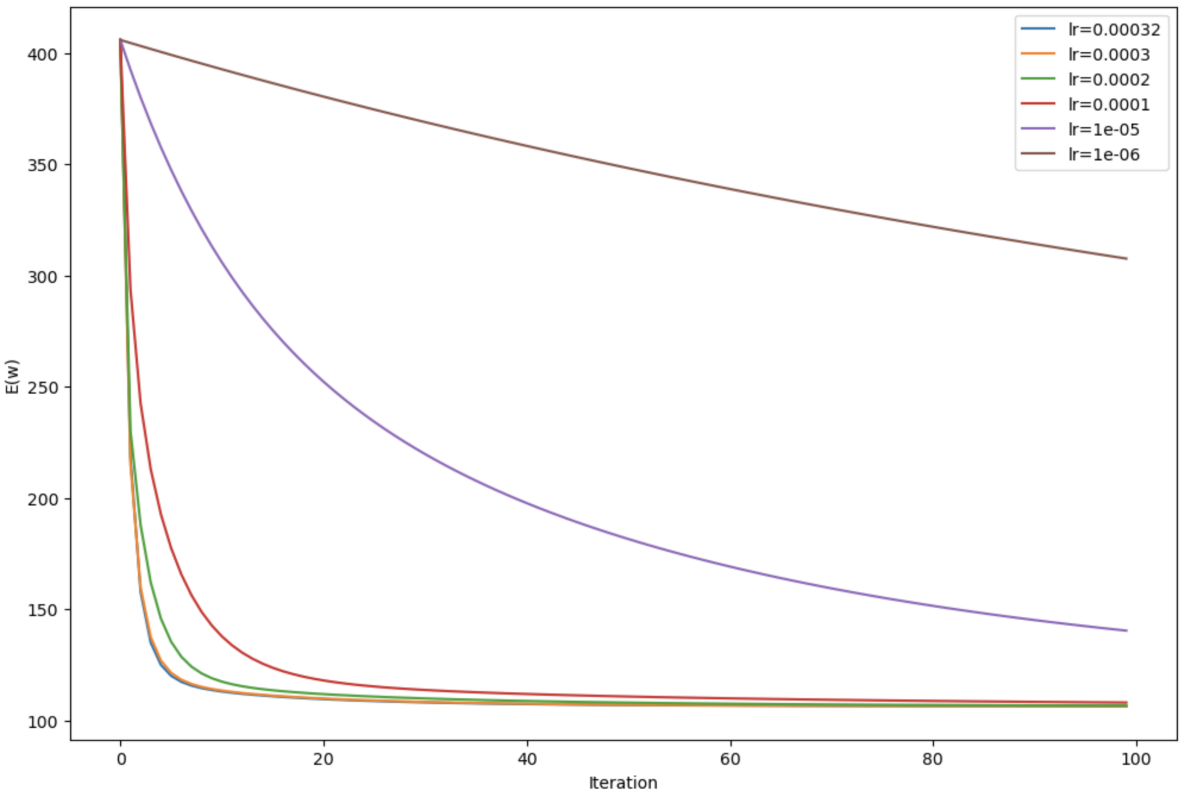

Figure 2: Convergence of $E(w)$ for several fixed step sizes

3 Question3

According to the experiment results above, the optimal step τ is 0.00032. This value allows for the fastest convergence of $E(w)$ without overshooting or causing instability in the learning process.

As we know, the theoretical upper bound of the step size τ_{upper} can be calculated as $\tau_{upper} = \frac{1}{\lambda_{max}(X^T X)}$, which is 0.00044 in this case. The experimentally optimal step size $\tau = 0.00032$ is smaller than the upper bound τ_{upper}, which is reasonable.

4 Question4

Here we implement a Multilayer Perceptron (MLP) with two layers: a single hidden layer consisting of q neurons and an output layer. In this implementation, we do not apply any activation function.

The output of the MLP, denoted as $g(\theta, x)$, is calculated as

$$g(\theta, x) := \sum_{k=1}^{q} w_k \langle v_k, x \rangle$$

where:

- $\theta = (V, w)$ are the parameters of the MLP, i.e. the inner weights V and the outer weights w.

- q is the number of neurons
- w_k represents the k-th outer weight associated with the k-th neuron in the hidden layer.
- v_k is the k-th column of the inner weight matrix V, corresponding to the k-th neuron's weights.

The original loss function we want to minimize is

$$F(\theta) := \sum_i |g(\theta, x_i) - y_i|^2.$$

5 Question5

In order to minimize the $F(\theta) := \sum_i |g(\theta, x_i) - y_i|^2.$, we can start by fixing $V = V_0$.

In this case, we will have:

$$\begin{aligned} E(w) :=& F(V_0, w) \\ =& \sum_i |g(V_0, w, x_i) - y_i|^2 \\ =& \sum_i |\sum_{k=1}^{q} w_k \langle v_k^0, x_i \rangle - y_i|^2 \\ =& \sum_i |\langle w, x_i' \rangle - y_i|^2 \\ =& \|X'w - y\|_2^2 \end{aligned}$$

where

- $X' = (x_1', x_2', ..., x_N')^T \in R^{N \times q}$ is the transformed feature matrix, with each $x_i' = (\langle v_1^0, x_i \rangle, \langle v_2^0, x_i \rangle, ..., \langle v_q^0, x_i \rangle) \in R^q$
- Easily, $X' = XV_0^T$

Thus, the function $E(w) := F(V_0, w)$ becomes a regression problem of the same form as Question2 but with a different matrix X'.

But it's important to note that there is not always a unique solution to this problem. We already know that $X \in R^{N \times d}$ has full column rank and V_0^T is a $d \times q$ random matrix.

- When $q > N$, $X' = XV_0^T \in R^{N \times q}$ cannot have full column rank because the number of columns is larger than the number of rows. In this case, the problem may have an infinite number of solutions.
- When $q \leq N$ but $q > d$, V_0^T cannot have full column rank because the number of columns is larger than the number of rows. Thus, $X' = XV_0^T$ cannot have full column rank. In this case, the problem may have an infinite number of solutions as well.
- When $q \leq N$ and $q \leq d$, V_0^T likely has full column rank. Thus, $X' = XV_0^T$ likely has full column rank. In this case, the problem may have a unique solution.(I use 'likely' because there is still a small chance that V_0^T does not have full column rank if the columns of V_0^T are linearly dependent.)

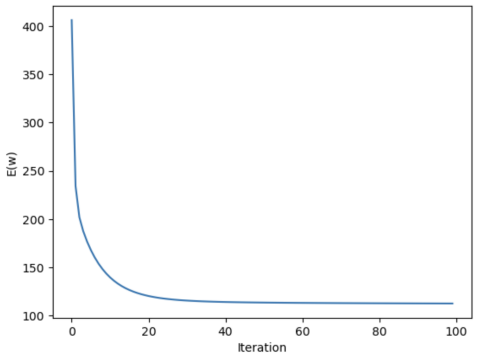

Figure 3: Convergence of E(w): q = 10

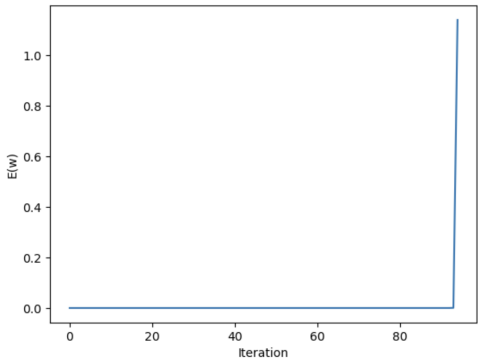

Figure 4: Convergence of E(w): q = 1000

What does the theory tell us about the convergence? We only consider the case that $q \leq N$ and $q \leq d$ here because it is the only case that the problem may have a unique solution.

Similar to Question 3, the theoretical upper bound of the step size τ'_{upper} can be calculated as $\tau'_{upper} = \frac{1}{\lambda_{max}(X'^T X')}$, which is 0.000035 in this case. The convergence of the algorithm is then guaranteed if $\tau < \tau'_{upper}$.

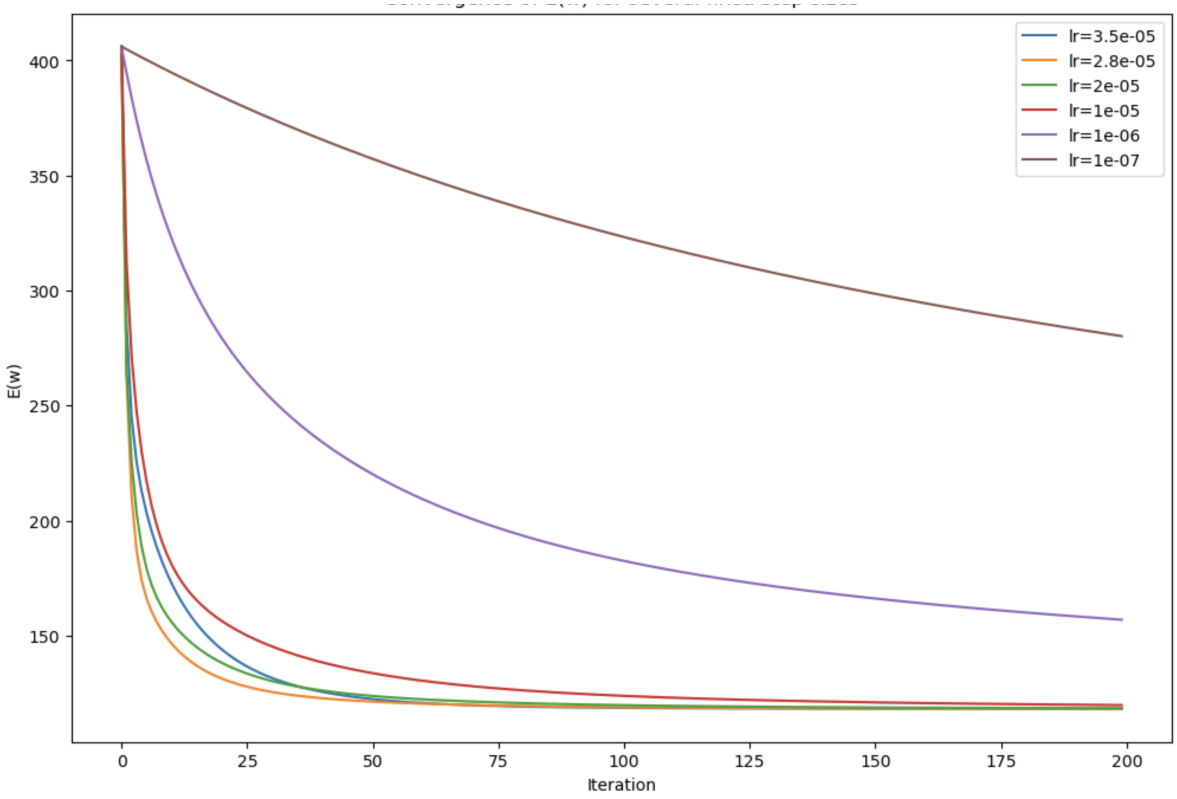

Figure 5: Convergence of $E(w) := F(V_0, w)$ for several fixed step sizes

According to the experiment results above, the optimal step size τ is 0.000028. This value allows for the fastest convergence of $E(w)$ without overshooting or causing instability in the learning process.

6 Question6

We now take the optimal w^* optimized from Question 5. By fixing $w = w^*$ this time, we can only optimize the inner weights V.

Firstly, we have

$$\begin{aligned} G(V) :=& F(V, w^*) \\ =& \sum_i |g(V, w^*, x_i) - y_i|^2 \\ =& \sum_i |\sum_{k=1}^{q} w_k^* \langle v_k, x_i \rangle - y_i|^2 \\ =& \sum_i |\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i|^2 \end{aligned}$$

where

- $w^* = (w_1^*, w_2^*, ..., w_q^*)^T \in R^q$ is the optimal outer weights obtained before.

Then we have

$$\frac{\partial G(V)}{\partial V} = \begin{pmatrix} \frac{\partial G(V)}{\partial v_1}^T \\ \\ \frac{\partial G(V)}{\partial v_2}^T \\ \\ \vdots \\ \\ \frac{\partial G(V)}{\partial v_q}^T \\ \\ \end{pmatrix}$$

where the k-th partial derivative is calculated as

$$\begin{aligned} \frac{\partial G(V)}{\partial v_k} =& \frac{\partial}{\partial v_k} \sum_i \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right)^2 \\ =& \sum_i \frac{\partial}{\partial v_k} \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right)^2 \\ =& \sum_i 2 \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right) \cdot \frac{\partial}{\partial v_k} \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right) \\ =& \sum_i 2 \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right) \cdot \frac{\partial}{\partial v_k} \langle v_k, w_k^* x_i \rangle \\ =& \sum_i 2 \left(\sum_{k=1}^{q} \langle v_k, w_k^* x_i \rangle - y_i \right) \cdot w_k^* x_i \end{aligned}$$

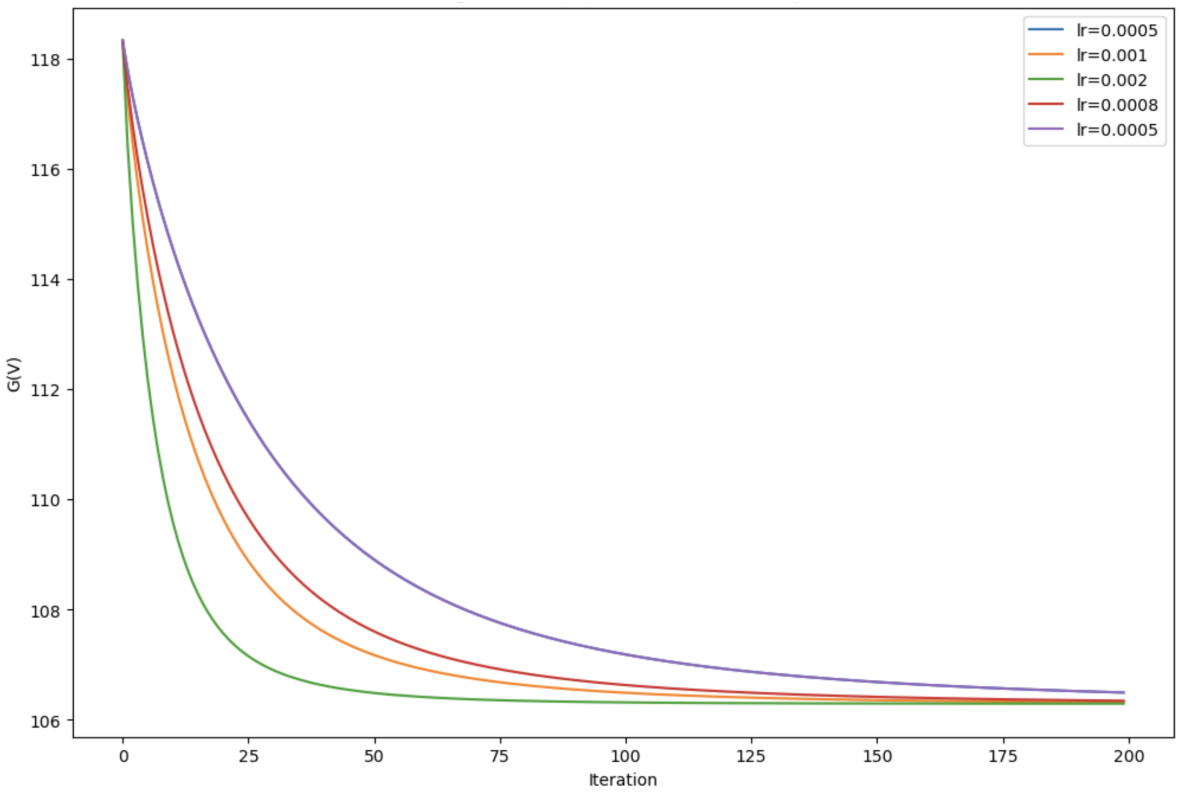

Figure 6: Convergence of $G(V) := F(V, w^*)$ for several fixed step sizes

7 Bonus Question

Using the test set we obtained before, we compute the test error using the linear model and the MLP.

The test error of the linear model is 32.721 while the test error of the MLP is 38.608.