Least Squares vs. Least Mean Squares

Tre' R. Jeter

September 14, 2023

Abstract

Linear regression is a fundamental technique in machine learning and statistics for modeling the relationship between input features and a target variable. In this study, we explore two prominent methods for creating linear regression models: the Least Squares (LS) method and the Least Mean Squares (LMS) method.The Boston Housing dataset, a classic benchmark dataset in regression analysis, serves as the basis for our investigation. We begin by applying the LS method, a closed-form analytical solution, provides an accurate and interpretable linear model for predicting housing prices based on various input features. Next, we apply the LMS method, a powerful iterative approach that employs gradient descent to adaptively update model weights. We investigate multiple learning rates within the LMS method to observe their impact on convergence speed and model performance.

1 Setup

The Boston Housing dataset set has 14 attributes (features) and I regress to PRICE, the 14th feature. Because PRICE is set as my target variable, I do not initially use it as an input. I will show the difference of using it as an input later in the results. Because the dataset is small, I partition the data to provide 2/3 as a training set and the remaining 1/3 as a test set. I have also scaled the data using the *StandardScaler* library.

2 Least Squares (LS) Method

With the LS method, we are instructed to present the error and variance in table form. We are also instructed to show the effects of different levels of regularization. Figure 1 is a table that shows the mean squared error (MSE), r-2 score, and variance for the LS method under Ridge, Lasso, and ElasticNet regularization. Notice that in the first column, the regularization level is set to 0 meaning that no regularization was actually used. This is to show different perspectives of *using* and *not using* regularization in the LS method. Figure 3a shows the relationship between MSE and regularization strength.

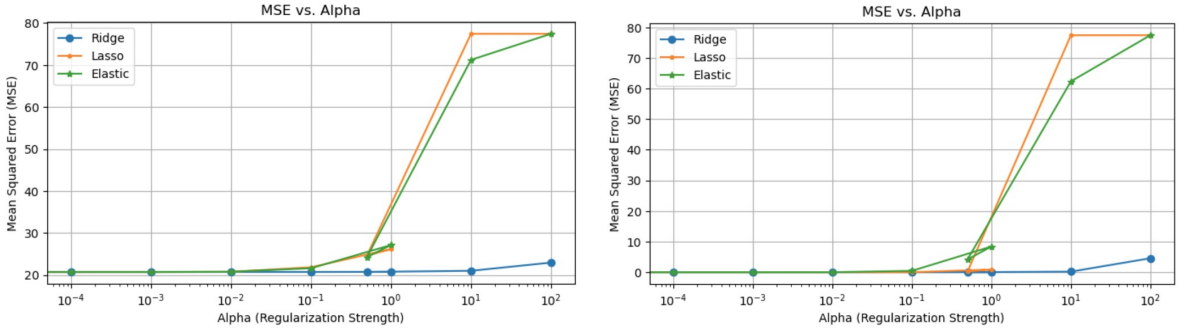
In the case of using the target value PRICE as an input, the results *curve-wise* don't change much visually. However, Figure 2 shows that the MSE values are much lower and the r-2 scores are much higher when including the target variable. This is because there is a higher correlation between the features now that the target is included. The trends in Figure 3b are very similar to those of Figure 3a.

	Alpha (Regularization Strength)	MSE (Ridge)	R-2 (Ridge)	Variance (Ridge)	MSE (Lasso)	R-2 (Lasso)	Variance (Lasso)	MSE (Elastic)	R-2 (Elastic)	Variance (Elastic)
0	0.0000	20.724023	0.726157	0.000000e+00	20.724023	0.726157	0.000000e+00	20.724023	0.726157	0.000000e+00
1	0.0001	20.724026	0.726157	2.000001e-12	20.724309	0.726153	2.040177e-08	20.724645	0.726149	9.665430e-08
2	0.0010	20.724052	0.726157	1.617794e-10	20.726890	0.726119	1.662563e-06	20.730277	0.726074	7.911047e-06
3	0.0100	20.724306	0.726153	1.404298e-08	20.761331	0.725664	2.477293e-04	20.791357	0.725267	7.991488e-04
4	0.1000	20.726853	0.726120	1.222427e-06	21.853021	0.711239	2.005017e-01	21.614785	0.714387	1.223592e-01
5	1.0000	20.752416	0.725782	1.080807e-04	26.166377	0.654243	3.934874e+00	27.140175	0.641375	5.480812e+00
6	0.5000	20.738197	0.725970	1.023783e-04	24.632242	0.674515	4.337039e+00	24.280321	0.679165	5.360385e+00
7	10.0000	21.001466	0.722491	8.117032e-03	77.439660	-0.023271	3.372192e+02	71.234374	0.058724	2.667412e+02
8	100.0000	22.954033	0.696690	4.807376e-01	77.439660	-0.023271	5.302659e+02	77.439660	-0.023271	4.745747e+02

Figure 1: Least Squares method with three different regularization techniques and multiple regularization strengths per technique. Included are MSE, r-2, and variance scores.

	Alpha (Regularization Strength)	MSE (Ridge)	R-2 (Ridge)	Variance (Ridge)	MSE (Lasso)	R-2 (Lasso)	Variance (Lasso)	MSE (Elastic)	R-2 (Elastic)	Variance (Elastic)
0	0.0000	1.523494e-28	1.000000	0.000000e+00	4.797252e-06	1.000000	0.000000e+00	0.000005	1.000000	0.000000e+00
1	0.0001	2.651995e-11	1.000000	1.758270e-22	2.342190e-06	1.000000	1.506832e-12	0.000005	1.000000	1.974541e-14
2	0.0010	2.651914e-09	1.000000	1.547338e-18	8.855194e-07	1.000000	2.605652e-12	0.000079	0.999999	1.207792e-09
3	0.0100	2.651098e-07	1.000000	1.309065e-14	8.785548e-05	0.999999	1.362401e-09	0.006739	0.999911	8.441454e-06
4	0.1000	2.642966e-05	1.000000	1.112093e-10	8.785548e-03	0.999884	1.228353e-05	0.458045	0.993947	3.332583e-02
5	1.0000	2.564055e-03	0.999966	9.094014e-07	8.785548e-01	0.988391	1.067798e-01	8.411434	0.888853	9.638433e+00
6	0.5000	6.518404e-04	0.999991	7.854161e-07	2.196387e-01	0.997098	9.215558e-02	4.177931	0.944794	9.153206e+00
7	10.0000	1.957095e-01	0.997414	4.170182e-03	7.743966e+01	-0.023271	6.533153e+02	62.370075	0.175855	4.084175e+02
8	100.0000	4.569765e+00	0.939616	2.043813e+00	7.743966e+01	-0.023271	1.032344e+03	77.439660	-0.023271	8.198853e+02

Figure 2: Least Squares method when including the target variable as an input.



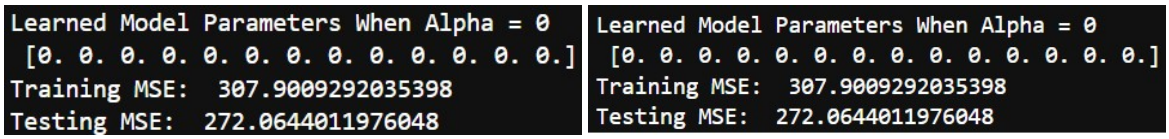
(a) Without using the target as an input.

(b) Using the target as an input.

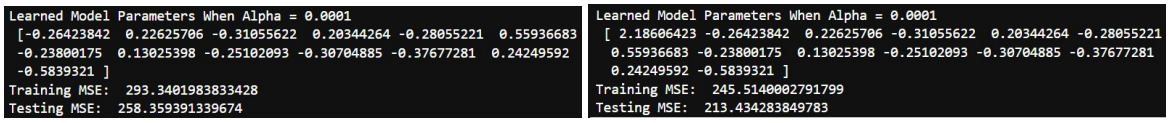
Figure 3: Whether or not the target is used as an input seems to have no major effect on the trends between MSE and regularization strength.

3 Least Mean Squares (LMS) Method

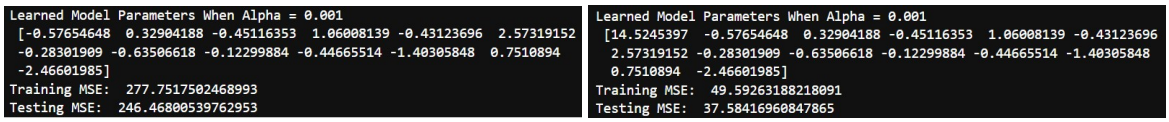
With the LMS method, we are instructed to experiement with multiple learning rates to show the effects through visualizing learning curves. Figure 4 shows the learned model parameters at a certain learning rate when a bias term is and is not present. The MSE of training and test sets are also presented. Figure 5 shows comparison learning curves from using learning rates of [0, 0.0001, 0.001, 0.01, 0.1] over 1000 iterations with and without a bias term. The learning curves corroborate the metrics found in Figure 4. For clarity, the x-axis is *Iteration* and the y-axis is *MSE*. As we can see, increasing the learning rate and the inclusion of a bias term plays a crucial role in the convergence of the algorithm. At smaller learning rates like 0 and 0.0001, the algorithm will either not converge or will take too long to converge (respective to 0 and 0.0001). At a larger learning rate like 0.1, we see the learning curve drastically converges, but it's possible to have missed some model parameters. In regards to which learning rate allowed the LMS method to perform better, I would choose 0.01 because as it converges quickly, it's still gradual and not abrupt. Figure 7 shows how the weights update over 1000 iterations for each respective learning rate with and without a bias term. Again, for clarity, the x-axis is *Iteration* and the y-axis is *Weight Value*. With learning rates of 0 and 0.0001, the weight updates are consistent with the learning curves seeing that the model converges very slowly or not at all. The weight updates for learning rates 0.001, 0.01, and 0.1 show the best estimation of weights indicating that they are the more optimal choices of learning rates to choose. Specifically, learning rate of 0.01 converges very fast and the weight updates are more likely to even out compared to others.



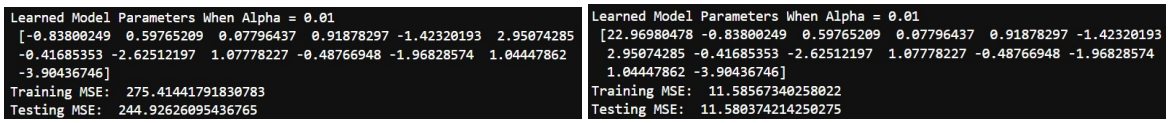
(a) Learning rate = 0. *Left:* No bias term. *Right:* With bias term.



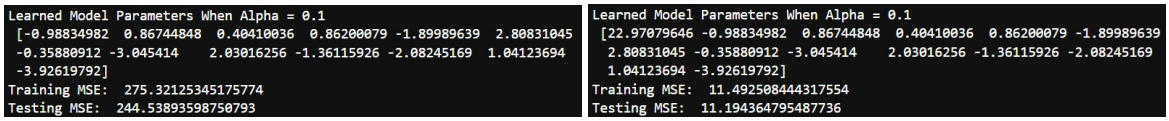
(b) Learning rate = 0.0001. *Left:* No bias term. *Right:* With bias term.



(c) Learning rate = 0.001. *Left:* No bias term. *Right:* With bias term.

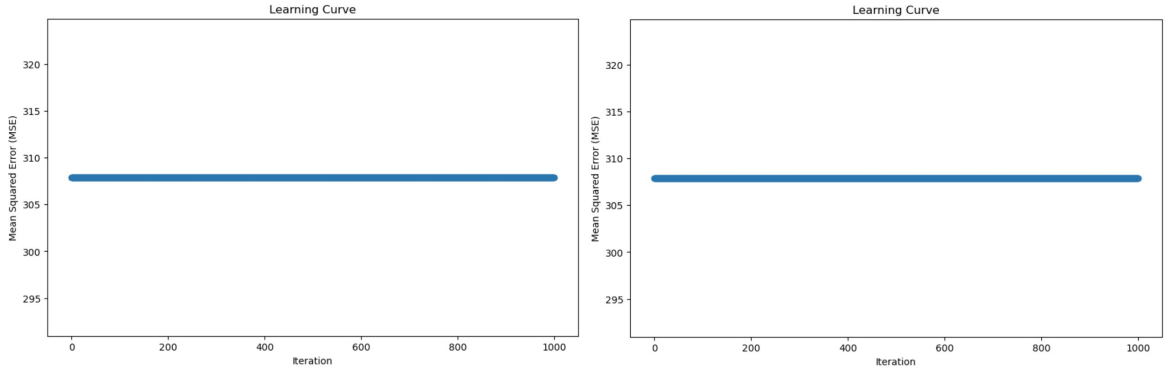


(d) Learning rate = 0.01. *Left:* No bias term. *Right:* With bias term.

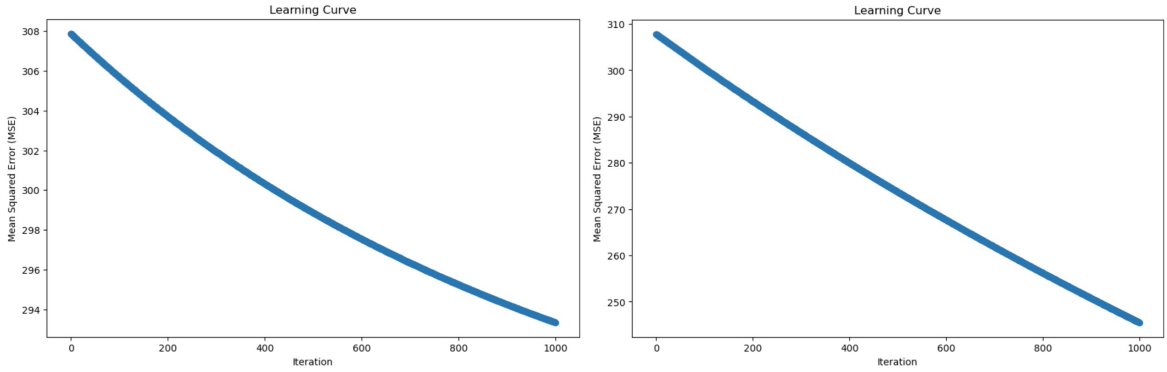


(e) Learning rate = 0.1. *Left:* No bias term. *Right:* With bias term.

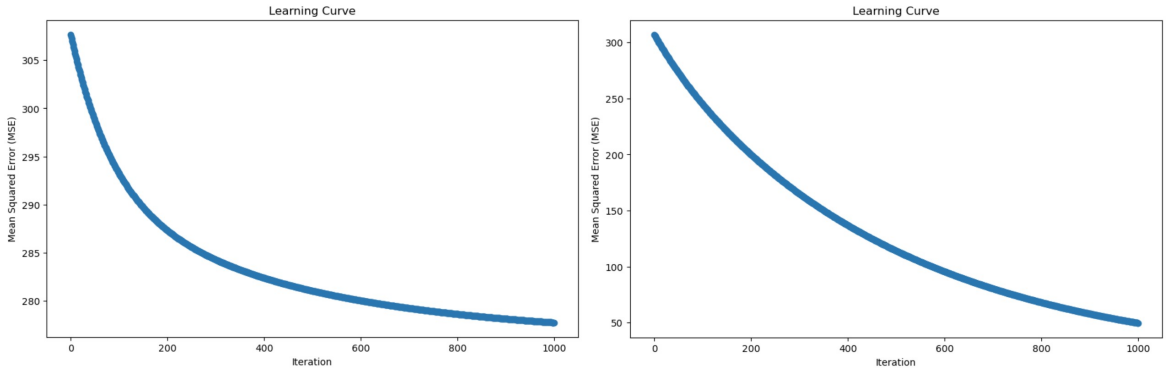
Figure 4: Learned model parameters with respect to learning rates with and without a bias term. Training and Testing MSE is also included.



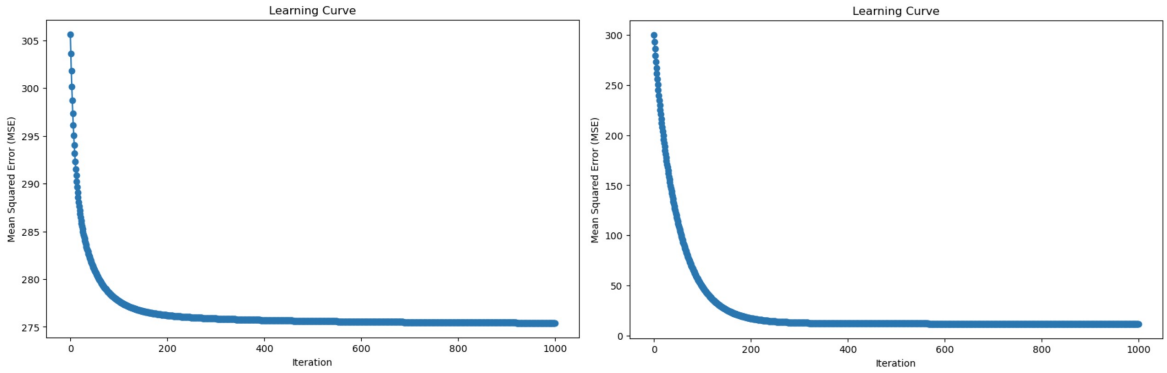
(a) Learning rate = 0. *Left:* No bias term. *Right:* With bias term.



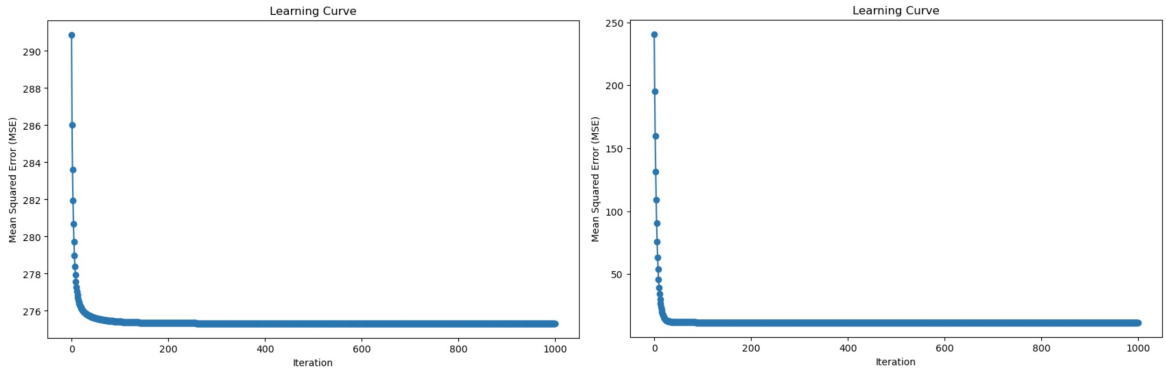
(b) Learning rate = 0.0001. *Left:* No bias term. *Right:* With bias term.



(c) Learning rate = 0.001. *Left:* No bias term. *Right:* With bias term.



(d) Learning rate = 0.01. *Left:* No bias term. *Right:* With bias term.



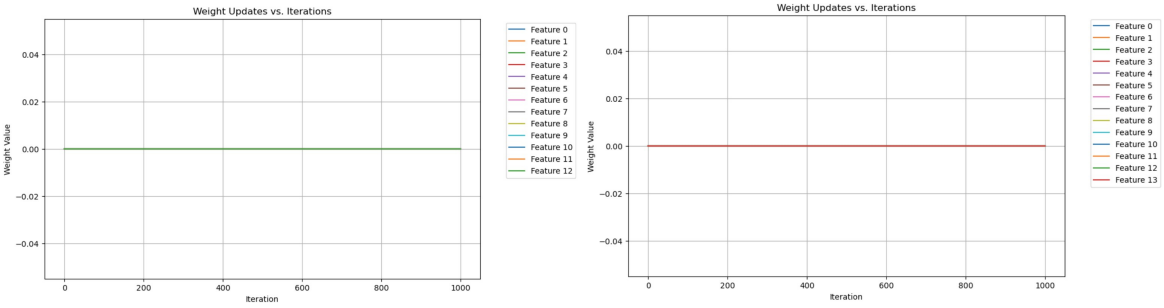
(e) Learning rate = 0.1. *Left:* No bias term. *Right:* With bias term.

Figure 5: Learning curves for the LMS method with differing learning rates with and without a bias term.

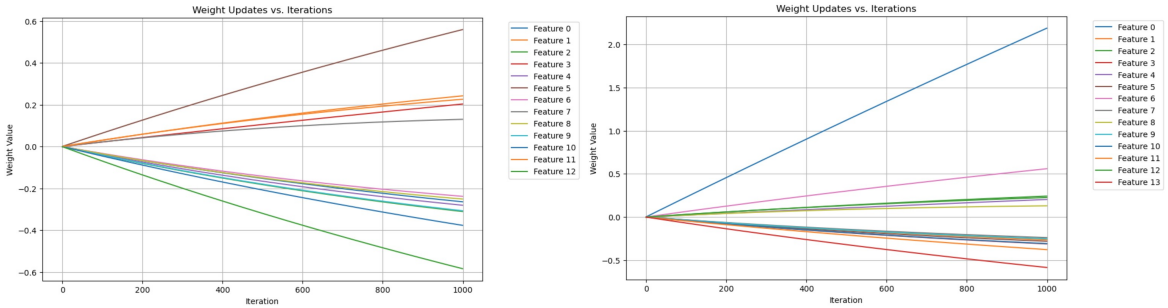
4 References

Below is a list of references used to assist with this assignment:

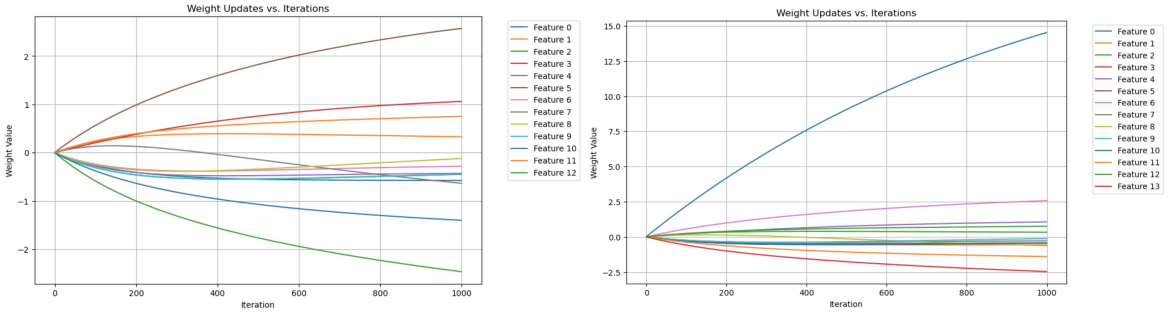
https://github.com/lakshyamahawar14/linearregressionpython/blob/main/linearregression.ipynb
https://predictivemodeler.com/2019/08/19/py-ols-boston-house-prices/
https://predictivemodeler.com/2019/08/19/py-ridge-boston-house-prices/
https://predictivemodeler.com/2019/08/19/py-lasso-boston-house-prices/
https://predictivemodeler.com/2019/08/19/py-elasticnet-boston-house-prices/
https://levelup.gitconnected.com/expensive-cheap-housing-prices-predict-prices-boston-housing-dataset-d60987b65c75
https://amitg0161.medium.com/sklearn-linear-regression-tutorial-with-boston-house-dataset-cde74afd460a
https://numpy.org/doc/stable/reference/generated/numpy.linalg.lstsq.html
https://www.pluralsight.com/guides/linear-lasso-ridge-regression-scikit-learn



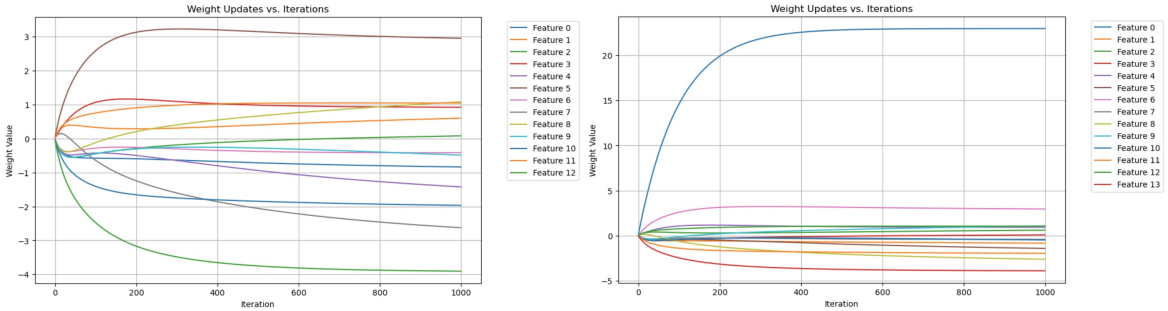
(a) Learning rate = 0. *Left:* Weights without bias term. *Right:* Weights with bias term.



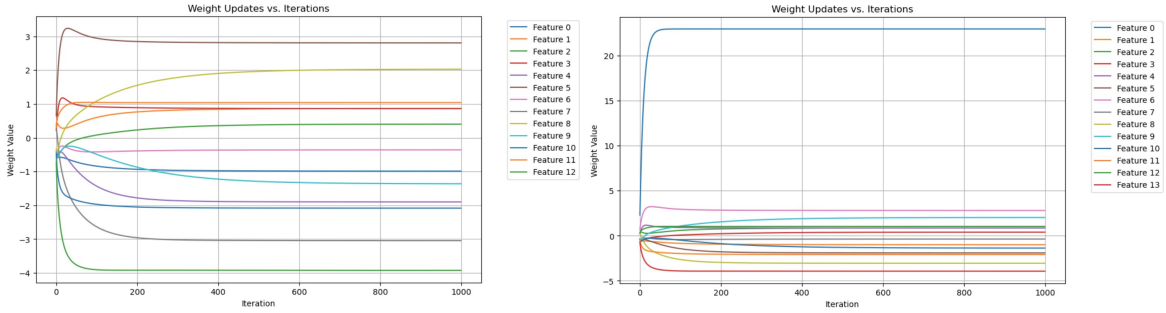
(b) Learning rate = 0.0001. *Left:* Weights without bias term. *Right:* Weights with bias term.



(c) Learning rate = 0.001. *Left:* Weights without bias term. *Right:* Weights with bias term.



(d) Learning rate = 0.01. *Left:* Weights without bias term. *Right:* Weights bias term.



(a) Learning rate = 0.1. *Left:* Weights without bias term. *Right:* Weights bias term.

Figure 7: Weight updates for the LMS method with differing learning rates with and without a bias term.