Comparisons of Linear Prediction of Stock Market Averages

Owen Sowatzke

Georgia Institute of Technology
School of Electrical and Computer Engineering

ECE 4271
Spring 2021

Introduction

The goal of this extra credit project was to design and evaluate linear predictors using MATLAB. Specifically, this project examined linear predictors in applications involving stock market prediction. This document details the different experiments performed and then makes conclusions based on the results of each experiment.

Part (i)

The first experiment performed involved using the first ten years (520 weeks) of data to predict the 2018 stock market data. This involved optimizing the number of linear predictor coefficients and then comparing the predicted 2018 stock market with the actual 2018 stock market data.

First, to determine the number of linear predictor coefficients, linear predictor coefficients, `a`, were determined for `p=1:10`. Then, the total squared error was calculated for each `p` value using `E = e'*e`, where `e = x+X*a`. The total squared prediction error is plotted vs the number of linear predictor coefficients (`p`). This result is shown in Figure 1.

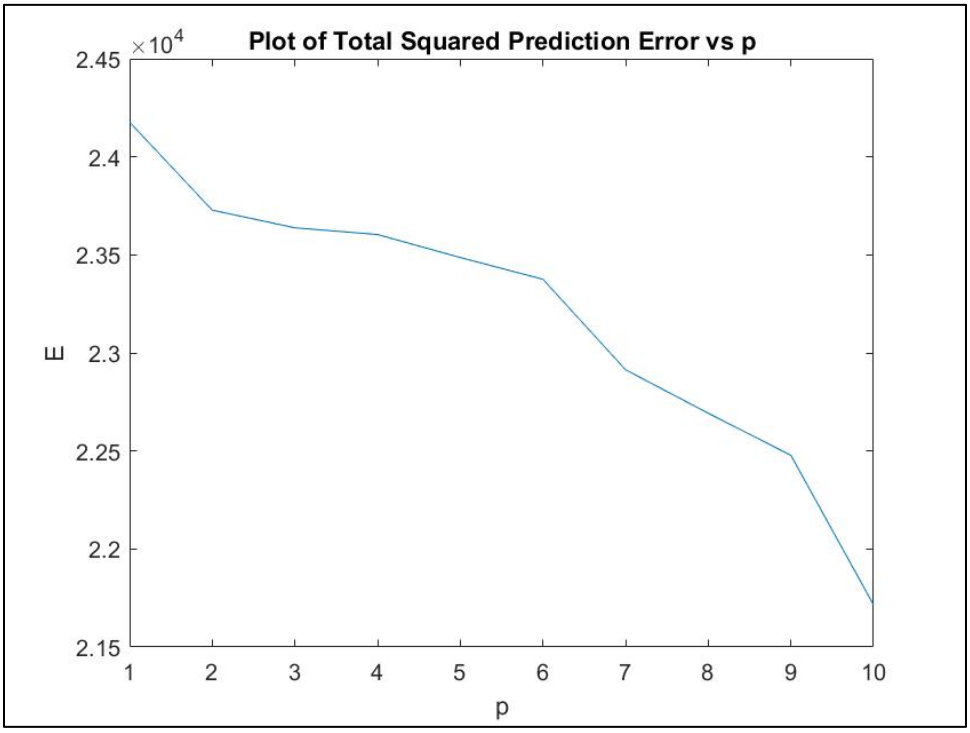

Figure 1 – Plot of the Total Squared Prediction Error vs the Number of Linear Predictor Coefficients (p).

Typically, the value of p is chosen at a knee point, the point after which total squared prediction stops drastically decreasing for each increase in `p`. However, examining the plot, a clear knee point cannot be found. Therefore, `p` is chosen to minimize the total squared prediction error. For our training dataset, this implies `p = 10`.

When optimizing our choice of p, we used the training data set. However, in order to see how our predictor actually performs, we must use the 2018 data set. In order to make a prediction, we use the `filter` command to generate stock market predictions using a weighted sum of previous values. In Figure 2, the predicted 2018 market data is plotted on the same plot as the actual 2018 stock market data.

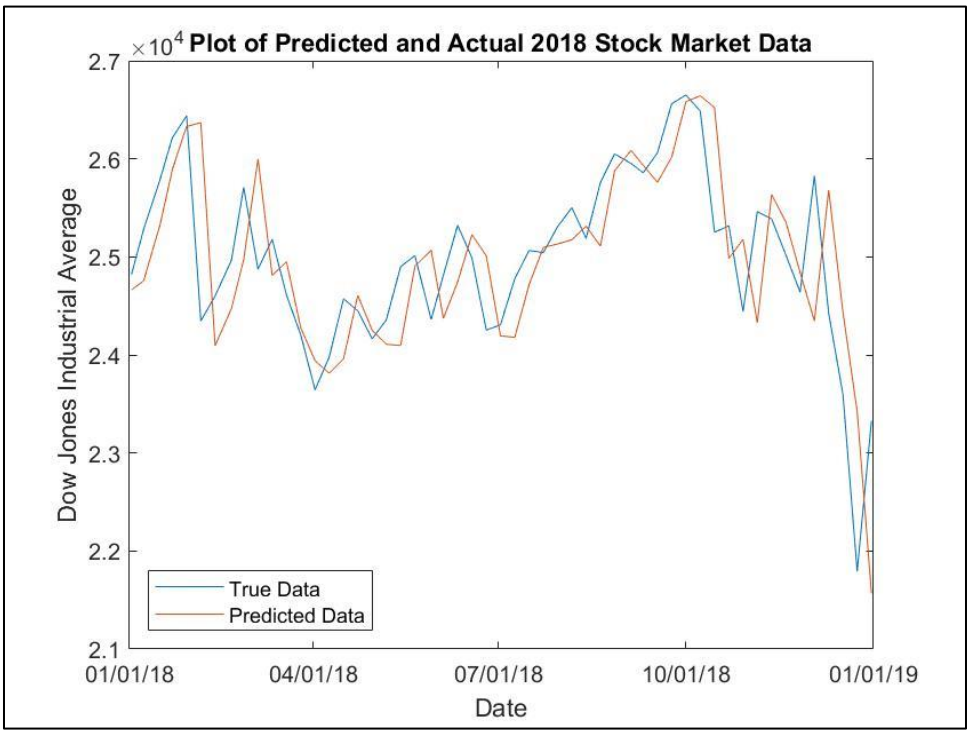

Figure 2 - Plot of the Predicted and Actual Stock Market Data when the Predictor is Designed Using the First 520 weeks of Stock Market Data.

Another metric of prediction performance is the squared error of the predicted data. We determine this metric using the following formula: $(\text{predicted 2018 data} - \text{real 2018 data})^2$. Using this formula, we find that the squared error of the predicted data is 2.57326×10^7.

Part (ii)

In this part, we design our linear predictor using the 2006-2007 stock market as training data. Then, we use this predictor to predict the 2018 stock market data. Using `p=10` as derived in part (i), we determine the predictor coefficients. After our predictor coefficients have been determined, we repeat the analysis of part (i). First, in Figure 3, the predicted 2018 market data is plotted on the same plot as the actual 2018 stock market data.

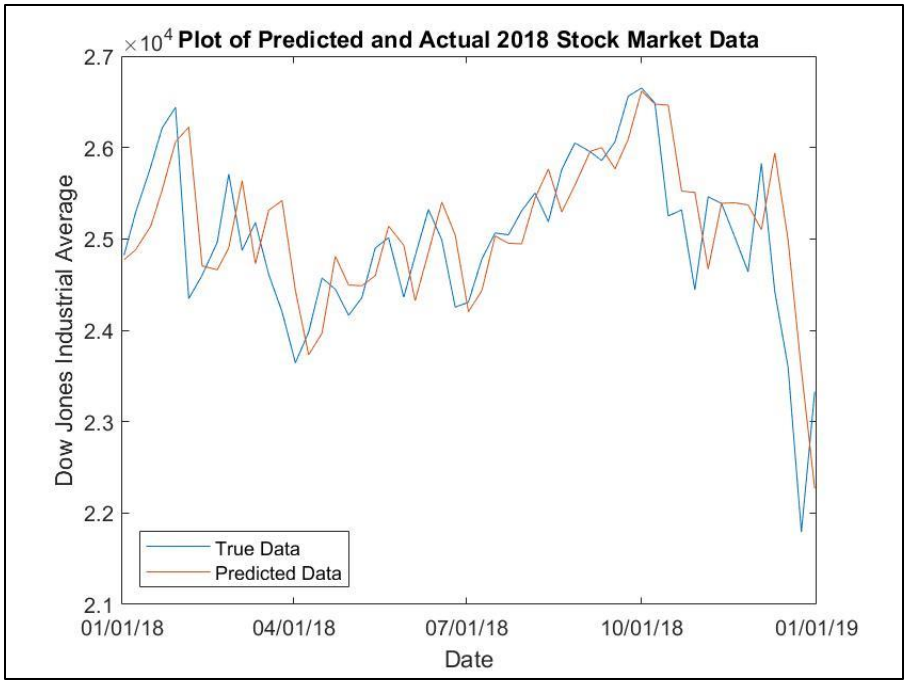

Figure 3 - Plot of the Predicted and Actual Stock Market Data when the Predictor is Designed Using the 2006-2007 Stock Market Data.

Next using the formula in part (i), we also compute the predicted squared error of the new predictor. Repeating the same analysis, we find that the predicted squared error is 2.55591×10^7.

Part (iii)

In this part, we design two linear predictors with 10 linear predictor coefficients (`p=10`). The first predicted uses the data from July – December 2017 as training data, and the second predictor uses the data from January – June 2018 as training data. Each predictor is then used to predict the next 6 months of data. Specifically, the first predictor predicts data from January – June 2018 and the second predictor predicts data from July – December 2018. By concatenating the results of each predictor, we can obtain the entire predicted 2018 dataset. Then, using this data, we can repeat the analysis performed on the predicted datasets. First, in Figure 4, the predicted 2018 market data is plotted on the same plot as the actual 2018 stock market data.

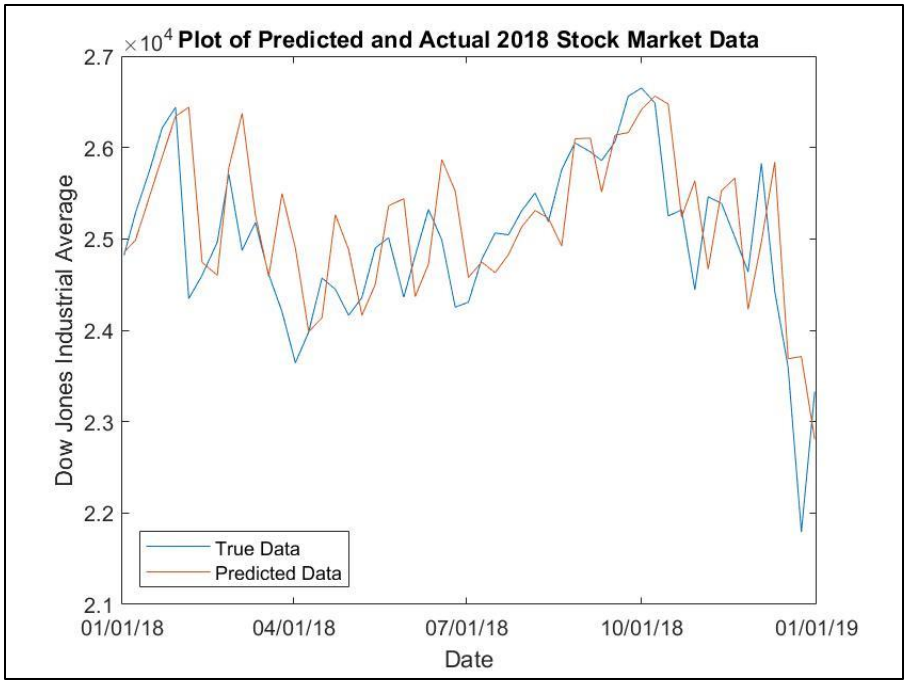

Figure 4 – Plot of the Predicted and Actual Stock Market Data when Two Predictors Trained with the Past 6 Months of Data are Used to Make Predictions.

Next using the formula in part (i), we also compute the predicted squared error of the new predictor. Repeating the same analysis, we find that the predicted squared error is 2.8462×10^7.

Part (iv)

Comparing the results of each predictor, we note that the total squared prediction error is lowest in part (ii), with the prediction error in part (i) only being slightly larger. Additionally, we note that the predictor error in part (iii) is significantly larger than that of both part (i) and part (ii).

There are two primary factors that create differences in the total squared prediction error. The first factor is predictor length. An inadequately trained predictor, as given in part (iii) does not predict as well as the predictors trained with larger data sets. As the predictor gets large (2 years vs 10 years), we note that the 2-year predictor predicts better. This occurs because of data relevancy. This second factor also influences the predictor results. Specifically, comparing part (i) and part (ii), we note that the more recent and relevant data provides better training data and in turn leads to better predictor results.

Conclusion

This document detailed the prediction results of different linear predictors. The trends in resulting prediction errors tell us important information about training a predictor. As described in part (iv), it is important to make sure that our training data set is long enough to capture important trends, but it is also important not to overtrain the predictor, especially with less relevant data.

Appendix A: Project Code

```
%% part (i)

% load stock market data
load('djiaw_2019.mat');

% size of predicted data set
N = 520;

% different values of p to try
p = 1:10;

% array to hold total squared prediction error vs different values of P
E = zeros(1,length(p));

% loop through different values of p
for k = 1:length(p)

    % initialize empty matrix for X
    X = zeros(N-p(k),p(k));

    % form matrix X from dataset
    for m = 1:N-p(k)
        for n = 1:p(k)
            X(m,n) = djiaw_total(m+n-1,2);
        end
    end

    % form vector x from dataset
    x = djiaw_total(p(k)+1:N,2);

    % determine predictor coefficients
    a = -X\x;

    % determine error with linear predictor coefficients
    e = X*a+x;

    % determine total squared predicted error for value of p
    E(k) = e'*e;
end

% plot E vs p
figure
plot(p,E);
xlabel('p');
ylabel('E');
title('Plot of Total Squared Prediction Error vs p');

% chosen value of p from plot
p = 10;

% determine filter coefficient for chosen value of p

% initialize empty matrix for X
X = zeros(N-p,p);
```

```
% form matrix X from dataset
for m = 1:N-p
    for n = 1:p
        X(m,n) = djiaw_total(m+n-1,2);
    end
end

% form vector x from dataset
x = djiaw_total(p+1:N,2);

% determine predictor coefficients
a = -X\x;

% determine first 2018 index
[~,start_index] = min(abs(datenum(2018,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,1,1)
    start_index = start_index + 1;
end

% determine last 2018 index
[~,end_index] = min(abs(datenum(2018,12,31)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,12,31)
    end_index = end_index - 1;
end

% determine 2018 predicted data using filter command
% predictor coefficients must be flipped
xhat = filter(-[0;flip(a)],1,djiaw_total(:,2));
xhat = xhat(start_index:end_index);

% actual 2018 data
x = djiaw_total(start_index:end_index,2);

% date range for plotting
date_range = djiaw_total(start_index:end_index,1);

% plot predicted vs actual values
figure
plot(date_range, x, date_range, xhat);
xlim([date_range(1) date_range(end)]);
datetick('x',2)
legend('True Data', 'Predicted Data', 'Location', 'southwest');
xlabel('Date');
ylabel('Dow Jones Industrial Average');
title('Plot of Predicted and Actual 2018 Stock Market Data');

% calculate the squared error of the predicted data
e = x-xhat;
E = e'*e;

% output squared error of the predicted data
fprintf("Part (i): Squared Error of the Predicted Data: %g\n", E);

%% part (ii)

% use 2006 - 2007 data to predict the 2018 data
% the p value from part (i) is used
```

```
p = 10;

% number of weeks used to train predictor
N = 104;

% determine starting index for 2006 data
[~,start_index] = min(abs(datenum(2006,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2006,1,1)
    start_index = start_index + 1;
end

% initialize empty matrix for X
X = zeros(N-p,p);

% form matrix X from dataset
for m = 1:N-p
    for n = 1:p
        X(m,n) = djiaw_total(start_index+m+n-2,2);
    end
end

%size(X)
% form vector x from dataset
x = djiaw_total(start_index+p:start_index+N-1,2);

% determine predictor coefficients
a = -X\x;

% determine first 2018 index
[~,start_index] = min(abs(datenum(2018,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,1,1)
    start_index = start_index + 1;
end

% determine last 2018 index
[~,end_index] = min(abs(datenum(2018,12,31)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,12,31)
    end_index = end_index - 1;
end

% determine 2018 predicted data using filter command
% predictor coefficients must be flipped
xhat = filter(-[0;flip(a)],1,djiaw_total(:,2));
xhat = xhat(start_index:end_index);

% actual 2018 data
x = djiaw_total(start_index:end_index,2);

% date range for plotting
date_range = djiaw_total(start_index:end_index,1);

% plot predicted vs actual values
figure
plot(date_range, x, date_range, xhat);
xlim([date_range(1) date_range(end)]);
datetick('x',2)
legend('True Data', 'Predicted Data', 'Location', 'southwest');
```

```
xlabel('Date');
ylabel('Dow Jones Industrial Average');
title('Plot of Predicted and Actual 2018 Stock Market Data');

% calculate the squared error of the predicted data
e = x-xhat;
E = e'*e;

% output squared error of the predicted data
fprintf("Part (ii): Squared Error of the Predicted Data: %g\n", E);

%% part (iii)

% use two linear predictors trained with last 6 months of data
% to predict the 2018 data

% the p value from part (i) is used
p = 10;

% First week in July 2017
[~,start_index] = min(abs(datenum(2017,7,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2017,7,1)
    start_index = start_index + 1;
end

% Last week in December 2017
[~,end_index] = min(abs(datenum(2017,12,31)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2017,12,31)
    end_index = end_index - 1;
end

% number of weeks used to train predictor
N = end_index-start_index+1;

% initialize empty matrix for X
X = zeros(N-p,p);

% form matrix X from dataset
for m = 1:N-p
    for n = 1:p
        X(m,n) = djiaw_total(start_index+m+n-2,2);
    end
end

% form vector x from dataset
x = djiaw_total(start_index+p:start_index+N-1,2);

% determine predictor coefficients
a = -X\x;

% First week in January 2018
[~,start_index] = min(abs(datenum(2018,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,1,1)
    start_index = start_index + 1;
end

% Last week in June 2018
```

```
[~,end_index] = min(abs(datenum(2018,6,30)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,6,30)
    end_index = end_index - 1;
end

% determine first set of 2018 predicted data using filter command
% predictor coefficients must be flipped
xhat1 = filter(-[0;flip(a)],1,djiaw_total(:,2));
xhat1 = xhat1(start_index:end_index);

% First week in January 2018
[~,start_index] = min(abs(datenum(2018,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,1,1)
    start_index = start_index + 1;
end

% Last week in June 2018
[~,end_index] = min(abs(datenum(2018,6,30)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,6,30)
    end_index = end_index - 1;
end

% number of weeks used to train predictor
N = end_index-start_index+1;

% initialize empty matrix for X
X = zeros(N-p,p);

% form matrix X from dataset
for m = 1:N-p
    for n = 1:p
        X(m,n) = djiaw_total(start_index+m+n-2,2);
    end
end

% form vector x from dataset
x = djiaw_total(start_index+p:start_index+N-1,2);

% determine predictor coefficients
a = -X\x;

% First week in July 2018
[~,start_index] = min(abs(datenum(2018,7,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,7,1)
    start_index = start_index + 1;
end

% Last week in December 2018
[~,end_index] = min(abs(datenum(2018,12,31)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,12,31)
    end_index = end_index - 1;
end

% determine second set of 2018 predicted data using filter command
% predictor coefficients must be flipped
xhat2 = filter(-[0;flip(a)],1,djiaw_total(:,2));
xhat2 = xhat2(start_index:end_index);
```

```
% determine total 2018 predicted dataset
xhat = [xhat1; xhat2];

% First week in January 2018
[~,start_index] = min(abs(datenum(2018,1,1)-djiaw_total(:,1)));
if djiaw_total(start_index,1) < datenum(2018,1,1)
    start_index = start_index + 1;
end

% Last week in December 2018
[~,end_index] = min(abs(datenum(2018,12,31)-djiaw_total(:,1)));
if djiaw_total(end_index,1) > datenum(2018,12,31)
    end_index = end_index - 1;
end

% actual 2018 data
x = djiaw_total(start_index:end_index,2);

% date range for plotting
date_range = djiaw_total(start_index:end_index,1);

% plot predicted vs actual values
figure
plot(date_range, x, date_range, xhat);
xlim([date_range(1) date_range(end)]);
datetick('x',2)
legend('True Data', 'Predicted Data', 'Location', 'southwest');
xlabel('Date');
ylabel('Dow Jones Industrial Average');
title('Plot of Predicted and Actual 2018 Stock Market Data');

% calculate the squared error of the predicted data
e = x-xhat;
E = e'*e;

% output squared error of the predicted data
fprintf("Part (iii): Squared Error of the Predicted Data ");
fprintf("when last 6 months\n");
fprintf("\tof data is used to train predictor: %g\n", E);
```