Introduction to Machine Learning
Lab Session 2

Group 15
Lubor Budaj (s4167376) & Stefan Uta (s3933563)

October 2, 2022

Introduction

In machine learning, linear regression is a type of supervised learning method. This technique determines a linear relationship between data (X) and labels (y). The simple linear regression quantifies the relationship of a dependent variable value (y) based on one given independent variable (x). As an extension, multiple linear regression takes more than one independent variable (x) as input.

We applied multiple linear regression on a 500 by 25-dim feature vector with a set of 500 target labels with random selection of training data. Firstly, we describe our methodology. Then we provide the resulting graphs capturing different sizes of training dataset. In the end we explain the results we produced.

Method

The idea behind linear regression is to minimize the difference between predicted and actual values of label vector y. We can get average difference by calculating mean squared error(MSE), which is given by:

$$E = \frac{1}{2} \sum_{\mu=1}^{P} [w \cdot x^{\mu} - y^{\mu}]^2$$

In order to minimize MSE we take a derivation of E and find the critical point. Since the error is not bounded from above, the critical value we find is a local minimum:

$$\delta_w E = \sum_{\mu=1}^{P} [w \cdot x^{\mu} - y^{\mu}] x^{\mu} = 0$$
$$X^T (Xw - Y) = 0$$
$$w^* = [X^T X]^{-1} X^T Y$$

Hence, in our algorithm we used the derived formula to calculate the vector w for each value of P. Next, MSE normalized by the number of training samples P taken randomly from the training data (E_{train}) is calculated for each value of P using the previous mentioned formula. Finally, we calculate MSE of test data (E_{test}), which is normalized in regards to the number of entries in the data set. We do it in the same way as with (E_{train}), but we use the test dataset instead of the training dataset.

Furthermore, our implementation has a support for regularization. Regularization is useful, when the training set is not big enough - the number of training samples is smaller than the number of categories of the data. In such case, regularization term λ can help provide better results for low values of P. To use regularization in our implementation, all that is needed is to set the input argument *lambda* to the desired valued of λ.

Our implementation is scalable - the only thing needed to get w for different values of P is to change the input vector containing the values of P. To avoid bias in selection of λ, we make a random selection of P samples from training data for each training. Therefore, for low numbers of P, each training session produces different results. To be able to compare the results using the same data selection from training dataset, when comparing results with different values of λ, seed value for random generator is another argument of the function.

Results

We produced a plot (Figure 1) showing normalized MSE for both E_{train} and E_{test} using values of $P \in \{30, 40, 50, 75, 100, 200, 300, 400, 500\}$. In this plot x-axis is value of P and y-axis is normalized MSE. E_{train} and E_{test} are distinguished by color (see the legend). Circles in the plot represent points of measurement.

Next we produced a set of bar graphs (Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9), one for each value of $P \in \{30, 40, 50, 75, 100, 500\}$. In these graphs, x-axis represents components of vector w for given P and y-axis is the estimated value of the components.

Lastly, we made 2 plots (Figure 2 and Figure 3), which show normalized MSE for both E_{train} and E_{test} using values of $P \in \{5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 200, 300, 400, 500\}$. The difference between these graph is the fact that the first one (Figure 2) does not make use of regularization term, while the second one (Figure 3) uses $\lambda = 0.7$. To generate these 2 plots we used the same permutation of data from training data.

Discussion

The key figure for us is Figure 1. using it we can show several features of linear regression. Firstly, we can see that the Normalized MSE for testing data (E_{test}) decreases as the size of training set P increases. This is the behaviour we expected to see. It can be explained by the fact that for bigger training sets, there is a lower bias, hence the test data fit better to resulting vector w we get. On the other hand, when the training set is small (e.g. $P = 30$), w is more biased towards the sample of the training data used, hence the normalized MSE is larger. With Normalized MSE for training data (E_{train}) it is the other way around. E_{train} increases as the size of the training set increases. This is again the expected result. It is caused by the fact that with small training set, the data can fit better the training set. As previously mentioned, it is more biased towards it. Hence, the value of E_{train} is smaller when P is small. On the other hand, as the size of the training set increases, w can fit the data from this training set less and less, hence the E_{train} increases. The last thing we can see in this plot is the fact that as the size of the training set increases, the 2 curves approach each other, but never intersect. Again, this is the expected outcome. As described earlier, with the increasing size of the training set the fit of w for it decreases, while fit of w for test data increases. However, assuming the sufficient size of both the training and the testing sets, the 2 curves should not intersect.

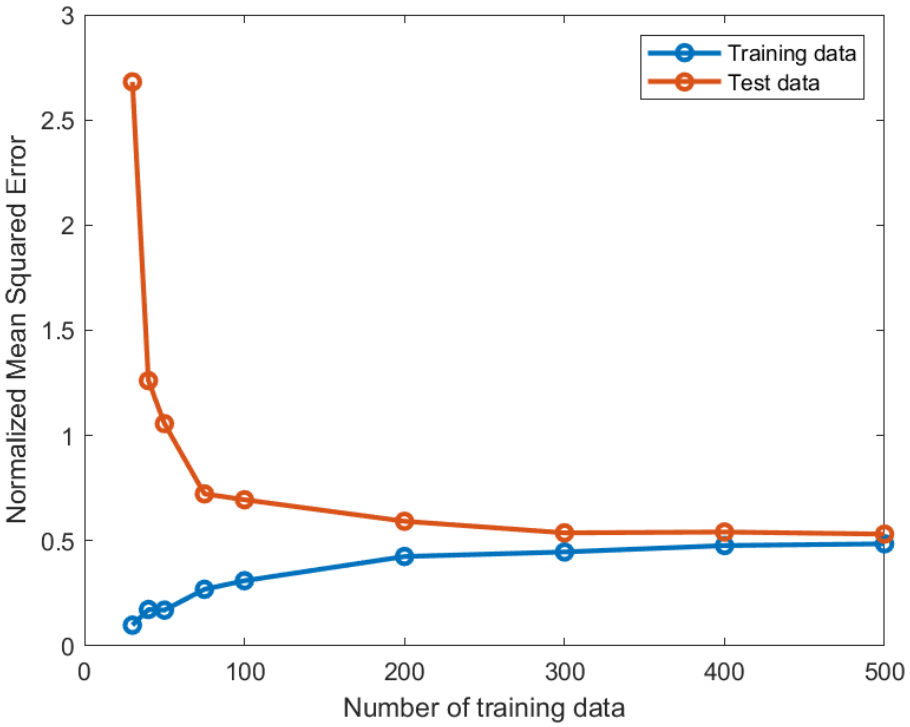

Figure 1: *Normalized MSE for E_{train} and E_{test} for values of $P \in \{30, 40, 50, 75, 100, 200, 300, 400, 500\}$*

Next we will compare the bar graphs for values of $P \in \{30, 40, 50, 75, 100, 500\}$ (Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9). When comparing these graph it is important to keep in mind different y-axis scale in each figure. In Figure 4, where $P = 30$, we can see a lot of spikes in components of vector w. We cannot deduce any kind of pattern. When we observe Figure 7, we can see that there appears to be some pattern in the values of components of w - a lot of the components have a similar value and the components 1-19 are alternating between negative and positive values. However, this is not enough for us to assume what generated the data. Lastly, we examine Figure 9. We can see that the components 1-20 of w have stabilized at roughly the same value, but are alternatingly negative or positive. Components 21-25 appear to have an approximately same value as well, but are all positive. If we compare all the figures mentioned at the beginning of this paragraph, we can see the trend of progressive stabilization of values of w as the training set gets larger. Higher the P, more pattern there seem to be. Hence, based on Figure 9, which is a highest P possible, we postulate that the true vector w can be explained with some value a, such that the even components between 1-20 and the components 21-25 came have value a, while the odd components between 1-20 have value $-a$.

Lastly, we will compare the plot that make use of the regularization term λ (Figure 3) and the one that uses the same training data, but does not make use of regularization term (Figure 2). Before we generated the graphs, we experimented with values of λ. We chose λ by finding the best fitting λ that produces stable results for a small training dataset ($P \in \{5, 10, 15, 20, 25\}$). To account for choosing a biased λ that fits well a specific selection of the data, in each of our runs we used different selection of the training data. In the end we found $\lambda = 0.7$ to be the most suitable value for random selection of size P from the training dataset. In Figure 2, when not using λ, we can see that there is a rather large spike in Normalized MSE at $P = 25$. At larger values of P, Normalized MSE for both datasets stabilizes at approximately 0.5. On the other hand, in Figure 3, which uses a regularization term $\lambda = 25$, we can see that the spike at $P = 25$ is much

smaller. On the other hand, at the larger values of P, Normalized MSE for both datasets stabilizes at approximately 1. This is an expected result. The regularization term accounts for the high bias, when the training set is small, hence its use provides better results for smaller values of P. Conversely, for larger training sets its use gives worse results, because for large values of P, bias is already small, hence there is no need to account for bias and it provides worse results.

Division of Workload

	Lubor	Stefan
Program Design	60	40
Program Implementation	60	40
Answering Questions Posed	70	30
Writing the Report	70	30

References

To make this report we used only the material presented in the lectures and MATLAB documentation. The plots were generated in MATLAB.

Appendix

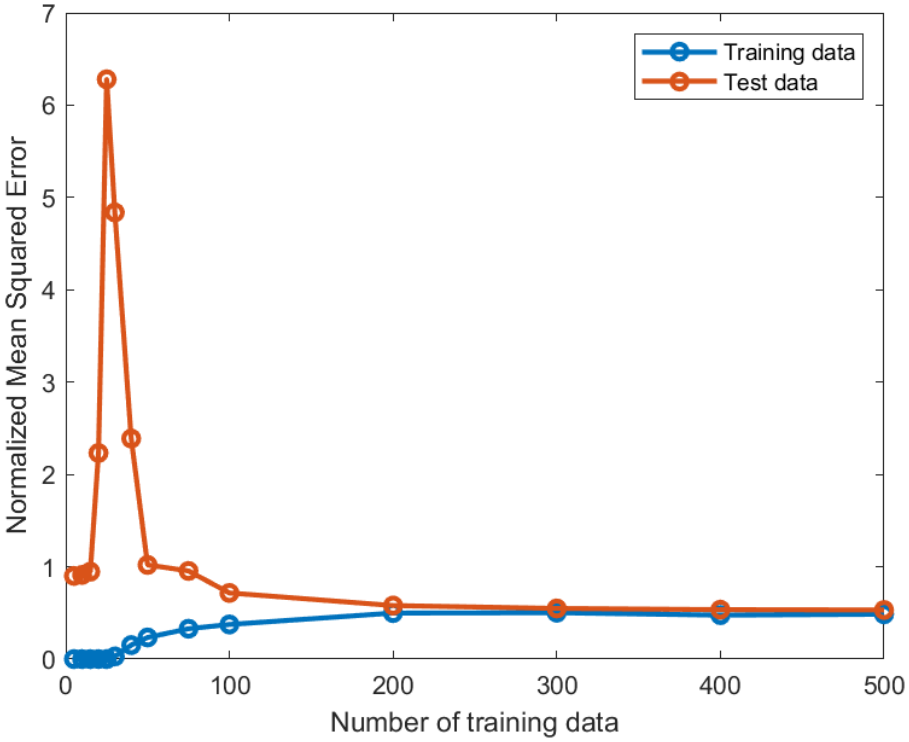

Figure 2: *Normalized MSE for E_{train} and E_{test} without regularization term for values of $P \in \{5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 200, 300, 400, 500\}$*

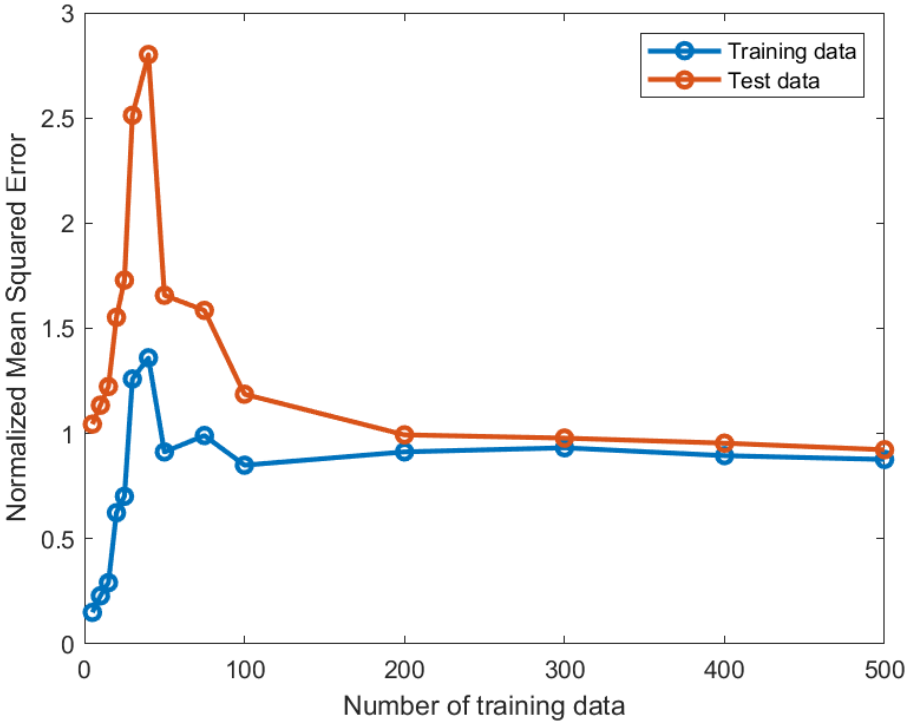

Figure 3: *Normalized MSE for E_{train} and E_{test} with regularization term $\lambda = 0.7$ for values of $P \in \{5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 200, 300, 400, 500\}$*

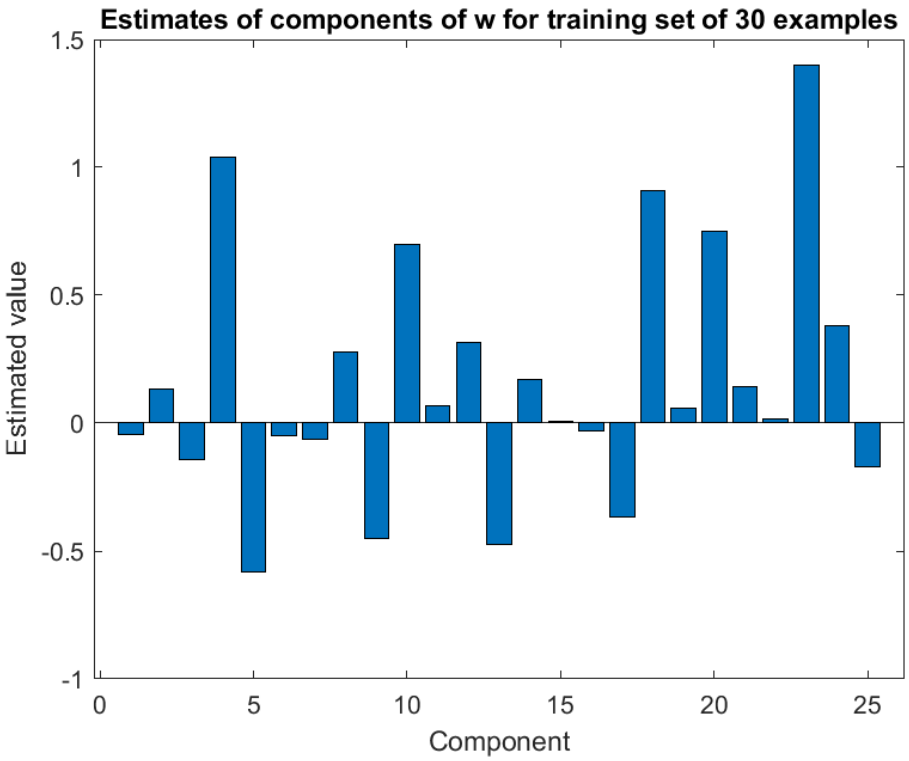

Figure 4: *Estimated values of components of vector w for $P = 30$*

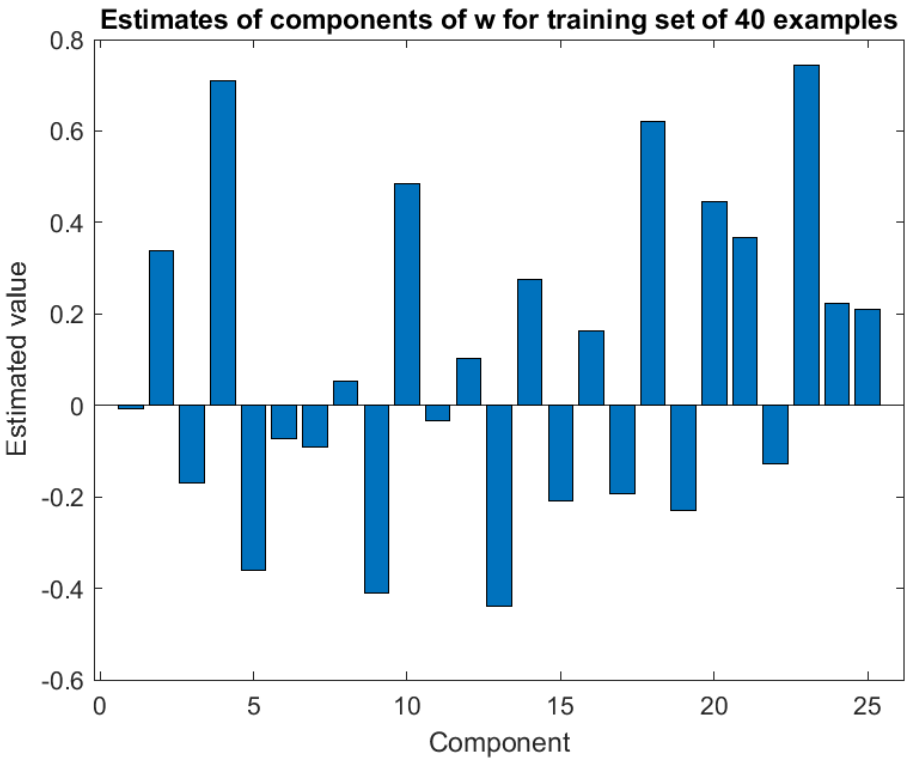

Figure 5: *Estimated values of components of vector w for $P = 40$*

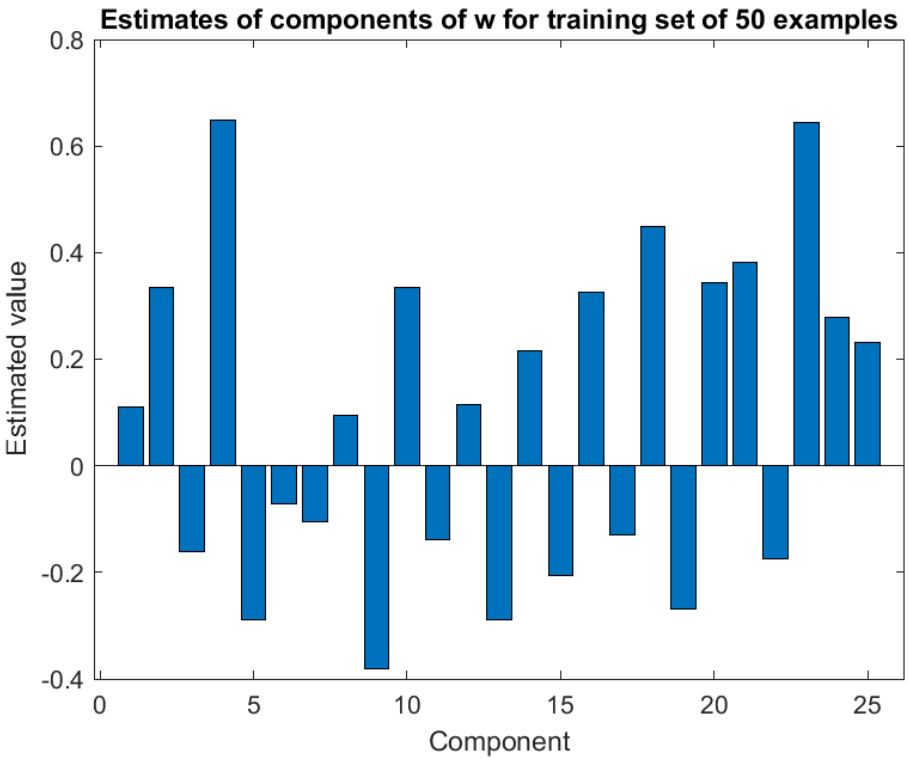

Figure 6: *Estimated values of components of vector w for $P = 50$*

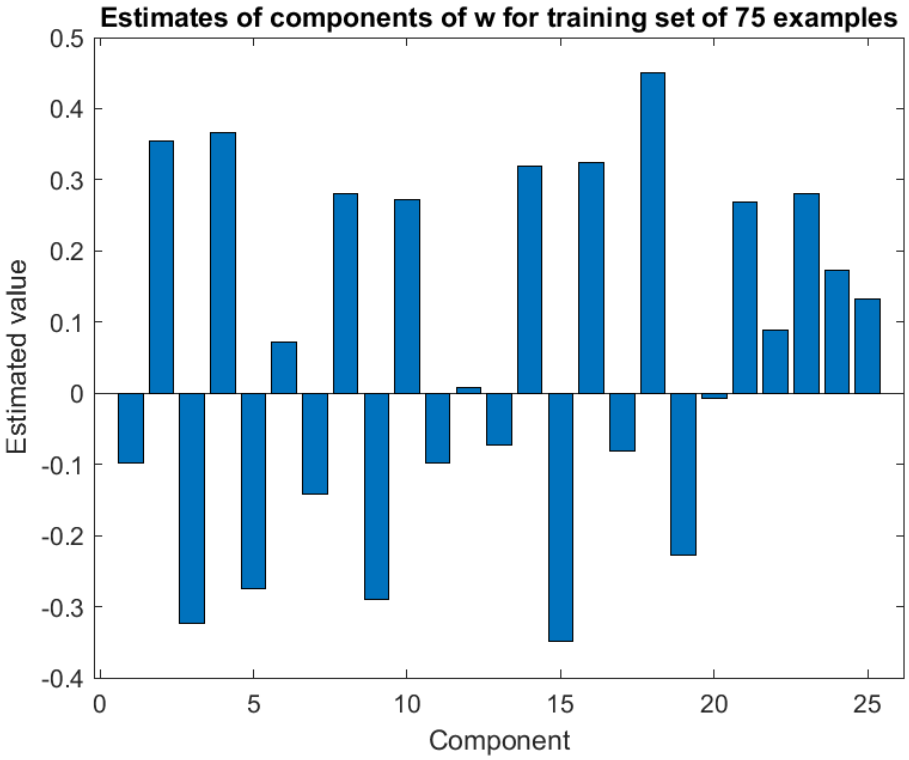

Figure 7: *Estimated values of components of vector w for $P = 75$*

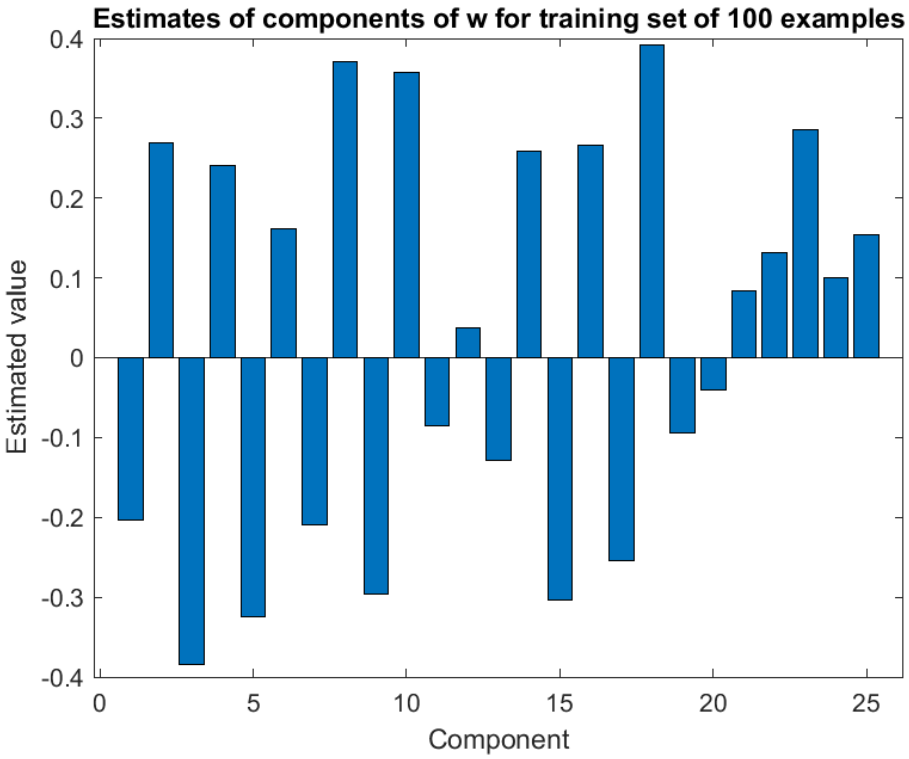

Figure 8: *Estimated values of components of vector w for $P = 100$*

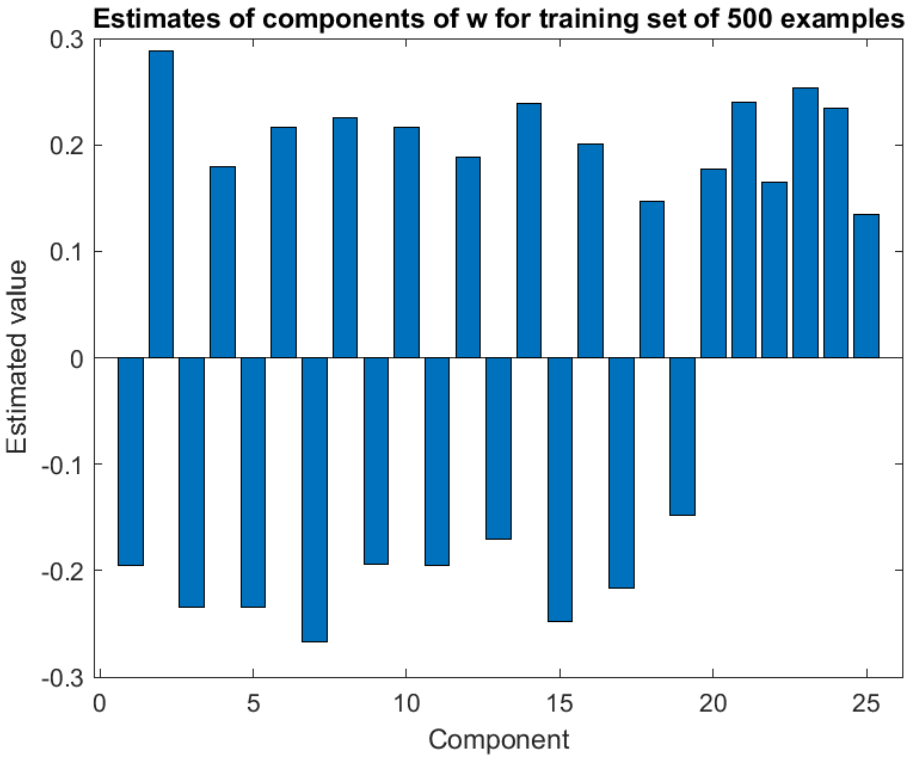

Figure 9: *Estimated values of components of vector w for $P = 500$*