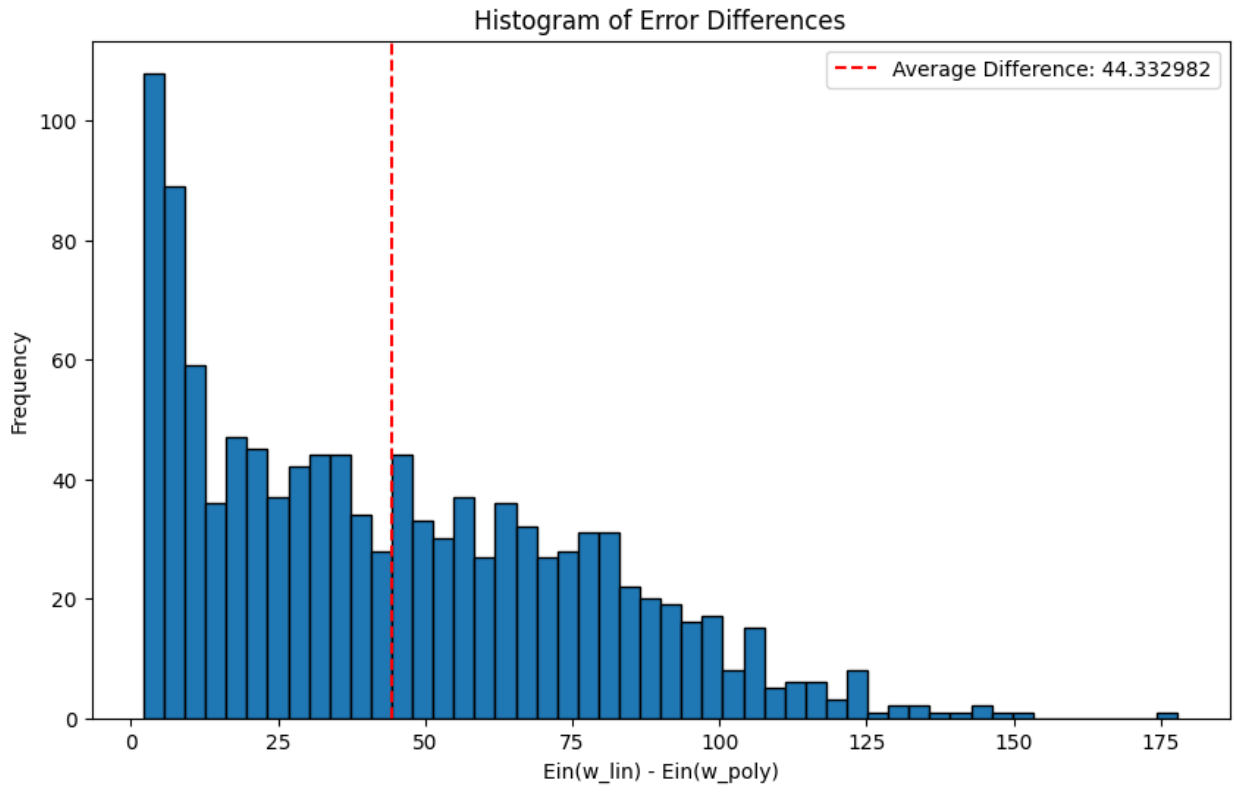
R13922136 周洛君

ML homework 4: question 11



Meaning:

From this plot we can see that the differences are mostly small positive values, which means that the error we got using linear regression is slightly greater than the error we got using polynomial transform.

This is because by using polynomial transform, we can use more parameters to fit our data, making our model more flexible to capture the nonlinear relationships among the features, thus, we can draw a function that is nearer to fitting all the examples over the 64 points.

However, the difference between the two approaches are not large, owing to our small dataset and small number of features, since under this situation, it is possible to find a line that nearly fits the points in a low dimensional space.

R13922136 周洛君

Code:

transform function $\Phi()$

We use this function to map the orignal data into a higher dimensional space, and save the augmented input vectors in `X_aug_arr`. Which means that each of the element in `X_aug_arr` is a 37-dimensional vector (12 features + 1 constant + 12 squared features + 12 cubed features).

```python
def Phi(X_arr):
    X_aug_arr = []
    for arr in X_arr:
        Phi_arr = np.concatenate((np.array([1]), arr, arr**2, arr**3))
        X_aug_arr.append(Phi_arr)
    return np.array(X_aug_arr)

```

[64] ✓ 0.0s

get weight vector

Use the pseudo inverse to get the weight vector.

```python
def weight_vector(X_in_sample_mat_lin, y_in_sample_array, X_in_sample_mat_poly):

    w_lin = np.linalg.pinv(X_in_sample_mat_lin) @ y_in_sample_array
    w_poly = np.linalg.pinv(X_in_sample_mat_poly) @ y_in_sample_array
    return w_lin , w_poly
```

[26] ✓ 0.0s

in sample error

```python
def in_sample_error(X_sample_mat_lin, X_sample_mat_poly, y_sample_array, w_lin, w_poly):

    in_sample_error_lin = np.mean((X_sample_mat_lin @ w_lin - y_sample_array) ** 2)
    in_sample_error_poly = np.mean((X_sample_mat_poly @ w_poly - y_sample_array) ** 2)
    return in_sample_error_lin, in_sample_error_poly
```

[27] ✓ 0.0s

Main function

```python
for experiment in range(1126):

    seed = experiment
    random_sample_indices = generate_random_sample(seed)
    X_sample = [X[i] for i in random_sample_indices]
    y_sample = [y[i] for i in random_sample_indices]

    X_sample_mat_lin = np.array(convert_dtype(X_sample))
    X_sample_mat_poly = Phi(convert_dtype(X_sample))

    y_sample_array = np.array(y_sample)

    w_lin, w_poly = weight_vector(X_sample_mat_lin, y_sample_array, X_sample_mat_poly)
    in_sample_error_lin, in_sample_error_poly = in_sample_error(X_sample_mat_lin, X_sample_mat_poly, y_sample_array, w_lin, w_poly)

    lin_sub_poly_error.append(in_sample_error_lin - in_sample_error_poly)
    avg_difference = np.mean(lin_sub_poly_error)



```

[28] ✓ 0.7s

```python
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.hist(lin_sub_poly_error, bins=50, edgecolor='black')
plt.axvline(x=avg_difference, color='r', linestyle='--',
            label=f'Average Difference: {avg_difference:.6f}')
plt.xlabel('Ein(w_lin) - Ein(w_poly)')
plt.ylabel('Frequency')
plt.title('Histogram of Error Differences')
plt.legend()
plt.show()
```

[29] ✓ 0.1s