GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING

ECE 4271 SPRING 2016
MATLAB PROJECT #3

Linear Prediction of Stock Market Averages

Assigned: Friday, April 1, 2016
Due: Thursday, April 14, 2016
Report (hardcopy) at *beginning* of lecture
Code and Report (zip file) in t-square by 3:05 PM

- This project is to be done *individually*. Collaboration on projects is *not* OK. Each student must work on the projects independently. Each student must develop his or her own analyses and computer code in its entirety. Any project-specific help needed should be sought from the TA and Dr. Alvaro Marenco. Students are not to discuss the theory or approaches to coding the theory with one another, nor are they to assist in debugging each other's work. ***The Georgia Tech honor code and its policies apply.***

  You may ask Dr. Alvaro Marenco or the TA questions regarding theory and implementation of the project, including asking them at the beginning or end of class or during office hours, when others can benefit as well.
- Data required for this project are available for download from t-square, in Resources under Project `#3`. You may also find reference links there, but class notes and CBESP are the primary reference for this project. ***Whether you begin working right away or not, be sure you download the data and make sure you can load it into MATLAB as soon as possible to avoid last minute difficulties.***
- Reports will be graded primarily on completeness in addressing the assignment and quality of results. Reports must be typed, not handwritten; should be well-organized; and must clearly explain answer questions posed in the assignment and explain your results. Reports must include any elements (*e.g.*, specific figures or analyses) called out below. Code must be provided, and should be commented well-enough to be clearly understood.
- Questions or clarifications should be directed to Dr. Alvaro Marenco.[1] Errata, revisions and hints (if any) will be made available via e-mail, t-square, or during class.

[1] e-mail: <alvaro.marenco@gtri.gatech.edu>.

1. PROBLEM

This document contains a copy of the "Linear Prediction on the Stock Market" project from the book *Computer Explorations in Signals and Systems Using MATLAB*, 2nd ed., by Buck, Daniel, and Singer (Prentice-Hall, 2002). This project is very similar to Exercise 1.4 in Chapter 11 of CBESP, but the write-up from Buck *et al* is more complete and also offers a few more hints on how to do the project.

Generally, you will be given weekly data on the closing values of the Dow Jones Industrial Average for the period Oct. 1, 1928 through April 10, 2006. You will be asked to design a linear predictor for this data and then experiment with various investment strategies.

Specifically, work all of the "Basic Problems" (parts (a) – (d)) and "Intermediate Problems" (parts (e) and (f)) in the attachment.

The "Advanced Problems" (parts (g) and (h)) can be worked for extra credit, but are not required.

2. REPORT AND MATLAB CODE REQUIREMENTS

You must submit a hard copy report of your methods and findings that provides the description, graphs, comments, and explanations required by the questions in the reproduced project assignment that follows. Code needed to generate your answers must be submitted.

3. DATA

All of the data you need is obtained by downloading the Winzip file `djia_week_2006.zip` available in t-square in Resources under Project `#3`. When unzipped, this will produce a single MATLAB data file, `djiaw_2006.mat`. When you `load djiaw_2006` in MATLAB, it will create the single 4044x2 matrix `djiaw`. The second column of this array, `djiaw(:,2)`, is the value of the Dow Jones Industrial Average (DJIA) at the beginning of each week from Oct. 1, 1928 to April 10, 2006; the values start at about 240 and end at 11,141. **NOTE***: the "4044" is the number of weeks (amount of data) available and IT IS different than the number used in the reference paper. In the paper, authors use 4861 weeks. You must use 4044 weeks.*

The first column of the row, `djiaw(:,1)`, is the date in MATLAB numerical format. This means, for instance, that midnight on Oct. 1, 1928 is encoded as the number 704462. To interpret this in a normal date format, use the function `datestr` with a `dateform` of your choosing (but probably 0, 1, or 2.). To label the *x* axis of a graph with human-readable dates, use the `datetick` function. (Warning: I don't have much experience with this function, but I find the number and spacing of the resulting tick marks to be kind of hard to control.)

Important for your calculations so we all have consistent results:

i) Basic Problem (a): use 4043 weeks of investment. This assumes that you put in $1000 at end of first week. Do all the increases from that point on. Do the same to figure out APRs and bank gains.

WARNING: **Be sure to use the data** I provide on t-square, not the data file in the CBESP MATLAB files. The CBESP data ends sometime in 1996, thus missing the telecommunication industry "bubble burst" in the stock market of the late 1990s. My file adds the data through the beginning of this week.

Even if you don't start work right away, be sure to download your data file(s) and make sure you can load and work with it as soon as possible!

4. REFERENCES

The only references are any URLs or documentation in the Project #3 area on t-square, the class lecture notes, and the sections of CBESP identified in the lecture notes.

5. MATLAB CODE AND REPORT SUBMISSION

1. You must submit all the codes used to generate plots and calculations that support the answers of the problems stated in the document by Buck et al. Submit a single zip file, named "*FirstInitialLastName.zip*", with all of the MATLAB code and your report. For example, my file would be AMarenco.zip. Do not include any functions that are built into Matlab or any of its toolboxes.
2. Your written report must include answers to all of the problem parts discussed in Section 1 of this assignment. **You must hand in hard copy** of your report at the beginning of class on the due date.

6. GRADING

The 100 points maximum will be allocated as follows:

POINTS	FOR WHAT?
30	General report quality and readability
10 each	Parts (a) – (d) of Project 6.6
15 each	Part (e) and (f) of Project 6.6
10	Extra Credit (parts (g) and (h))
110	**TOTAL**

NOTE: ***In the scanned pages that follow, use*** `djia_week_2006.mat` ***instead of*** `djia.mat`. ***This will load the variable*** `djiaw_2006` ***instead of*** `djia`. ***You will have 78 years of data instead of 94 years of data. You will have 4044 weeks of data instead of 4861 weeks of data.***

■ 6.6 Linear Prediction on the Stock Market

Linear prediction is one of the most widely-used approaches to time series analysis involving applications such as speech coding, seismology, and frequency response modeling. In this exercise, you will learn how linear prediction can be used to design a discrete-time finite-length impulse response (FIR) filter to solve both a time-domain prediction problem and a frequency-domain modeling problem.

In the prediction problem, you observe a signal $x[n]$ and wish to design a system that can predict future values of the signal based solely upon past values. For linear prediction, this system is an FIR filter which computes a prediction based upon a linear combination of past values,

$$\hat{x}[n] = -\sum_{k=1}^{p} a_k x[n-k], \tag{6.27}$$

where $\hat{x}[n]$ is the predicted value of $x[n]$. Since p previous values of the signal are used to formulate the prediction, this is a pth-order predictor. Given a fixed filter order, p, the linear prediction problem is to determine a set of filter coefficients, a_k, that best perform the prediction in Eq. (6.27). The most common measure of determining the "best" coefficients, a_k, is to select those coefficients that minimize the total squared prediction error

$$E = \sum_{n=1}^{N} |e[n]|^2 = \sum_{n=1}^{N} |x[n] - \hat{x}[n]|^2 , \tag{6.28}$$

assuming the sequence $x[n]$ has length N.

Several approaches can be used to solve for the a_k's that minimize E in Eq. (6.28). Perhaps the simplest is to use that MATLAB \ operator for solving simultaneous linear equations. Assuming $N > P$, the linear prediction problem can be posed in matrix form as

$$-\underbrace{\begin{bmatrix} x[1] & \dots & x[p] \\ x[2] & \dots & x[p+1] \\ \vdots & \dots & \vdots \\ x[N-p] & \dots & x[N-1] \end{bmatrix}}_{\mathtt{X}} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{\mathtt{a}} + \underbrace{\begin{bmatrix} e[p+1] \\ e[p+2] \\ \vdots \\ e[N] \end{bmatrix}}_{\mathtt{e}} = \underbrace{\begin{bmatrix} x[p+1] \\ x[p+2] \\ \vdots \\ x[N] \end{bmatrix}}_{\mathtt{x}}, \tag{6.29}$$

or compactly as `-Xa+e=x`. This equation can be used to solve for the vector `a` which minimizes the total squared prediction error, `e'*e`. The convention of incorporating the minus sign on the left hand side of Eq. (6.29) is so that the "prediction-error filter" can be expressed as `e=Xa+x`.

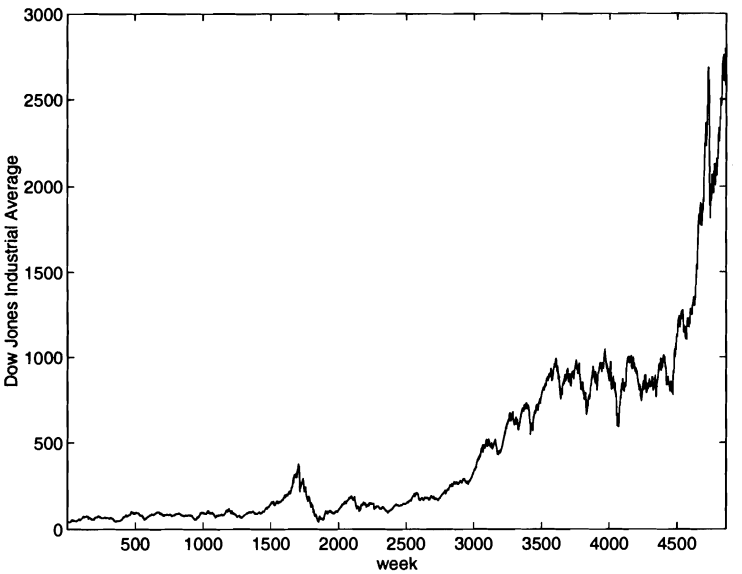

Figure 6.4. The Dow Jones Industrial Average over nearly 5000 weeks.

The problems in this exercise will apply linear prediction to the financial data stored in the file `djia.mat`, which is in the Computer Explorations Toolbox. If this file has been loaded correctly, then typing `who` should result in

```
>> who
Your variables are:
djia
```

where `djia` is the Dow Jones Industrial Average (DJIA) index sampled weekly for approximately 94 years. The DJIA for these weeks is plotted in Figure 6.4.

In this set of problems, you will attempt to make a fortune by investing with the following strategy:

(i) construct a linear predictor based on past DJIA data;

(ii) use your predictor to guess the value of next week's DJIA based on the past p weeks;

(iii) if the DJIA increases by more than the risk-free interest rate earned by a savings account, you invest all of your money in the DJIA;

(iv) if the DJIA increases by less, you put all of your money in the bank.

You will assume that if you decide put all of your money in the DJIA for the week, then you will earn exactly the gain that was earned in the DJIA. For example, if you had \$1000 in the DJIA at week n and the DJIA at week $n+1$ was given by `djia(n+1)`, then at the end of week $n+1$, you have `$1000*djia(n+1)/djia(n)`. Also assume that the savings account always earns $r = 3\%$ annual interest, compounded weekly, i.e., your \$1000 would be worth `$1000*(1+0.03/52)` after one week in the bank.

Basic Problems

(a). Plot the DJIA data on both a linear and a semi-logarithmic scale. Assuming that you started with \$1000 and invested all of your money in the DJIA, how much money would you have at the end of the investment interval (4861 weeks)? If you had put all of your money in the bank at 3% annual percentage rate (APR), compounded weekly, what rate would you need to achieve the same level of performance? If $r = 0.03$ is the APR, then the bank balance after N weeks from a weekly compounded interest bearing account is equal to $g = (1 + r/52)^N$ times the initial balance.

(b). Assume that $p = 3$ and create the vector `x` and matrix `X` in Eq. (6.29) from the first decade of data, i.e., use $N = 520$ weeks. The MATLAB \ operator can be used to solve for the vector `a` that minimizes the inner product `e'*e` in Eq. (6.29). Solve for the linear predictor coefficients using the MATLAB \ operator by `a=-X\x`.

(c). Create the vector of predicted values for the first decade of data using `xhat1=-X*a`. Also create the vector `xhat2` by appropriately using `filter` on the sequence `djia`. Note that the coefficients in the vector `a` are in the reverse of the order required by `filter`. Plot the predicted values on the same set of axes as the actual weekly average. Also determine the total squared error between the predicted and actual values. As a check, do this two ways. First use `e=x+X*a` to compute the prediction error, and then calculate the error by subtracting your predicted sequence `xhat2` from the actual values and make sure that these are the same.

(d). Calculate and plot the total squared prediction error as a function of p for $p = 1, \ldots, 10$. You will have to find the predictor coefficients `a1`, ..., `a10` for each model order p, and then calculate each of the prediction errors. What is an appropriate value for p, i.e., is there a value of p after which the decrease in prediction error is negligible?

Intermediate Problems

(e). Given the predictor you designed based on the first decade of data and the model order you have selected from Part (d), you will now test the investment strategy outlined in the introduction. Give yourself \$1000 at the end of the p-th week, and make 520 trading decisions based on the output of your predictor. First, determine an upper bound on the amount of money you could make. This would be how much you could make if you were omniscient, i.e., if you knew which direction the stock market was going each week and were always invested in the better of either the bank or the DJIA. Now, as a lower bound, calculate how much money you would make if you left all of your money in the bank and earned a gain of (1 + 0.03/52) each week. As another lower bound, determine how much you would make with the "buy-and-hold" strategy, where you put all your money in the DJIA every week. Finally, calculate how much money you would make with your predictor. What is the equivalent APR that the bank would have had to pay you to achieve the same gain as your predictor?

(f). Now use your prediction strategy on the most recent decade in the data, i.e., the last 520 weeks of the DJIA. Calculate how you did and how each of the bounds perform:

best-possible, all in the bank account, and buy-and-hold. Also calculate the equivalent APR for your predictor.

Advanced Problems

(g). Compute the maximum gain possible over all of the data. That is if you knew what the DJIA was going to do each week, and you had the option of making $(1 + 0.03/52)$ in the bank, or the weekly gain in the DJIA, how much could you make over all 4861 weeks? You may be motivated now to look for additional prediction strategies that could come closer to this maximum gain than the simple linear prediction scheme developed in previous parts. For example, you might try updating your predictor coefficients based on the most recent decade before making each prediction. There are several fast algorithms for doing exactly this, like the recursive least squares (RLS) algorithm[3].

(h). Show that the linear predictor can be used to model the DTFT of the sequence $x[n]$ by analytically demonstrating (using Parseval's relation) that the coefficients a_k are chosen to minimize

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\frac{X(e^{j\omega})}{\hat{X}(e^{j\omega})}\right|^2 d\omega, \tag{6.30}$$

where

$$\hat{X}(e^{j\omega}) = \frac{1}{1+\sum_{k=1}^{p} a_k e^{-j\omega k}}.$$

Plot the DTFT of the DJIA sequence and the frequency response of the linear predictor on the same set of axes. Since it is not the difference, but rather the ratio, that is minimized, you should see that $\hat{X}(e^{j\omega})$ has the proper shape, but is off by a scale factor G. Scale $\hat{X}(e^{j\omega})$ by $G = \sum e^2[n]$ and re-plot the two DTFTs. Can you figure out why this value of G was chosen?

[3]For more on recursive least squares, see *Adaptive Filter Theory*, by S. Haykin.