MODELING AND FORECASTING STOCK MARKET PRICES WITH SIGMOIDAL CURVES

A Thesis

Presented to

The Faculty of the Department of Mathematics

California State University, Los Angeles

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Mathematics

By

Daniel Tran

May 2017

The thesis of Daniel Tran is approved.

Dr. Melisa Hendrata, Committee Chair

Dr. Debasree Raychaudhuri

Dr. Xiaohan Zhang

Dr. Grant Fraser, Department Chair

California State University, Los Angeles

May 2017

ABSTRACT

Modeling and Forecasting Stock Market Prices

with Sigmoidal Curves

By

Daniel Tran

Pricing stock market data is difficult because it is inherently noisy and prone to unexpected events. However, stock market data generally exhibits trends in the medium and long term. A typical successful stock index exhibits an initiation phase, rapid growth, and then saturation whereby the price plateaus. Sigmoidal curves can effectively model and forecast stock market data because it can represent nonlinear stock behavior within confidence interval bounds. This thesis surveys various members of the sigmoidal family of curves and determines which curves best fit stock market data. We explore several techniques to filter our data, such as the moving average, single exponential smoothing, and the Hodrick-Prescott filter. We fit the sigmoidal curves to raw data using the Levenberg-Marquardt algorithm. This thesis aggregates these analysis techniques and apply them towards gauging the opportune time point to sell stocks.

ACKNOWLEDGMENTS

The combination of support from family, friends, and colleagues all culminated towards the completion of my thesis.

First and foremost I would like to express gratitude and appreciation towards my mother Kelly Tran, my father David Hao Tran, and my sister Tina Tran for their support.

I would like to thank my graduate advisor Dr. Melisa Hendrata for guiding and mentoring me. Without her mentorship and encouragement, completion of this thesis would not have been possible. I would also thank members of my committee, Dr. Xiaohan Zhang for providing an economics perspective for my thesis, and Dr. Debasree Raychaudhuri for evaluating my thesis.

I would also like to thank everyone else who I may not have mentioned. The random conversations, quick insight and answers all added nuance to my thesis.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

Introduction to Stock Market Behavior and Sigmoidal Curves

The stock market is a system that connects buyers and sellers of stock. Stock is partial ownership of a company in exchange for a certain amount of cash. The owner of stock hopes that the value of stock increases in the future in order to sell stock for cash profit. One may guess that the value of a stock is directly tied to the profits a company can generate, but market exchanges announce the price of a stock through a black box algorithm that depends upon buyer's and seller's bids and offers. This allows for human psychology and market speculation to be priced into stocks. For instance, suppose there exists stock of a company that sells poultry. If a rumor of avian flu speculates drop in profits, the panic may cause owners of the stock to worry and assume a drop in stock price, even though the outbreak may not infect any chickens. Owners of stock may irrationally sell all their shares before the spread of avian flu takes place.

This thesis will not attempt to forecast stock prices in the short term because human psychology and geopolitical events that can affect stock market prices in unpredictable ways. Stock prices with time frames that are less than a year generally exhibit a random walk. Professor Jeremy J. Siegel generated stock market data with a random walk algorithm and asked stock brokers to identify real data mixed with simulated data. Aside from the October 19th, 1987 crash, none of the brokers could distinguish which was real data [18].

Instead, this thesis will explore long term trends, or time scales of at least one year with daily data. Long term prices of stock indices show a positive correlation.

Recall that a stock index is the sum of the price of every unique stock price. The Dow-Jones Industrial Average (DJIA) is a price-weighted index, meaning the prices of 30 large major US industries are summed together, then divided by the number of firms in the index [18]. Siegel fits a best fit line onto 1997 dollars adjusted data and shows the DJIA increases 1.70% per annum. Notice that this time period covers major events in US history, including The Great Depression, World War 2, oil shortages, and many other unpredictable geopolitical events.

Sigmoidal curves were first used for modeling population dynamics. Sigmoidal curves assume that a population will grow at an increasing rate until it passes an inflection point, then the curves approaches a certain limit, called the *carrying capacity*. In terms of demographics, this carrying capacity might be the average mortality of a species or the maximum population a given ecosystem can sustain.

In a similar vein, the economy has finite resources and labor for goods and services, so the growth of any particular company will also have a carrying capacity in an economic environment. This paper will demonstrate that sigmoidal curves may be utilized as a tool to predict long term stock market prices.

Stock market data is noisy because of market volatility and general uncertainty about future market conditions. This thesis will follow assumptions outlined by Choliz (2007). Choliz characterizes stock market values following three phases: emergent, inflection, and saturation. The emergent phase is when a stock is initially accelerating in growth, the inflection phase is when the growth rate becomes linear, and the saturation phase is when growth decelerates. Stocks have a lower bound of zero because stock prices cannot be negative. Stocks also have a rapid phase of growth

with an inflection point that defines a decrease in the rate of stock market growth. Stocks also have an upper bound once it saturates the market.

Our sigmoidal growth curve models need to have variable growth rates and asymmetry [2]. Schumpeter observes in advanced economies over two centuries suggest that periods of expansion are generally longer than periods of decline. In this thesis, we will use the Logistic, Gompertz, Weibull, Generalized Logistic, and Chapman-Richards equations as the models to fit stock market data. All of these curves have a positive horizontal asymptotes which define the carrying capacity and a horizontal asymptote that defines a stock market price of minimum of \$0. All of these sigmoidal curves exhibit an emergent, inflection, and saturation phase. The inflection points of each of these sigmoidal curves can vary, allowing for asymmetric fits. The Logistic and Gompertz equation have inflection points that are multiplied by a constant. The Weibull, Generalized Logistic, and Chapman-Richards are multiplied by a variable, so these three sigmoidal curves provide flexibility when fitting and forecasting stock market data. This thesis will show that the last three sigmoidal provide better fits and forecasts than the classical Logistic and Gompertz equations.

CHAPTER 2

Various Members of the Sigmoidal Family of Curves

Sigmoidal curves have initially been used to model the growth of biological species populating a given ecosystem with limited resources. The economy similarly has finite resources for goods and services, so the growth of any particular company must have a carrying capacity in unconstrained economic environment. This metaphor motivates the use of sigmoidal curves to model stock market prices. We need to find a function that accelerates initially as it grows, then decelerates as the size of a stock approaches a limit. The sigmoidal curves exhibit this pattern. The term "sigmoidal" literally means s-shaped.

The inflection point is the turning point where the rate of growth starts to decrease. The Logistic and Gompertz equations are classic examples of sigmoidal curves. The problem with these functions is that the inflection point, $Y_{inflection}$, is a fixed product between the carrying capacity and a constant. The Generalized Logistic, Chapman-Richards and Weibull equations have inflection points that are dependent upon some variables, so the inflection point is adjustable along the x-axis and y-axis.

This chapter will explore the phase diagram and instantaneous growth rate for each type of curves. The phase diagram is the derivative of the closed form solution, $\frac{dY_t}{dt}$, whose unit is $\frac{\text{[amount]}}{\text{[unit time]}}$. The inflection point occurs at the maximum value of the phase diagram. In all of our graphs, when Y_t is at the carrying capacity Y_∞, the growth rate must necessarily be zero. Growth does not occur past the carrying capacity for sigmoidal curves.

The instantaneous growth rate divides $\frac{dY_t}{dt}$ with Y_t, with units of $\frac{[1]}{\text{[unit time]}}$. This

can be interpreted as the percentage change of Y_t per unit time forward.

2.1 The Logistic Model

Given the closed form of the logistic model:

$$Y(t) = Y_t = \frac{Y_\infty}{1 + \alpha e^{-\beta t}}, \qquad t \geq 0 \tag{2.1}$$

where α, β are constant growth parameters, with β being the maximum growth rate, and Y_∞ is the carrying capacity. The derivatives for the logistic models are given by

$$\frac{dY_t}{dt} = \frac{\beta}{Y_\infty} Y_t (Y_\infty - Y_t) \tag{2.2}$$

$$\frac{d^2 Y_t}{dt^2} = \frac{\beta}{Y_\infty} (Y_\infty - 2Y_t) \frac{dY_t}{dt}. \tag{2.3}$$

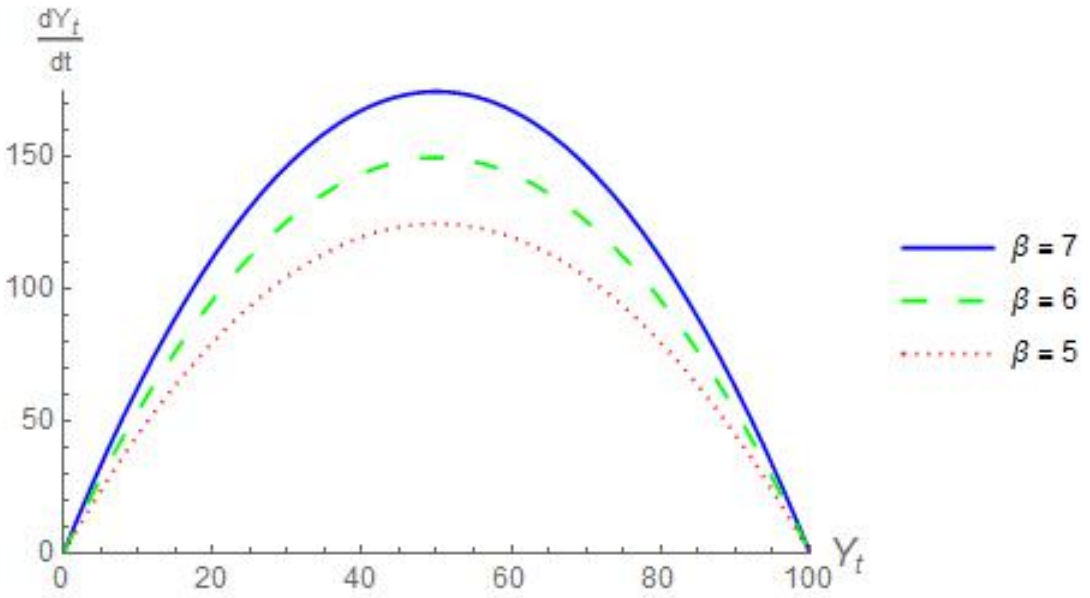


Figure 2.1: Phase diagram of logistic curve with parameters $\beta = 5, 6, 7$, $Y_\infty = 100$.

Due to symmetry, the maximum of $\frac{dY_t}{dt}$ occurs at the midpoint between 0 and Y_∞, as shown in the phase diagram in Figure 2.1. Even though the height of the

maximum can change with β, the inflection point $t_{inflection}$ is fixed. The y-value of the inflection point occurs at $Y_t = \frac{Y_\infty}{2}$, that is when $\frac{d^2Y_t}{dt^2} = 0$. Substituting this value into the closed form of the logistic equation (2.1) gives $t = \frac{1}{\beta}\ln(\alpha)$. Hence, the inflection point occurs at

$$(t_{inflection}, Y_{inflection}) = \left(\frac{1}{\beta}\ln(\alpha), \frac{Y_\infty}{2}\right). \tag{2.4}$$

The instantaneous growth rate is

$$\frac{\frac{dY_t}{dt}}{Y_t} = \frac{\beta}{Y_\infty}(Y_\infty - Y_t). \tag{2.5}$$

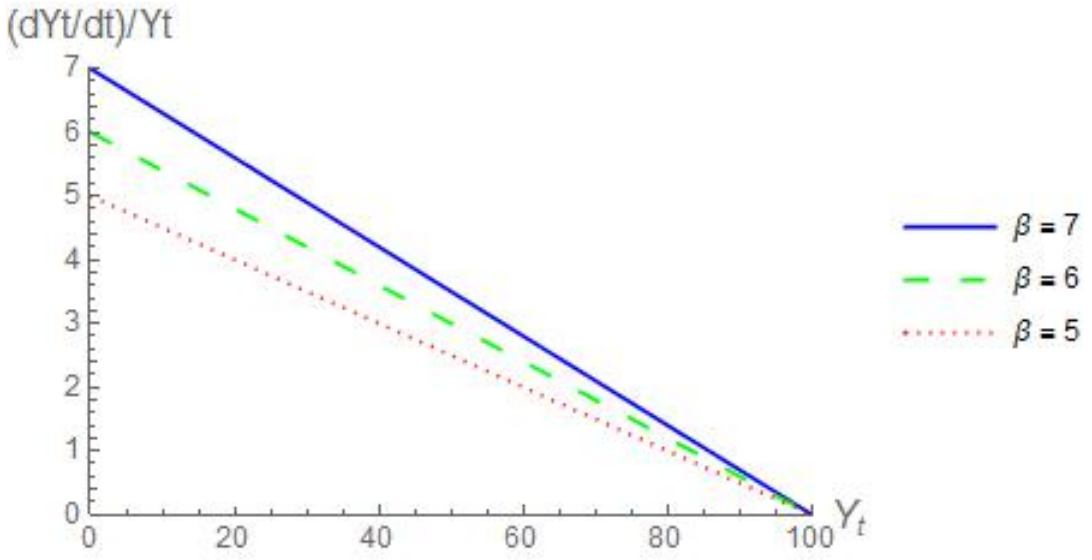


Figure 2.2: Instantaneous growth rate with logistic curve with parameters $\beta = 5$, 6, 7, $Y_\infty = 100$.

Notice that $Y_{inflection}$ is dependent only on the carrying capacity Y_∞, sometimes referred to as the ceiling value. To realistically model stock prices, we need functions that are more malleable where we can adjust the inflection points, and whose curves that are not necessarily symmetric.

2.2 The Gompertz Model

The closed form of the Gompertz model is:

$$Y_t = Y_\infty e^{-\alpha e^{-\beta t}}, \qquad t \geq 0 \tag{2.6}$$

where α and β are constant growth parameters, and $Y_\infty > 0$.

Manipulation of the closed form solution (2.6) will be useful for understanding the derivatives of the Gompertz equations. Note that

$$\begin{aligned} Y_t &= Y_\infty e^{-\alpha e^{-\beta t}} \\ \frac{Y_t}{Y_\infty} &= e^{-\alpha e^{-\beta t}} \\ \frac{Y_\infty}{Y_t} &= e^{\alpha e^{-\beta t}} \\ \ln\left(\frac{Y_\infty}{Y_t}\right) &= \alpha e^{-\beta t} \\ e^{-\beta t} &= \frac{1}{\alpha}\ln\left(\frac{Y_\infty}{Y_t}\right) \end{aligned}$$

The derivatives of the Gompertz equation are:

$$\frac{dY_t}{dt} = \alpha\beta e^{-\beta t} Y_t = \beta Y_t \ln\left(\frac{Y_\infty}{Y_t}\right) \tag{2.7}$$

$$\frac{d^2 Y_t}{dt^2} = \alpha\beta^2 e^{-\beta t}(\alpha e^{-\beta t} - 1)Y_t = \beta^2 \ln\left(\frac{Y_\infty}{Y_t}\right)\Big(\ln\left(\frac{Y_\infty}{Y_t}\right) - 1\Big)Y_t \tag{2.8}$$

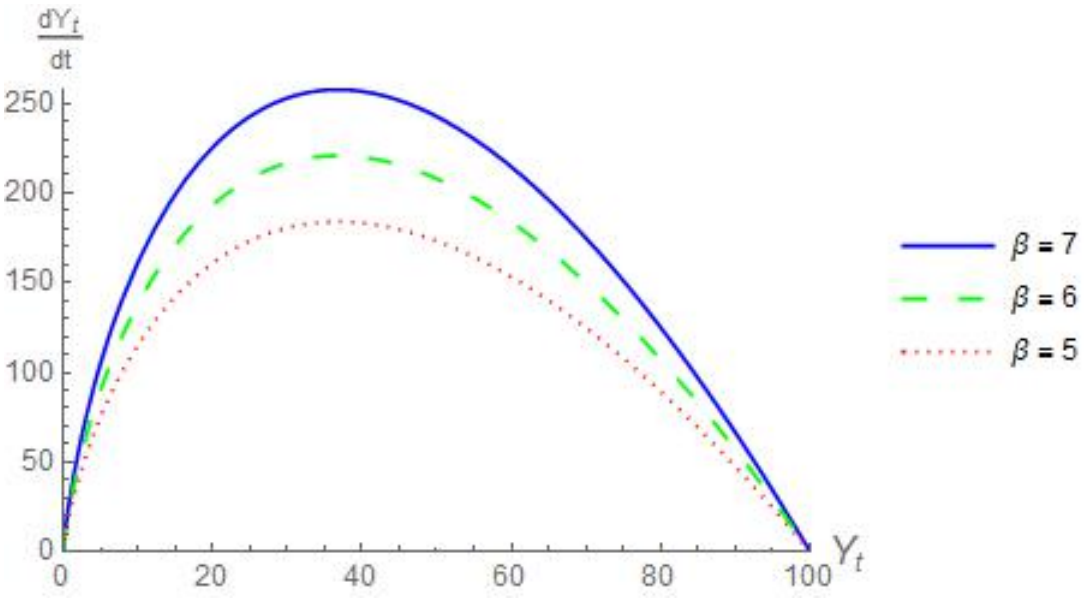


Figure 2.3: Phase diagram of Gompertz model with parameters $\beta = 5, 6, 7$, $Y_\infty = 100$.

The phase diagram shows that the inflection point occurs at a fixed point on the x-axis, the same characteristic as the logistic equation.

The instantaneous growth rate is:

$$\frac{\frac{dY_t}{dt}}{Y_t} = \alpha\beta e^{-\beta t} = \beta(\ln Y_\infty - \ln Y_t). \tag{2.9}$$

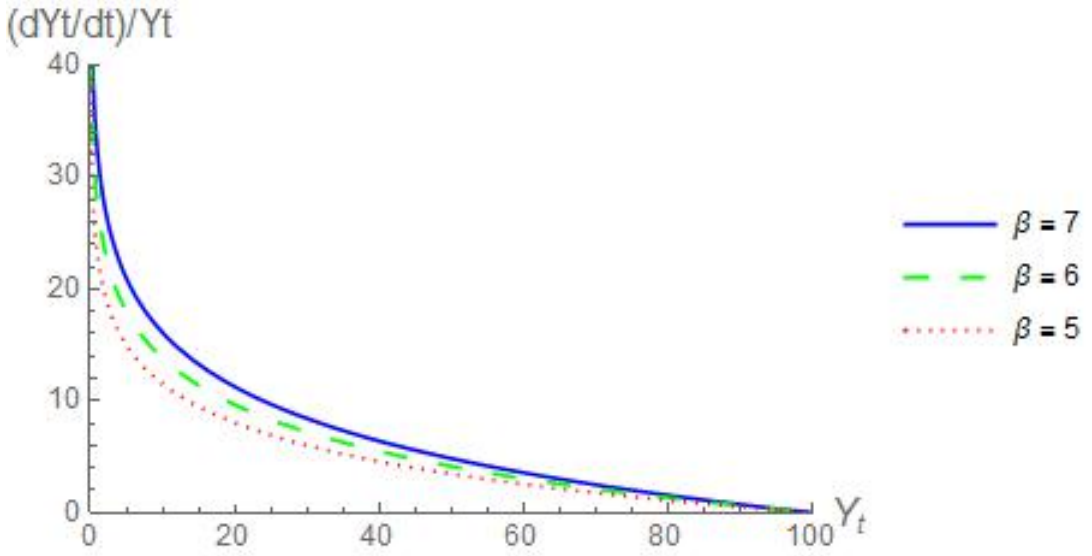


Figure 2.4: Instantaneous growth rate of Gompertz model with parameters $\beta = 5$, 6, 7, $Y_\infty = 100$.

The instantaneous growth rate has a vertical asymptote at $Y_t = 0$. This is no matter for applications towards the stock market because a stock price is de-listed at zero. Our sigmoidal curves assume that stock will always be greater than zero.

To calculate the inflection point:

$$0 = \alpha e^{-\beta t} - 1$$

$$1 = \alpha e^{-\beta t}$$

$$\frac{1}{\alpha} = e^{-\beta t}$$

$$\alpha = e^{\beta t}$$

$$\beta t = \ln(\alpha)$$

$$t_{inflection} = \frac{\ln(\alpha)}{\beta}$$

Substituting this value into the closed form solution (2.6), we obtain

$$Y_t = Y_\infty e^{-\alpha e^{-\beta \frac{\ln(\alpha)}{\beta}}}$$

$$Y_t = Y_\infty e^{-\alpha e^{-\ln(\alpha)}}$$

$$Y_t = Y_\infty e^{-\alpha \frac{1}{\alpha}}$$

$$Y_{inflection} = Y_\infty e^{-1}.$$

So the inflection point occurs at:

$$(t_{inflection}, Y_{inflection}) = \left(\frac{\ln(\alpha)}{\beta}, Y_\infty e^{-1} \right). \tag{2.10}$$

2.3 The Generalized Logistic Equation

As derived in Appendix C, the closed form solution of the generalized logistic equation is given by:

$$Y_t = \frac{Y_\infty}{(1 + \alpha e^{-\beta r t})^{\frac{1}{r}}}, \text{ for } t \geq 0 \text{ and } \alpha = \frac{Y_\infty^r}{Y_0^r} - 1. \tag{2.11}$$

Note that the derivatives are:

$$\frac{dY_t}{dt} = \beta Y_t \left[1 - \left(\frac{Y_t}{Y_\infty} \right)^r \right] \tag{2.12}$$

$$\frac{d^2 Y_t}{dt^2} = \beta^2 Y_t \left[1 - \left(\frac{Y_t}{Y_\infty} \right)^r \right] \left[1 - (r+1) \left(\frac{Y_t}{Y_\infty} \right)^r \right] \tag{2.13}$$

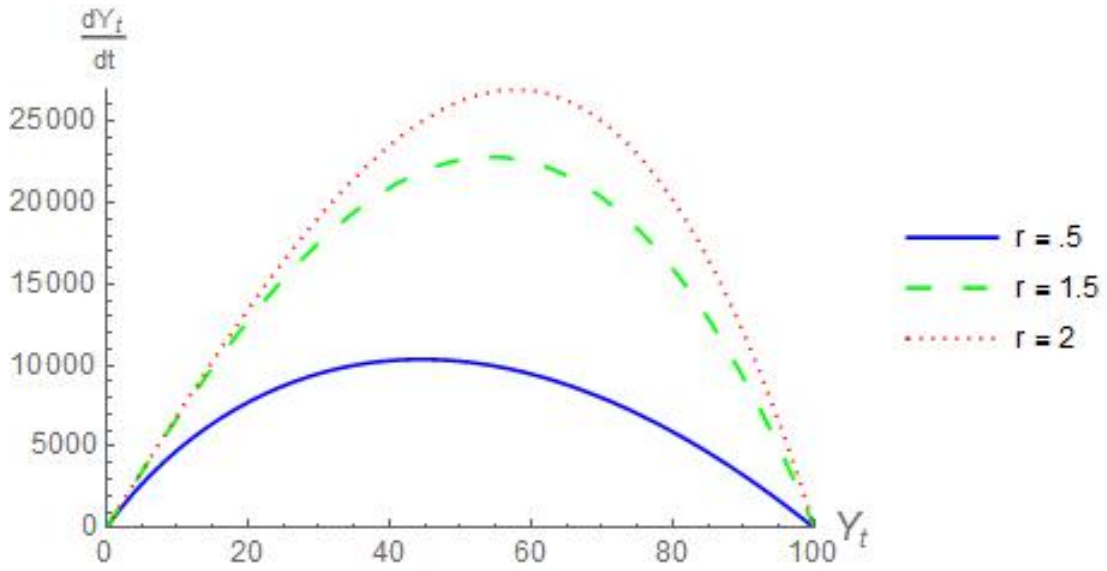


Figure 2.5: Phase diagram of generalized logistic with parameters $\beta = 7, r = 0.5, 1.5, 2, Y_\infty = 100$.

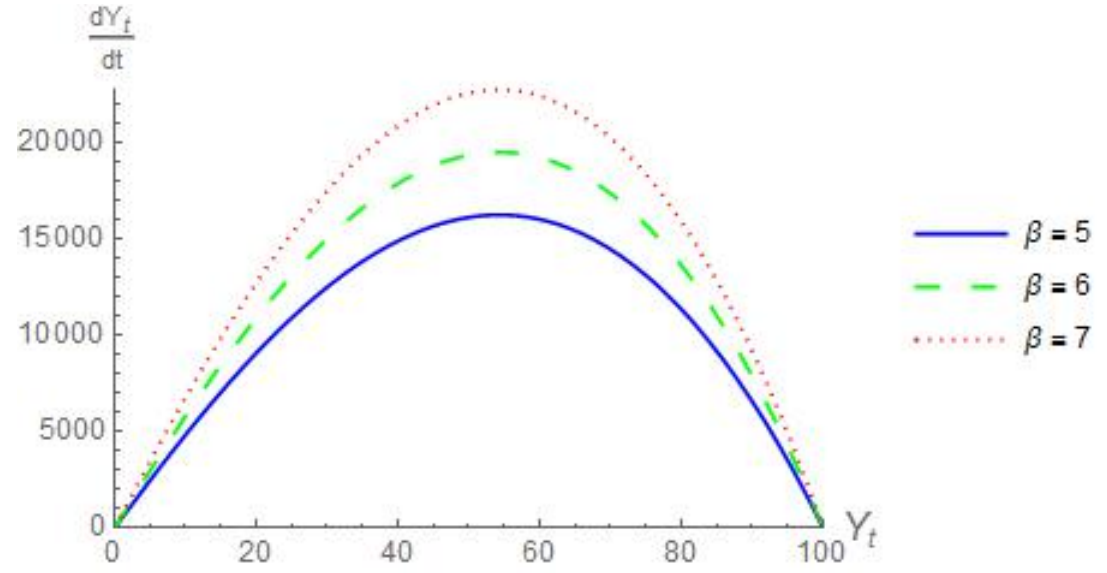


Figure 2.6: Phase diagram of generalized logistic with parameters $\beta = 5, 6, 7, r = 1.5, Y_\infty = 100$.

The phase diagrams for the generalized logistic equation show it is possible to

shift the maximum along the x-axis. The value of the r parameter allows for change of the inflection point to correspond to various values of Y_t.

The instantaneous growth rate is:

$$\frac{\frac{dY_t}{dt}}{Y_t} = \beta \left[1 - \left(\frac{Y_t}{Y_\infty} \right)^r \right] \tag{2.14}$$

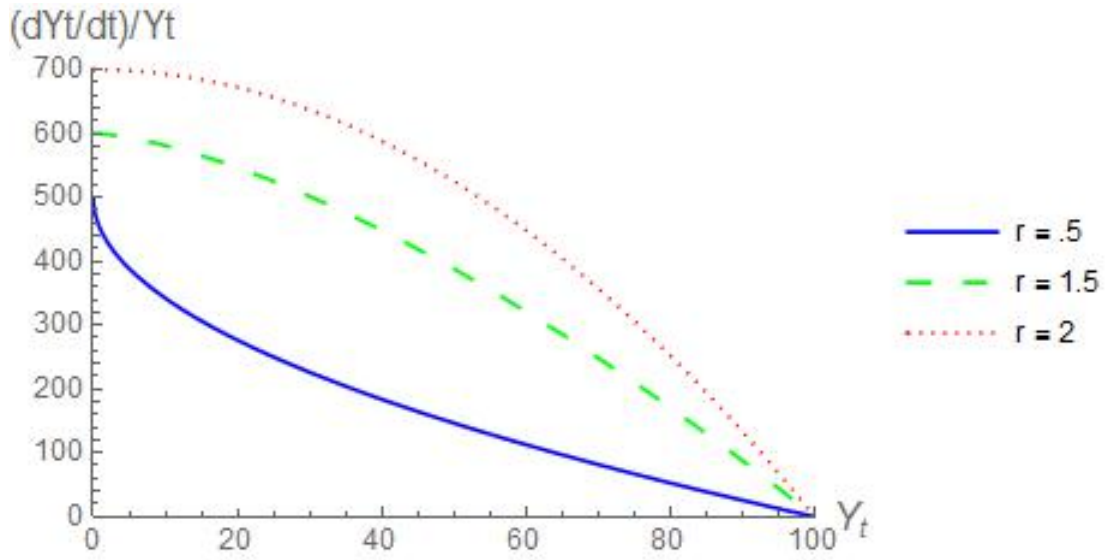


Figure 2.7: Instantaneous growth rate of generalized logistic with parameters $\beta = 7, r = 0.5, 1.5, 2, Y_\infty = 100$.

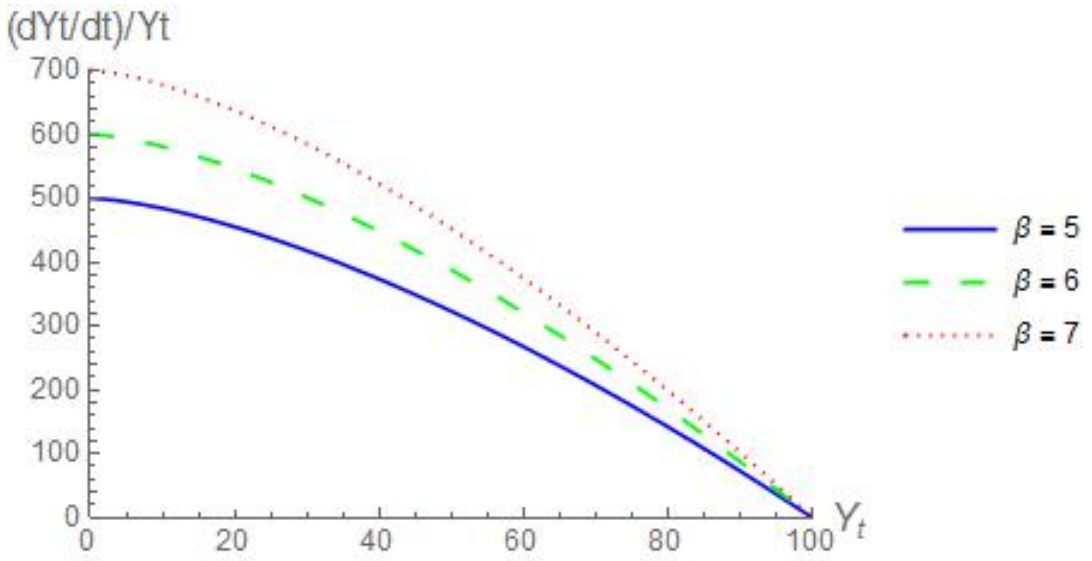


Figure 2.8: Instantaneous growth rate of generalized logistic with parameters $\beta = 5, 6, 7, r = 1.5, Y_\infty = 100$.

We can change the concavity of the instantaneous growth rate. When $r > 1$, the instantaneous growth rate decreases at an increasing rate. When $r < 1$, the instantaneous growth rate decreases at a decreasing rate. When $r = 1$, we get back the logistic equation.

To calculate the inflection point:

$$0 = 1 - (r+1)\left(\frac{Y_t}{Y_\infty}\right)^r$$

$$1 = (r+1)\left(\frac{Y_t}{Y_\infty}\right)^r$$

$$\frac{1}{r+1} = \left(\frac{Y_t}{Y_\infty}\right)^r$$

$$\frac{1}{(r+1)^{1/r}} = \frac{Y_t}{Y_\infty}$$

$$Y_{inflection} = \frac{Y_\infty}{(r+1)^{1/r}}$$

To calculate t, substitute $Y_{inflection}$ into the closed form solution (2.11):

$$\begin{aligned}
\frac{Y_\infty}{(r+1)^{1/r}} &= \frac{Y_\infty}{(1+\alpha e^{-\beta r t})^{1/r}} \\
(r+1)^{1/r} &= (1+\alpha e^{-\beta r t})^{1/r} \\
r &= \alpha e^{-\beta r t} \\
\frac{r}{\alpha} &= e^{-\beta r t} \\
\ln\left(\frac{\alpha}{r}\right) &= \beta r t \\
t_{inflection} &= \frac{1}{\beta r}\ln\left(\frac{\alpha}{r}\right).
\end{aligned}$$

So the inflection point for this curve is:

$$(t_{inflection}, Y_{inflection}) = \left(\frac{1}{\beta r}\ln\left(\frac{\alpha}{r}\right), \frac{Y_\infty}{(r+1)^{1/r}}\right). \tag{2.15}$$

2.4 The Chapman-Richards Equation

The closed form solution of the Chapman–Richards equation is [13]:

$$Y_t = Y_\infty[1 - ae^{-\lambda t}]^m, t \geq 0. \tag{2.16}$$

Before calculating the derivatives, we will need the following equations from the closed form solution.

$$\left(\frac{Y_t}{Y_\infty}\right) = [1 - ae^{-\lambda t}]^m \tag{2.17}$$

$$\left(\frac{Y_t}{Y_\infty}\right)^{1/m} = 1 - ae^{-\lambda t} \tag{2.18}$$

The first and second derivatives are:

$$
\begin{aligned}
\frac{dY_t}{dt} &= Y_\infty a\lambda m e^{-\lambda t}(1-ae^{-\lambda t})^{m-1} \\
&= m\lambda a e^{-\lambda t} Y_\infty \frac{(1-ae^{-\lambda t})^m}{(1-ae^{-\lambda t})} \\
&= m\lambda Y_t \frac{ae^{-\lambda t}}{(1-ae^{-\lambda t})} \\
&= m\lambda Y_t \left(1-\left(\frac{Y_t}{Y_\infty}\right)^{1/m}\right)\left(\frac{Y_\infty}{Y_t}\right)^{1/m} \\
&= m\lambda Y_t \left(\left(\frac{Y_\infty}{Y_t}\right)^{1/m}-1\right)
\end{aligned}
\tag{2.19}
$$

$$
\frac{d^2Y_t}{dt^2} = m\lambda^2 Y_t \left(\left(\frac{Y_\infty}{Y_t}\right)^{1/m}-1\right)\left[(m-1)\left(\frac{Y_\infty}{Y_t}\right)^{1/m}-m\right] \tag{2.20}
$$

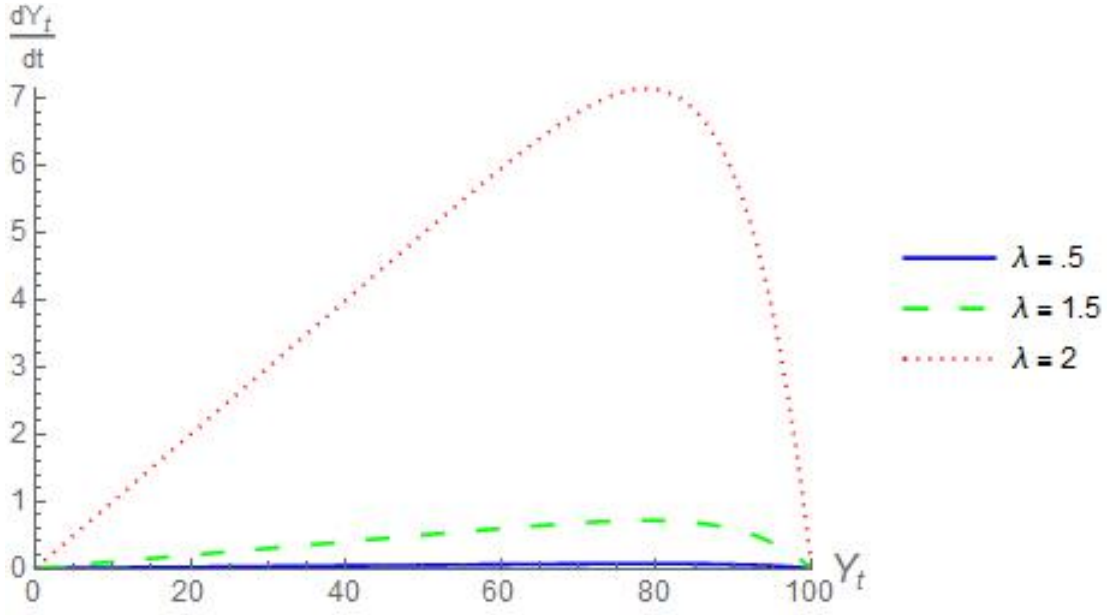


Figure 2.9: Chapman–Richards phase diagram with $m = -.1, \lambda = .01, .1, 1, Y_\infty = 100$.

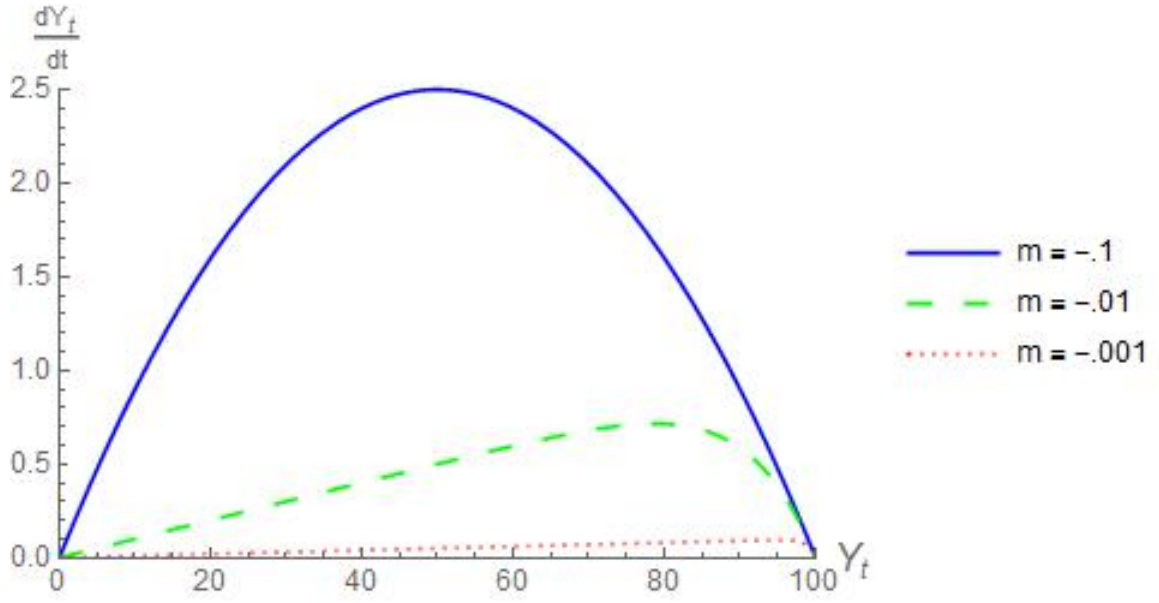


Figure 2.10: Chapman–Richards phase diagram with $m = -1, -.1, -.01, \lambda = .1, Y_\infty = 100$.

To calculate the inflection point:

$$
\begin{aligned}
0 &= (m-1)\left(\frac{Y_\infty}{Y_t}\right)^{1/m} - m \\
\left(\frac{Y_\infty}{Y_t}\right)^{1/m} &= \frac{m}{m-1} \\
\left(\frac{Y_\infty}{Y_t}\right) &= \left(\frac{m}{m-1}\right)^m \\
\frac{Y_t}{Y_\infty} &= \left(\frac{m-1}{m}\right)^m \\
Y_{inflection} &= Y_\infty \left(\frac{m-1}{m}\right)^m
\end{aligned}
$$

Direct substitution of $Y_{inflection}$ to the closed form (2.16) gives:

$$\begin{aligned}
Y_\infty \left(\frac{m-1}{m}\right)^m &= Y_\infty[1-ae^{-\lambda t}]^m \\
\frac{m-1}{m} &= 1-ae^{-\lambda t} \\
1-\frac{1}{m} &= 1-ae^{-\lambda t} \\
\frac{1}{am} &= e^{-\lambda t} \\
am &= e^{\lambda t} \\
\ln(am) &= \lambda t \\
t_{inflection} &= \frac{\ln(am)}{\lambda}
\end{aligned}$$

So the inflection point for this curve is:

$$(t_{inflection}, Y_{inflection}) = \left(\frac{\ln(am)}{\lambda}, Y_\infty \left(\frac{m-1}{m}\right)^m\right) \tag{2.21}$$

The instantaneous growth rate from equation (2.19) gives:

$$\frac{\frac{dY_t}{dt}}{Y_t} = m\lambda\left(\left(\frac{Y_\infty}{Y_t}\right)^{1/m} - 1\right) \tag{2.22}$$

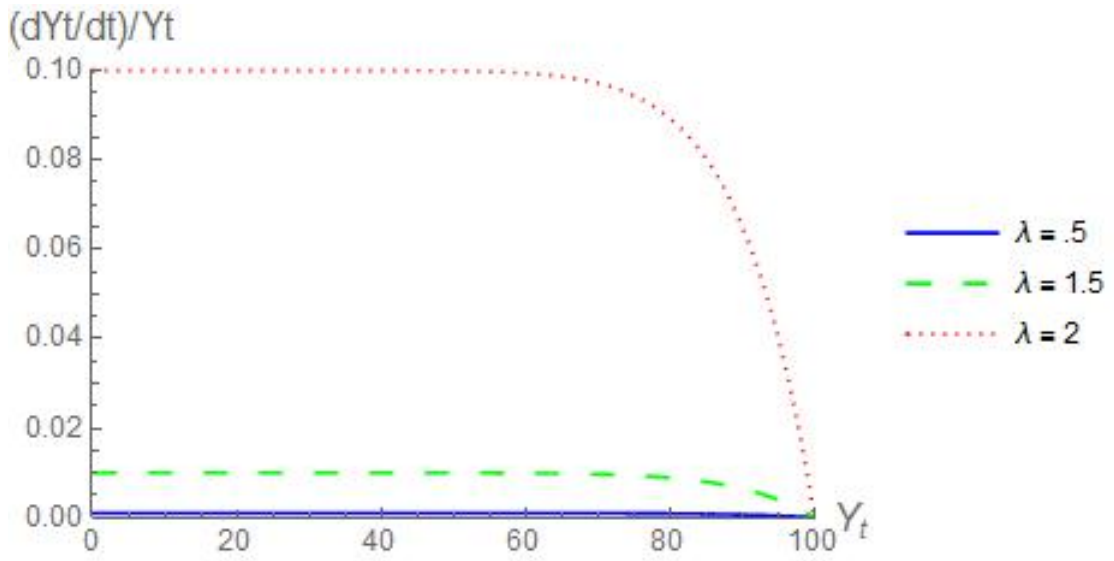


Figure 2.11: Chapman–Richards instantaneous growth rate with $m = -.1, \lambda = .01, .1, 1, Y_\infty = 100$.

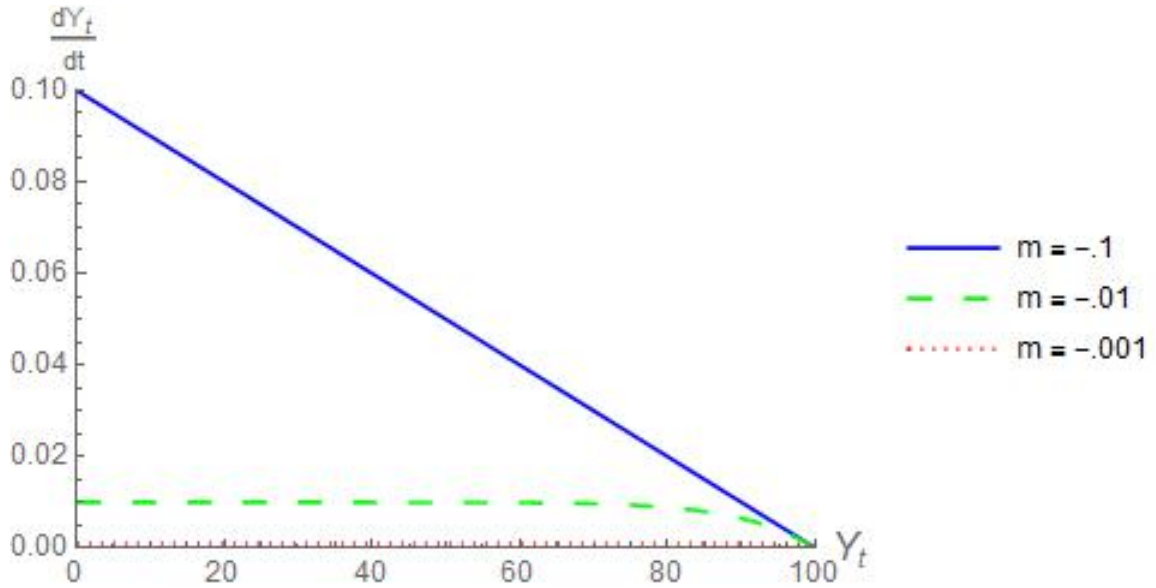


Figure 2.12: Chapman–Richards instantaneous growth rate with $m = -1, -.1, -.01, \lambda = .1, Y_\infty = 100$.

Since the Chapman-Richards equation is of similar form to the generalized

logistic equation, we have the same patterns for parameter adjustments.

2.5 The Weibull Equation

The closed form solution of the Weibull equation is [13]:

$$Y_t = Y_\infty - \alpha e^{-\beta t^\gamma},\ t \geq 0 \tag{2.23}$$

Its first and second derivatives are

$$\frac{dY_t}{dt} = \beta\gamma t^{\gamma-1}(Y_\infty - Y_t) \tag{2.24}$$

$$\frac{d^2Y_t}{dt^2} = \beta\gamma t^{\gamma-1}\left[(\gamma-1)t^{-1}(Y_\infty - Y_t) - \frac{dY_t}{dt}\right] \tag{2.25}$$

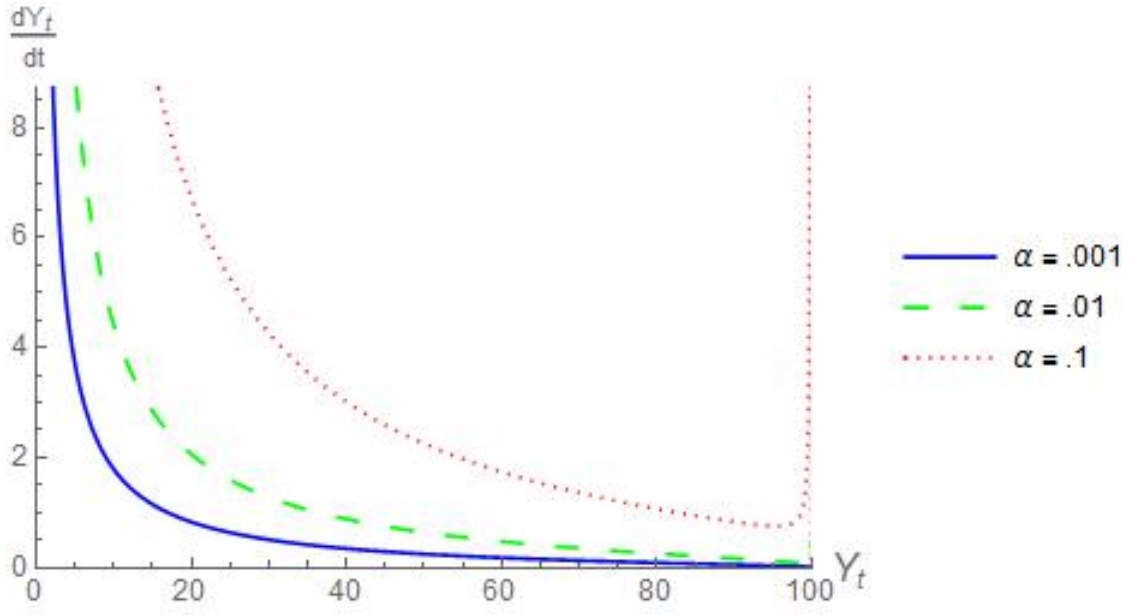


Figure 2.13: Weibull phase diagram with parameters $\alpha = .1, .01, .001, \beta = 7, \gamma = 1/5, Y_\infty = 100$.

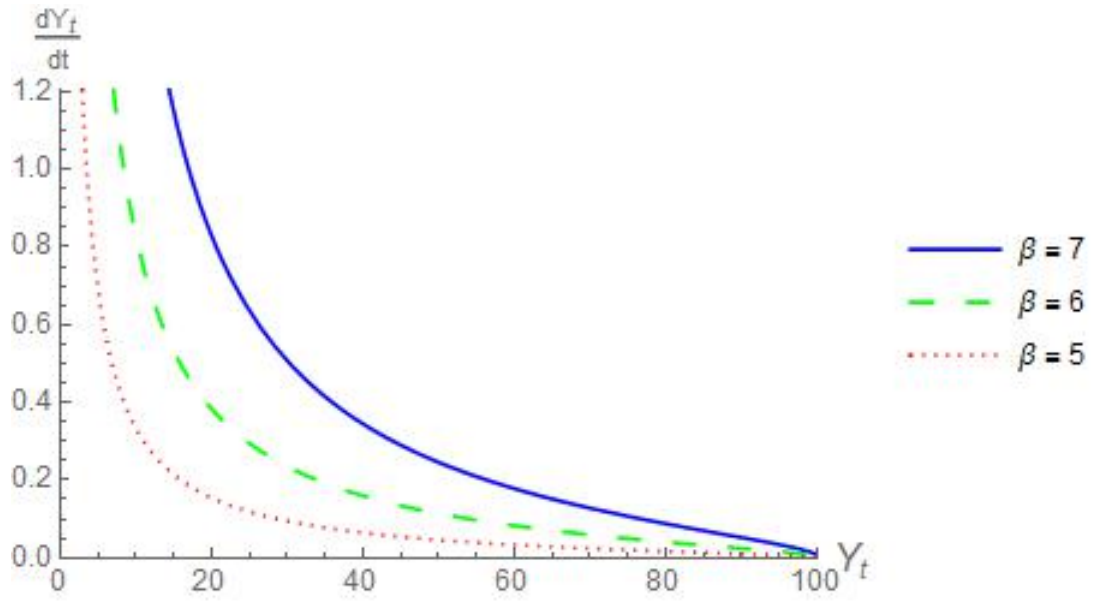


Figure 2.14: Weibull phase diagram with parameters $\alpha = .001, \beta = 5, 6, 7, \gamma = 1/5, Y_\infty = 100$.

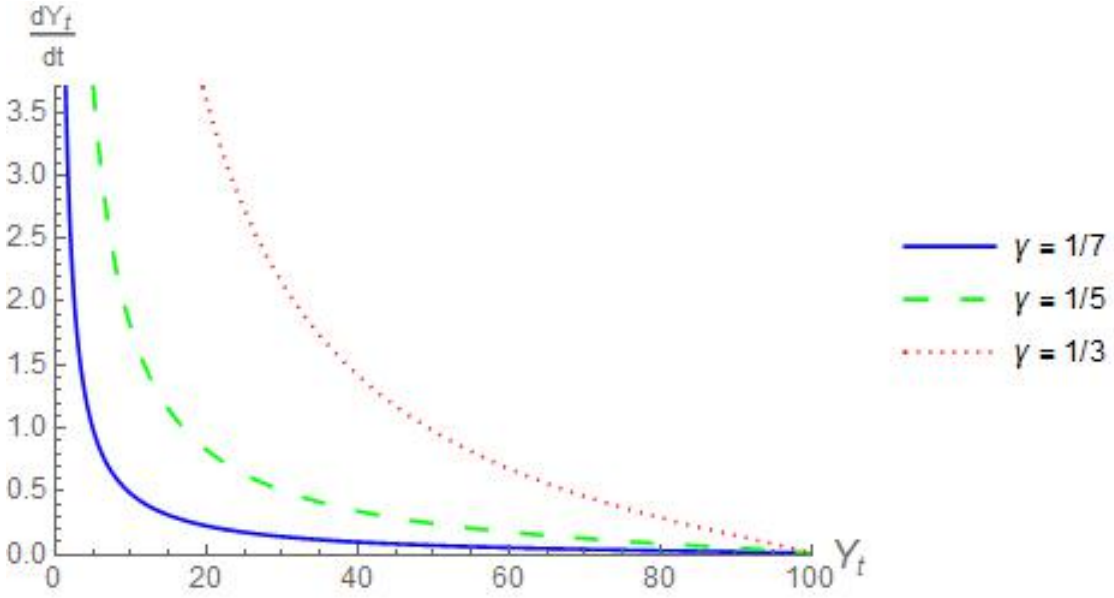


Figure 2.15: Weibull phase diagram with parameters $\alpha = .001, \beta = 7, \gamma = 1/3, 1/5, 1/7, Y_\infty = 100$.

To calculate the inflection point:

$$0 = \beta\gamma t^{\gamma-1}\left[(\gamma-1)t^{-1}(Y_\infty - Y_t) - \frac{dY_t}{dt}\right]$$

$$0 = (\gamma-1)t^{-1}(Y_\infty - Y_t) - \frac{dY_t}{dt}$$

$$\frac{dY_t}{dt} = (\gamma-1)t^{-1}(Y_\infty - Y_t)$$

$$\beta\gamma t^{\gamma-1}(Y_\infty - Y_t) = (\gamma-1)t^{-1}(Y_\infty - Y_t)$$

$$t^\gamma = \frac{\gamma-1}{\beta\gamma}$$

$$t_{inflection} = \left(\frac{\gamma-1}{\beta\gamma}\right)^{1/\gamma}$$

By direct substitution of $t_{inflection}$ into the closed form solution (2.23), we get:

$$Y_{inflection} = Y_\infty - \alpha e^{-(\gamma-1)/\gamma} \tag{2.26}$$

So the inflection point for this curve is:

$$(t_{inflection}, Y_{inflection}) = \left(\left(\frac{\gamma-1}{\beta\gamma}\right)^{1/\gamma}, Y_\infty - \alpha e^{-(\gamma-1)/\gamma}\right). \tag{2.27}$$

The instantaneous growth rate derived from equation (2.24) is given by

$$\frac{\frac{dY_t}{dt}}{Y_t} = \beta\gamma t^{\gamma-1}\left(\frac{Y_\infty}{Y_t} - 1\right) \tag{2.28}$$

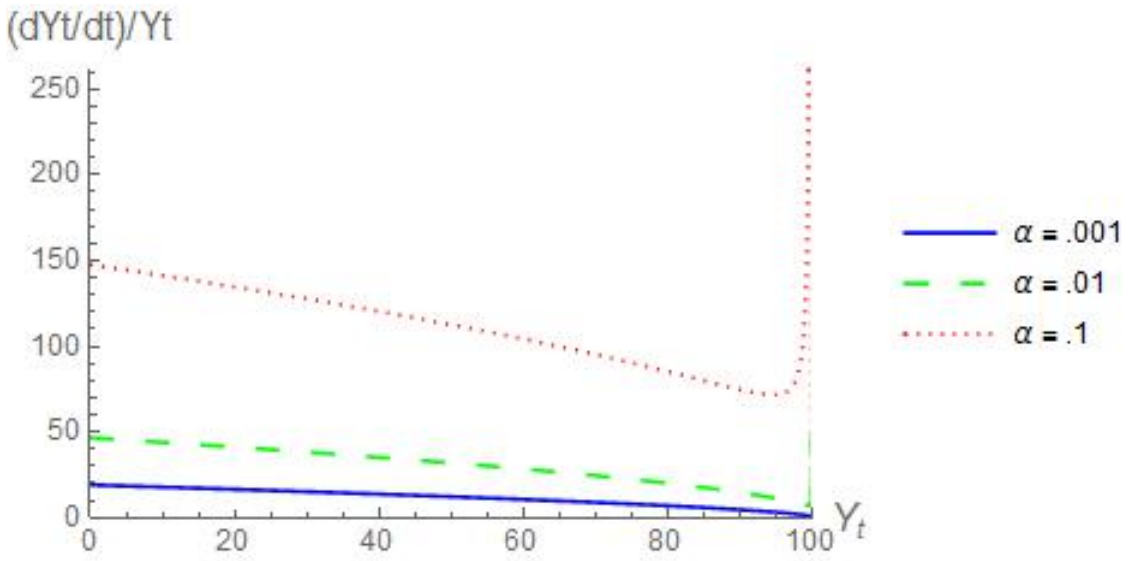


Figure 2.16: Weibull instantaneous growth rate with parameters $\alpha = .1, .01, .001, \beta = 7, \gamma = 1/5, Y_\infty = 100$.

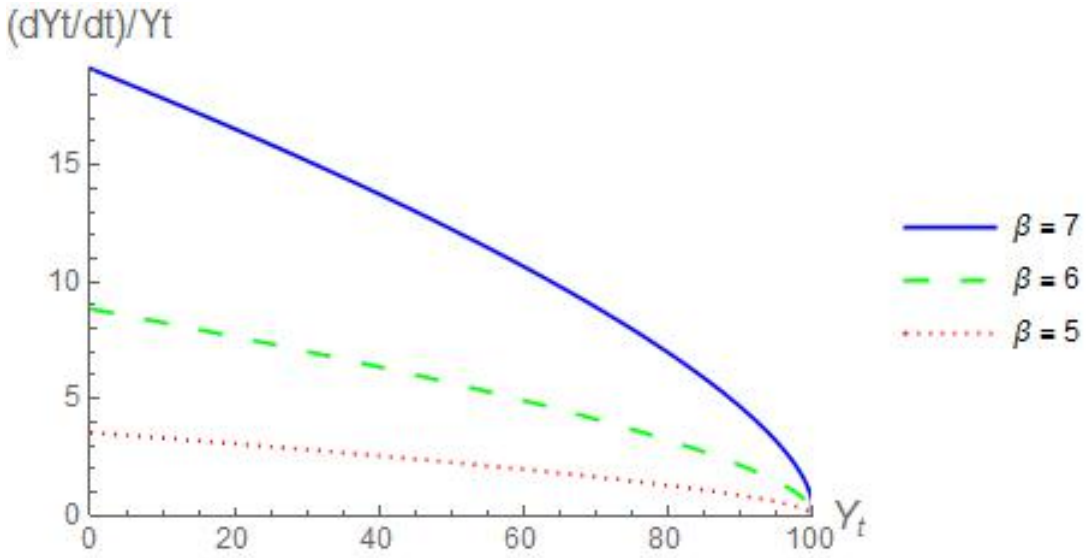


Figure 2.17: Weibull instantaneous growth rate with parameters $\alpha = .001, \beta = 5, 6, 7, \gamma = 1/5, Y_\infty = 100$.

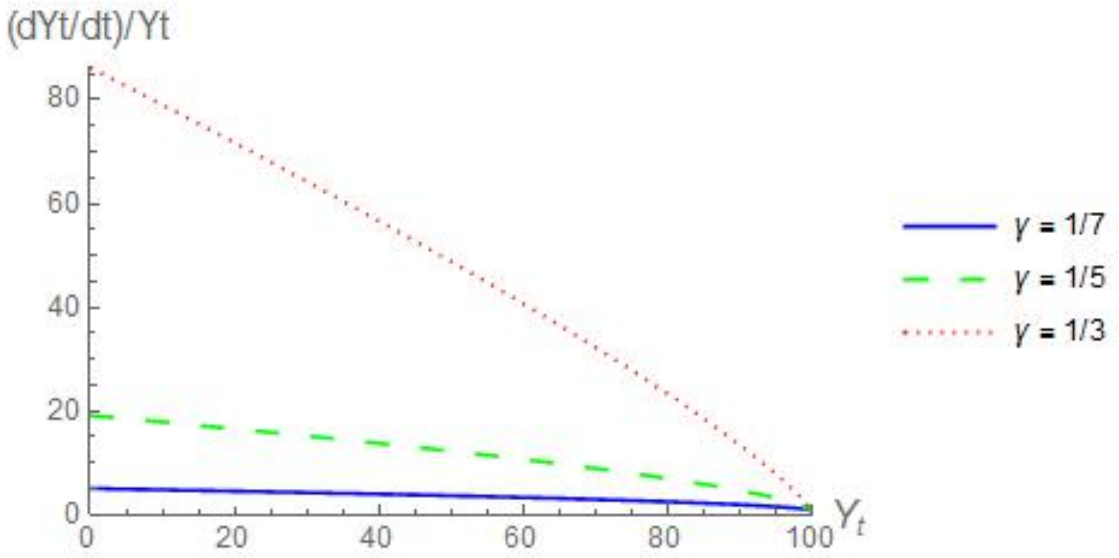


Figure 2.18: Weibull instantaneous growth rate with parameters $\alpha = .001, \beta = 7, \gamma = 1/3, 1/5, 1/7, Y_\infty = 100$.

CHAPTER 3

Filtering Noise

Before attempting to fit our models into the raw data, we need to smooth out the noise from the data to reduce forecasting error.

3.1 Moving Average Filtering

The simplest smoothing function is the moving average [9]:

$$F_{t+1} = \frac{1}{k} \sum_{i=t-k+1}^{t} Y_i \tag{3.1}$$

where Y_i is raw data point, F_{t+1} is the smoothed data, and k is the number of previous data points to average.

The function takes the arithmetic average of its previous k data points. If we assume time is initialized at $t = 0$, the output of the moving average function starts at $t = k$. The output needs a minimum of k input points. This function places equal weight for each previous k data point.

3.2 Single Exponential Smoothing

The single exponential smoothing function [9] is:

$$F_{t+1} = F_t + \alpha(Y_t - F_t), \tag{3.2}$$

where α is constant such that $0 < \alpha < 1$, F_t is the smoothed data, and Y_t is the raw data. The difference $Y_t - F_t$ can be regarded as the forecast error for time period t. In this interpretation, the new forecast F_{t+1} is the previous forecast F_t plus an adjustment for the error that occurred in the last forecast.

We initialize the smoothing function by either letting $F_1 = Y_2$ or taking the arithmetic average of $k-1$ terms. The constant α applies a weight on the difference between smoothed data point and raw data at a given time point t. An α close to 0 has a small adjustment from the previous forecast error, while an α close to 1 has a large adjustment. Here is an graph that illustrates the single exponential smoothing filter with an arbitrary set of data.

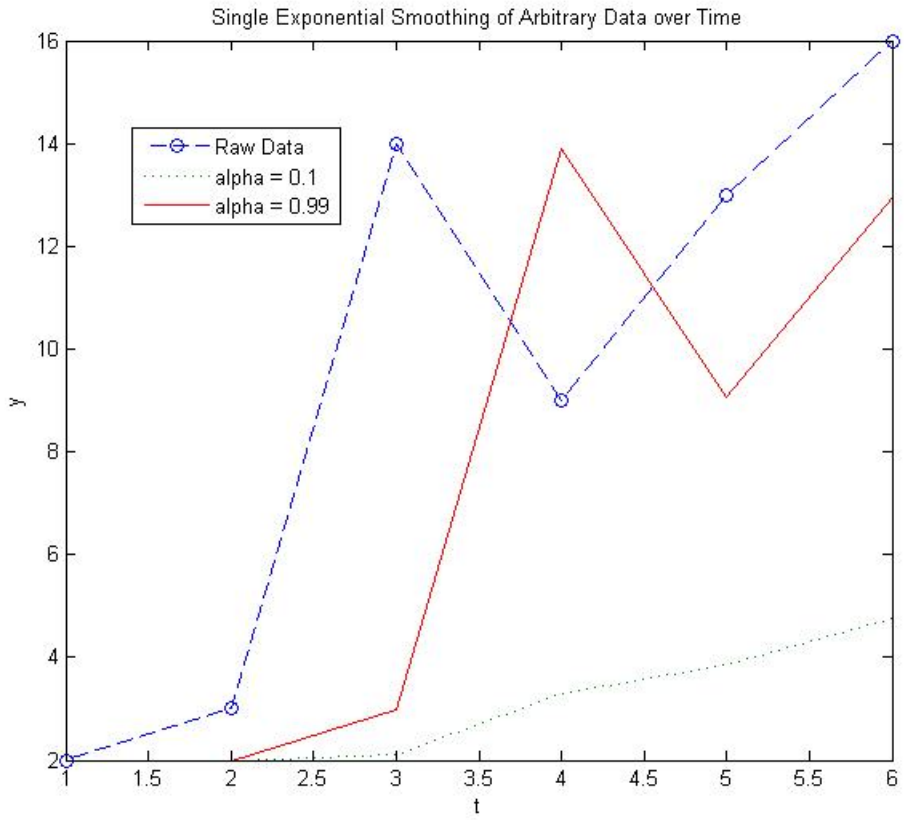


Figure 3.1: Example of single exponential smoothing filter.

Notice for this data, a high α looks almost like a transposition of the raw data, shifted to the right on the x-axis. On the other extreme, the trend line barely increases relative to the shape of the raw data. Also, low α has very low small fluctuations in

slope in comparison to high alpha.

3.3 The Hodrick-Prescott Filter

The Hodrick-Prescott filter [6] is a technique for finding correlations in economic data by separating raw data into a trend function and a cyclic function.

Kim [7] summarizes the Hodrick-Prescott filter as follows.

Suppose a given set of raw data y_t can be decomposed as follows:

$$y_t = \tau_t + c_t,\ t = 1, 2, \ldots, T, \tag{3.3}$$

where τ_t is the trend component and c_t is the cyclical component. The Hodrick-Prescott filter isolates c_t by minimizing the function

$$f(\tau_1, \tau_2, \ldots, \tau_T) = \left[\sum_{t=1}^{T}(y_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1}(\tau_{t+1} - 2\tau_t + \tau_{t-1})^2\right], \tag{3.4}$$

where λ is called the *penalty parameter*. We want to minimize changes in the growth rate, thereby producing a curve with minimal sudden changes in acceleration. This parameter can be estimated by square rooting the quotient of the percent fluctuation of the cyclic component with the percentage growth rate of one quarter. Quarterly data typically assumes $\lambda = 1600$ because Hodrick and Prescott assumes 5% fluctuation for the cyclical component, and 1/8 % growth for a fiscal quarter. When λ approaches 0, the trend component τ_t matches the raw data, and when λ approaches infinity, τ_t becomes linear, or zero acceleration.

The objective function (3.4) shows two summations. The summation on the left is the variance between raw data and the trend component. The right summation is the variance of the acceleration of the trend component.

To minimize f, we set

$$\frac{\partial f}{\partial \tau_1} = \frac{\partial f}{\partial \tau_2} = \ldots = \frac{\partial f}{\partial \tau_T} = 0 \tag{3.5}$$

Note that

$$\frac{\partial f}{\partial \tau_1} = -2(y_1 - \tau_1) + 2\lambda(\tau_3 - 2\tau_2 + \tau_1) = 0$$

This implies

$$\begin{aligned} y_1 &= (1+\lambda)\tau_1 - 2\lambda\tau_2 + \lambda\tau_3 \\ &= \lambda(\tau_1 - 2\tau_2 + \tau_3) + \tau_1 \end{aligned}$$

For τ_2 :

$$\frac{\partial f}{\partial \tau_2} = -2(y_2 - \tau_2) + 2\lambda(\tau_3 - 2\tau_2 + \tau_1)(-2) + 2\lambda(\tau_4 - 2\tau_3 + \tau_2) = 0$$

This implies

$$\begin{aligned} y_2 &= (-2\lambda)\tau_1 + (1 + 4\lambda + \lambda)\tau_2 + (-2\lambda - 2\lambda)\tau_3 + \lambda\tau_4 \\ &= \lambda(-2\tau_1 + 5\tau_2 - 4\tau_3 + \tau_4) + \tau_2 \end{aligned}$$

In general,

$$\begin{aligned} \frac{\partial f}{\partial \tau_k} &= -2(y_k - \tau_k) + 2\lambda(\tau_k - 2\tau_{k-1} + \tau_{k-2}) + 2\lambda(\tau_{k+1} - 2\tau_k + \tau_{k-1})(-2) \\ &\quad + 2\lambda(\tau_{k+2} - 2\tau_{k+1} + \tau_k) = 0 \end{aligned}$$

This implies

$$\begin{aligned} y_k &= \lambda\tau_{k+2} + (-2\lambda - 2\lambda)\tau_{k+1} + (1 + \lambda + 4\lambda + \lambda)\tau_k + (-2\lambda - 2\lambda)\tau_{k-1} + \lambda\tau_{k-2} \\ &= \lambda(\tau_{k+2} - 4\tau_{k+1} + 6\tau_k - 4\tau_{k-1} + \tau_{k-2}) + \tau_k \end{aligned}$$

We can now rewrite the minimization function in matrix notation as:

$$\mathbf{y}_T = (\lambda \mathbf{F} + \mathbf{I}_T)\boldsymbol{\tau}_T \tag{3.6}$$

where $\mathbf{y}_T = (y_1, y_2, \ldots, y_T)^T$ is a $T \times 1$ vector of the raw data, $\mathbf{I}_T$ is $T \times T$ identity matrix , $\boldsymbol{\tau}_T = (\tau_1, \tau_2, \ldots, \tau_T)^T$ is the $T \times 1$ trend component vector and $\mathbf{F}$ is a pentadiagonal symmetric matrix given by

$$\mathbf{F} = \begin{bmatrix} 1 & -2 & 1 & 0 & \ldots & & & & 0 \\ -2 & 5 & -4 & 1 & 0 & \ldots & & & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \ldots & & \vdots \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & \ldots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ & & \ldots & 0 & 1 & -4 & 6 & -4 & 1 \\ & & & \ldots & 0 & 1 & -4 & 5 & -2 \\ & & & & \ldots & 0 & 1 & -2 & 1 \end{bmatrix}.$$

From (3.6), the trend component vector can be isolated

$$\boldsymbol{\tau}_T = (\lambda \mathbf{F} + \mathbf{I}_T)^{-1}\mathbf{y}_T. \tag{3.7}$$

The equation (3.7) has some computational advantages. The only unknown parameter needed to smooth raw data is a single real number λ. Since we are smoothing daily data, Ravn and Uhlig [16] shows that $\lambda = 1600\left(\frac{365}{4}\right)^4 = 110930628906.250$. The pentadiagonal symmetric matrix $\mathbf{F}$ can be easily inverted with fewer flops. The Hodrick-Prescott filter was implemented with MATLAB code given in the appendix [5].

3.4 Comparison of Various Smoothing techniques

To see which smoothing technique is best for sigmoidal curve fitting, this paper will use the mean square error as a metric for the best fitting technique. The

following data set is the daily closing price of Chipotle's stock price from its initial public offering date, January 26th, 2006, to June 17th, 2016 [30].

The equation for the mean square error [26] is:

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(S_t - R_t)^2, \tag{3.8}$$

where S_t it the smoothed data and R_t is the raw data.

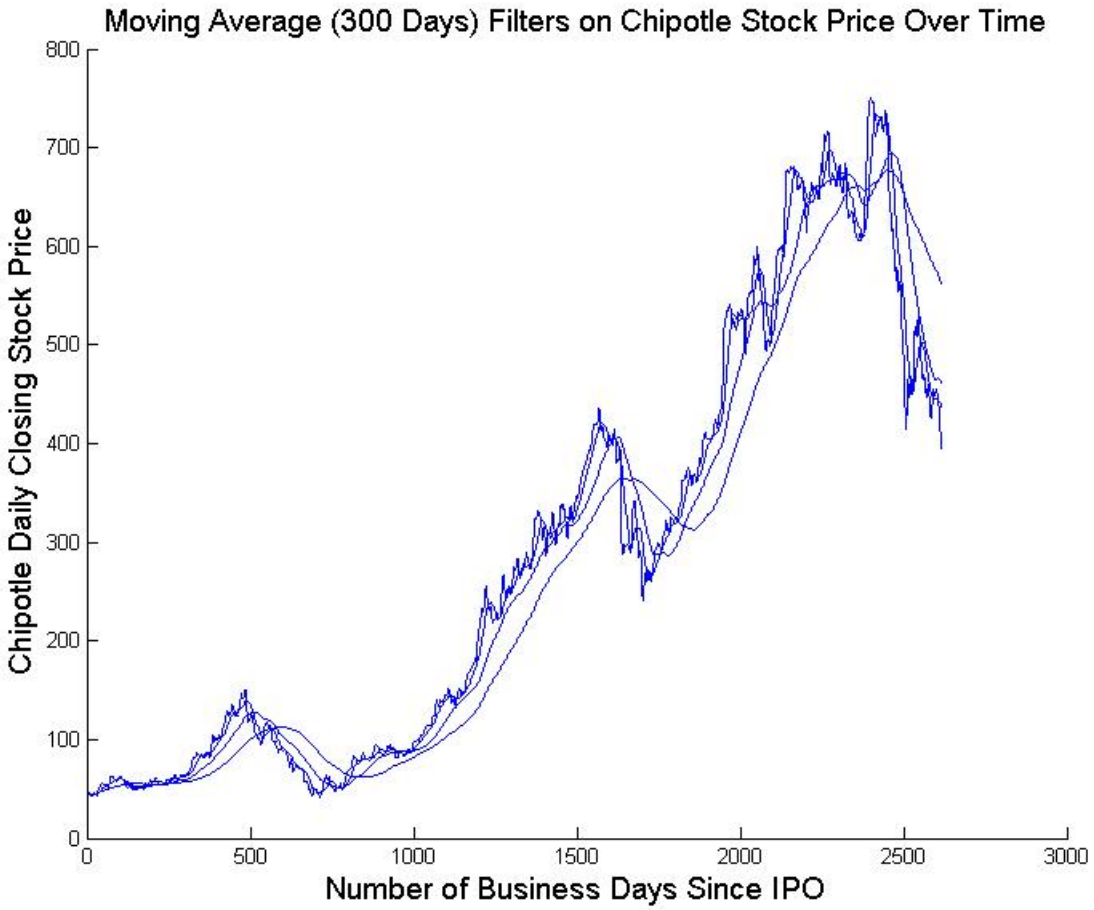


Figure 3.2: Plot of moving average filter with various k days.

Table 3.1: MSE of moving average filtering

k	MSE
5	57.796012692925402
30	480.525902866916
100	1760.21885253903
300	4227.2094117137203

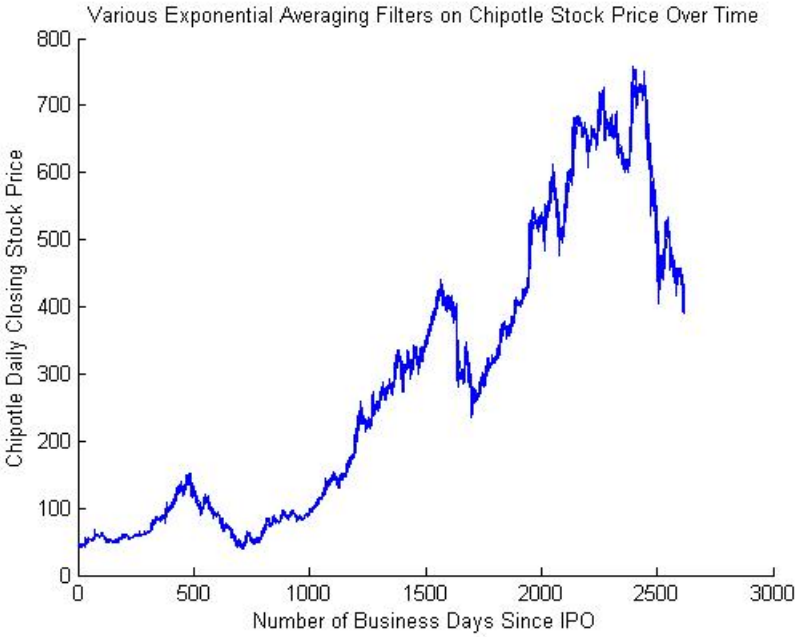


Figure 3.3: Plot of single exponential filter with various α.

Table 3.2: MSE of single exponential filtering

α	MSE
0.1	2.60E+02
0.2	1.33E+02
0.3	93.3976601
0.4	74.56602442
0.5	63.73344863
0.6	56.91617951
0.7	56.91617951
0.8	49.67360402
0.9	48.08265613

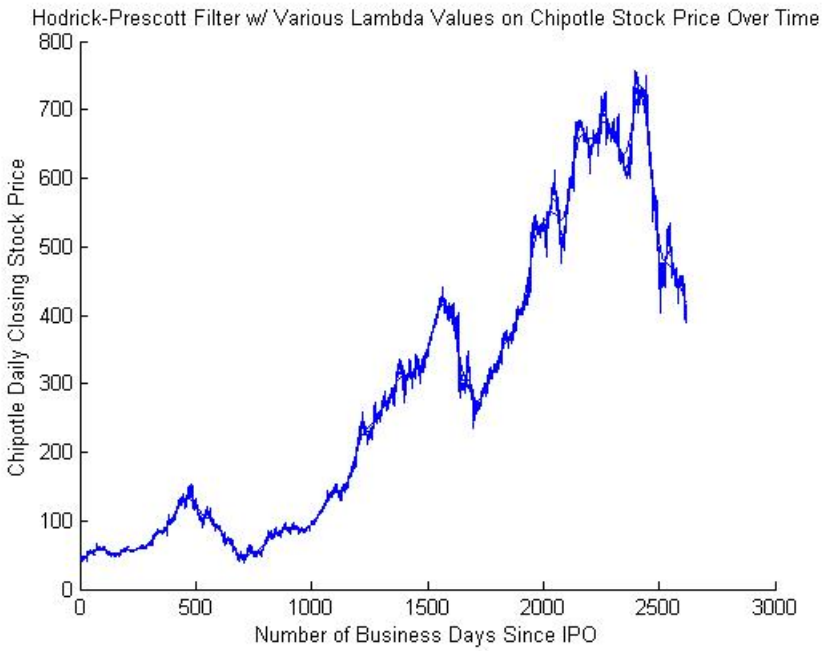


Figure 3.4: Plot of Hodrick-Prescott filter with various λ.

Table 3.3: MSE of Hodrick-Prescott filtering

λ	MSE
160	44.51601627
800	63.33548743
1600	74.14633128
3200	88.30237894
16000	1.38E+02
160000	2.52E+02

The MSE can only measure the extent to which the smoothed data deviates from the raw data. After we explore fitting algorithms used in this paper, the MSE will reveal how well sigmoidal curves fit with the raw data and how well sigmoidal curves forecast data.

For moving average filtering, the choice of using 5, 30, 100, and 300 days is used to approximate the average of a fiscal week, fiscal month, fiscal quarter, and

fiscal year, respectively. The deviation in moving average filtering increases as the number of days averaged increases. For single exponential smoothing, the smoothing deviation decreases as α increases. For the Hodrick-Prescott filter, the MSE increases as λ increases.

CHAPTER 4

Fitting Data and The Levenberg-Marquardt Algorithm

This chapter starts with the discussion of polynomial interpolation as one of the basic techniques for curve fitting. Next we look into nonlinear least squares problems that arise in the context of fitting a more general parameterized function to a set of data points by minimizing the sum of the squares of the errors between the data points and the function. The Levenberg-Marquardt algorithm is a standard technique for solving nonlinear least squares problems. We present the derivation of the Levenberg-Marquardt algorithm along with its convergence theorem. A computational example is also presented to illustrate the algorithm.

4.1 Polynomial Interpolation

One of the most common and simplest ways to fit data is by fitting polynomial functions into a given data set. Given a data set $\{(x_i, y_i),\ i = 1, 2 \dots n\}$, we aim to find a k-th order polynomial, where $k < n$:

$$y = a_0 + a_1 x + \cdots + a_k x^k. \tag{4.1}$$

The error r, also called *residual*, is defined to be the difference between the fitted function and the data points. The sum of the square error can be written as

$$R(a_0, a_1, \dots, a_k) = r^2 = \sum_{i=1}^{n} [y_i - (a_0 + a_1 x_i + \dots + a_k x_i^k)]^2. \tag{4.2}$$

Note that R is a function of $k+1$ variables $a_0, a_1, \ldots, a_k$. To minimize R, we take the partial derivative with respect to each a_k and set it equal to zero:

$$\begin{aligned}
\frac{\partial R}{\partial a_0} &= -2\sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)] = 0 \\
\frac{\partial R}{\partial a_1} &= -2\sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)]x_i = 0 \\
&\vdots \\
\frac{\partial R}{\partial a_k} &= -2\sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)]x_i^k = 0
\end{aligned}$$

By dividing both sides by the constants and distributing terms we get:

$$\begin{aligned}
\frac{\partial R}{\partial a_0} &= \sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)] = 0 \\
\frac{\partial R}{\partial a_1} &= \sum_{i=1}^{n}[x_i y_i - (a_0 x_i + a_1 x_i^2 + \ldots + a_k x_i^{k+1})] = 0 \\
&\vdots \\
\frac{\partial R}{\partial a_k} &= \sum_{i=1}^{n}[x_i^k y_i - (a_0 x_i^k + a_1 x_i^{k+1} + \ldots + a_k x_i^{2k})] = 0.
\end{aligned}$$

We now separate each summation term and move all terms containing y to one side, we get:

$$\begin{aligned} a_0 n + a_1 \sum_{i=1}^{n} x_i + \ldots + a_k \sum_{i=1}^{n} x_i^k &= \sum_{i=1}^{n} y_i \\ a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + \ldots + a_k \sum_{i=1}^{n} x_i^{k+1} &= \sum_{i=1}^{n} x_i y_i \\ &\vdots \\ a_0 \sum_{i=1}^{n} x_i^k + a_1 \sum_{i=1}^{n} x_i^{k+1} + \ldots + a_k \sum_{i=1}^{n} x_i^{2k} &= \sum_{i=1}^{n} x_i^k y_i \end{aligned} \tag{4.3}$$

The above system of equations is called the *normal equations* and can be written in the following matrix form

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \ldots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \ldots & \sum_{i=1}^{n} x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \ldots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^k y_i \end{bmatrix}. \tag{4.4}$$

A *Vandermonde matrix* is a matrix with the terms of a geometric progression in each row. The matrix

$$V = \begin{bmatrix} 1 & x_1 & \ldots & x_1^k \\ 1 & x_2 & \ldots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^k \end{bmatrix} \tag{4.5}$$

is a Vandermonde matrix. Note that (4.4) can be decomposed in terms of Vandermonde matrix V as shown below:

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \ldots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & \ldots & x_1^k \\ 1 & x_2 & \ldots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \ldots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \tag{4.6}$$

that is,

$$V^T V \mathbf{a} = V^T \mathbf{y}, \tag{4.7}$$

where $\mathbf{a} = [a_0, a_1, \ldots, a_k]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$. Therefore, the coefficients $\mathbf{a}$ can be written as

$$\mathbf{a} = (V^T V)^{-1} V^T \mathbf{y}. \tag{4.8}$$

Note that the dimension of V is $n \times (k+1)$ and it easily becomes very large as the number of data points is large. Solving for coefficients $\mathbf{a}$ from the system (4.7) takes $O((k+1)^3)$ using Gaussian elimination. Moreover, the behavior of polynomial functions as t increases approaches $\pm\infty$, which is impractical for modeling a carrying capacity. In the next section, we will look at the least square problems that arise from fitting parameterized functions, such as the sigmoidal curves, to a set of data points.

4.2 Nonlinear Least Square Problems

Given a set of data points $\{(t_1, y_1), (t_2, y_2), \ldots, (t_m, y_m)\}$, the nonlinear least square problem is a problem of finding a function $p(t, x_1, x_2, \ldots, x_n)$ of n parameters $x_1, x_2, \ldots, x_n$ that best fits the data. We want to find the parameter values $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ through iterative improvement that minimizes the sum of the squares of the errors between the data points and the function. The problem can be formulated as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \tag{4.9}$$

where

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(\mathbf{x}), \tag{4.10}$$

r_j are residuals, or more specifically $r_j = |\text{raw data} - \text{fitted function}| = |y_j - p(t_j, \mathbf{x})|, j = 1, \ldots, m$. We assume that $m \geq n$.

The minimization function can be rewritten as:

$$f(\mathbf{x}) = \frac{1}{2}||\mathbf{r}(\mathbf{x})||^2 = \frac{1}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x}), \tag{4.11}$$

where $\mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots, r_m(\mathbf{x}))^T$.

Recall that the Jacobian $J(\mathbf{x})$ of $\mathbf{r}$ is the $m \times n$ matrix of the first partial derivatives, that is,

$$J = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \frac{\partial r_m}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix}. \tag{4.12}$$

Recall also

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}(2r_1(\mathbf{x})\frac{\partial r_1}{\partial x_1} + 2r_2(\mathbf{x})\frac{\partial r_2}{\partial x_1} + \cdots + 2r_m(\mathbf{x})\frac{\partial r_m}{\partial x_1}) \\ \vdots \\ \frac{1}{2}(2r_1(\mathbf{x})\frac{\partial r_1}{\partial x_n} + 2r_2(\mathbf{x})\frac{\partial r_2}{\partial x_n} + \cdots + 2r_m(\mathbf{x})\frac{\partial r_m}{\partial x_n}) \end{bmatrix} \\ &= \begin{bmatrix} r_1\frac{\partial r_1}{\partial x_1} + r_2\frac{\partial r_2}{\partial x_1} + \cdots + r_m\frac{\partial r_m}{\partial x_1} \\ \vdots \\ r_1\frac{\partial r_1}{\partial x_n} + r_2\frac{\partial r_2}{\partial x_n} + \cdots + r_m\frac{\partial r_m}{\partial x_n} \end{bmatrix} \end{aligned}$$

We can now rewrite $\nabla f(\mathbf{x})$ as

$$\begin{aligned}
\nabla f(\mathbf{x}) &= \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_2}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_1} \\ \frac{\partial r_1}{\partial x_2} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_1}{\partial x_n} & \frac{\partial r_2}{\partial x_n} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \\
&= J^T \mathbf{r} \\
&= [r_1, r_2, \ldots, r_m] \begin{bmatrix} \nabla r_1 \\ \nabla r_2 \\ \vdots \\ \nabla r_m \end{bmatrix}, \text{ where } \nabla r_j = \begin{bmatrix} \frac{\partial r_j}{\partial x_1} \\ \frac{\partial r_j}{\partial x_2} \\ \vdots \\ \frac{\partial r_j}{\partial x_n} \end{bmatrix} \\
&= r_1 \nabla r_1 + r_2 \nabla r_2 + \cdots + r_m \nabla r_m \\
&= \sum_{j=1}^{m} r_j \nabla r_j.
\end{aligned}$$

The derivatives of f can be expressed in terms of the Jacobian matrix $J(\mathbf{x}) = \left[\frac{\partial r_i}{\partial x_j}\right]$, $1 \leq i \leq m,\ 1 \leq j \leq n$, as follows

$$\nabla f(\mathbf{x}) = \sum_{j=1}^{m} r_j(\mathbf{x}) \nabla r_j(\mathbf{x}) = J(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \tag{4.13}$$

$$\nabla^2 f(\mathbf{x}) = J(\mathbf{x})^T J(\mathbf{x}) + \sum_{j=1}^{m} r_j(\mathbf{x}) \nabla^2 r_j(\mathbf{x}) \tag{4.14}$$

In the vicinity of a solution, $\mathbf{r}(\mathbf{x})$ is usually small, so the summation in the second term of (4.14) is negligible and $J(\mathbf{x})^T J(\mathbf{x})$ can be taken as an approximation for the Hessian:

$$\nabla^2 f(\mathbf{x}) \approx J(\mathbf{x})^T J(\mathbf{x}). \tag{4.15}$$

4.3 Line Search Algorithms

A general procedure of line search algorithms for function minimization is as follows. We start with an initial guess, $\mathbf{x}_0 \in \mathbb{R}^n$, and produce a sequence of points

$\{\mathbf{x}_k\}$ that, under appropriate conditions, will converge to a minimizer $\mathbf{x}^*$. At each iteration k, the next iterate $\mathbf{x}_{k+1}$ is determined from the current iterate $\mathbf{x}_k$ as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{4.16}$$

where $\mathbf{p}_k \in \mathbb{R}^n$ is a suitably chosen direction and α_k is a suitably chosen step size.

In line search algorithms, we first determine the direction $\mathbf{p}_k$, then compute the step size α_k to determine how far we need to move along that direction. The search direction $\mathbf{p}_k$ can be written in the form

$$\mathbf{p}_k = -B_k^{-1} \nabla f_k, \tag{4.17}$$

where $B_k = B(\mathbf{x}_k)$ is an $n \times n$ matrix and $\nabla f_k = \nabla f(\mathbf{x}_k)$ is the gradient of f at the current iterate $\mathbf{x}_k$. There are many choices for $\mathbf{p}_k$, but in most line search algorithms, $\mathbf{p}_k$ is chosen to be a *descent direction*.

Definition: Let $f : \mathbb{R}^n \to \mathbb{R}$. A vector $\mathbf{p} \in \mathbb{R}^n$ is a descent direction for f at $\mathbf{x}$ if $\mathbf{p}^T \nabla f(\mathbf{x}) < 0$.

Using Taylor's theorem one can show that if we move in sufficiently small step along the descent direction $\mathbf{p}$, then the function value is reduced. Moreover, since $\mathbf{p}$ is a descent direction, we also have from (4.17)

$$\mathbf{p}^T \nabla f(\mathbf{x}) < 0 \Leftrightarrow (-B^{-1} \nabla f(\mathbf{x}))^T \nabla f(\mathbf{x}) < 0 \tag{4.18}$$

$$\Leftrightarrow -\nabla f(\mathbf{x})^T B^{-T} \nabla f(\mathbf{x}) < 0 \tag{4.19}$$

$$\Leftrightarrow \nabla f(\mathbf{x})^T B^{-T} \nabla f(\mathbf{x}) > 0 \tag{4.20}$$

which implies that B^{-T} is a positive definite matrix and so is B.

Two commonly used methods in the family of line search algorithms are the gradient descent and Gauss-Newton methods, which will be described next.

4.3.1 Gradient descent method

In gradient descent method, the direction $\mathbf{p}_k$ is chosen to obtain the greatest decrease in f. For any direction $\mathbf{p}$ with $\|\mathbf{p}\| = 1$ we have

$$\nabla f(\mathbf{x})^T \mathbf{p} = \|\nabla f(\mathbf{x})\| \|\mathbf{p}\| \cos\theta, \tag{4.21}$$

where θ is the angle between $\mathbf{p}$ and $\nabla f(\mathbf{x})$. Since $-1 \leq \cos\theta \leq 1$, this implies that

$$-\|\nabla f(\mathbf{x})\| \leq \nabla f(\mathbf{x})^T \mathbf{p} \leq \|\nabla f(\mathbf{x})\| \tag{4.22}$$

and hence the greatest decrease of f occurs when

$$\nabla f(\mathbf{x})^T \mathbf{p} = -\|\nabla f(\mathbf{x})\| \tag{4.23}$$

that is,

$$\mathbf{p} = \frac{-\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}. \tag{4.24}$$

This direction $\mathbf{p}$ is known as the *steepest descent direction*. In the form of equation (4.17), the matrix $B = I$, the $n \times n$ identity matrix.

In spite of its simplicity, slow convergence of gradient descent method is one of its major disadvantages, especially for functions with long and narrow valley structures.

4.3.2 The Gauss-Newton algorithm

In Gauss-Newton algorithm, the sum of the square errors is reduced by assuming that the objective function f is locally quadratic and finding the minimum of

the quadratic approximation.

Let $m_k(\mathbf{p}_k)$ be the quadratic approximation to $f(\mathbf{x}_k + \mathbf{p}_k)$ at the point $\mathbf{x}_k$. From Taylor's theorem we have

$$m_k(\mathbf{p}_k) = f(\mathbf{x}_k) + \mathbf{p}_k^T \nabla f_k + \frac{1}{2}\mathbf{p}_k^T \nabla^2 f_k \mathbf{p}_k. \tag{4.25}$$

We seek to find $\mathbf{p}_k$ that minimizes m_k. Taking the derivative of (4.25) with respect to $\mathbf{p}_k$ and setting it equal to 0, we obtain

$$\nabla m_k(\mathbf{p}_k) = \nabla f_k + \nabla^2 f_k \mathbf{p}_k = 0, \tag{4.26}$$

which gives us the Newton's direction

$$\mathbf{p}_k = -(\nabla^2 f_k)^{-1} \nabla f_k. \tag{4.27}$$

Gauss-Newton method takes advantage of the special structure of the least square problems. Rather than using the complete second-order Hessian matrix for the quadratic model, the Gauss-Newton method uses an approximation (4.15). Hence, the search direction for Gauss–Newton method is given by:

$$\mathbf{p}_k = -(J_k^T J_k)^{-1} \nabla f_k, \tag{4.28}$$

where $J_k = J(\mathbf{x}_k)$. In the form of equation (4.17), the matrix $B_k = J_k^T J_k$.

4.4 Trust-Region Methods (TRM)

Another approach for solving minimization problem is by using the trust region methods. Line search methods calculate a direction towards the minimizer, then figure out the appropriate step size. Trust region methods take the opposite approach. The

trust region algorithm defines a region around an iterate and constructs a model function that approximates the objective function in that region. The algorithm finds the minimizer of the model function and then takes an iterative step.

In other words, for every k-th iterate, given the model function m_k of a trust region within $\mathbf{p}$ of the current position $\mathbf{x}_k$, the algorithm minimizes $m_k(\mathbf{x}_k + \mathbf{p})$ with respect to $\mathbf{p}$. If sufficient reduction in the function value f is obtained, then m_k is accepted to be a good representation of f in that region. Otherwise the trust region needs to be adjusted accordingly. The goal of the trust region method is to find an approximate trust region radius to arrive at the minimizer $\mathbf{x}^*$.

The algorithm for the trust region method is as follows [12]:

4.4.1 Trust-Region Method Algorithm

Given $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in [0, \frac{1}{4})$

for $k = 0, 1, 2, ...$

(1) Approximate $\mathbf{p}_k$ by solving:

$$\min_{\mathbf{p} \in \mathbb{R}^n} m_k(\mathbf{p}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x}_k) \mathbf{p}, \qquad ||\mathbf{p}|| \leq \Delta_k \tag{4.29}$$

(2) Evaluate:

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{p}_k)}{m_k(\mathbf{0}) - m_k(\mathbf{p}_k)}. \tag{4.30}$$

(3) Determine how to change trust region radius for the next iteration:

if $\rho_k < \frac{1}{4}$

$\Delta_{k+1} = \frac{1}{4} \Delta_k$

else

if $\rho_k > \frac{3}{4}$ and $||\mathbf{p}_k|| = \Delta_k$

$\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$

else

$\Delta_{k+1} = \Delta_k$

(4) Determine the next iterate:

if $\rho_k > \eta$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$

else

$\mathbf{x}_{k+1} = \mathbf{x}_k$.

(End of algorithm)

Letting $g_k = \nabla f(\mathbf{x}_k)$ and using B_k as an approximation to $\nabla^2 f(\mathbf{x}_k)$, we can rewrite (4.29) as

$$m_k(\mathbf{p}) = f_k + g_k^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T B_k \mathbf{p}. \tag{4.31}$$

The following theorems from [12] will be useful in proving the convergence of the Levenberg-Marquardt algorithm in later section.

Theorem 4.1. *Let m be the quadratic function defined by*

$$m(\mathbf{p}) = g^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T B \mathbf{p}, \tag{4.32}$$

where B is any symmetric matrix. Then

(1) *a minimizer of m exists if and only if B is positive semidefinite and g is in the range of B. If B is positive semidefinite, then every $\mathbf{p}$ satisfying $B\mathbf{p} = -g$ is a global minimizer of m.*

(2) *m has a unique minimizer if and only if B is positive definite.*

Proof. For statement (1), assuming B is positive semidefinite and g is in the range of B, we want to show there exists some $\mathbf{p}^*$ that minimizes $m(\mathbf{p})$.

Since g is in the range of B, there exists some $\mathbf{p}^*$ such that $B\mathbf{p}^* = -g$. For any $\mathbf{w} \in \mathbb{R}^n$:

$$\begin{aligned} m(\mathbf{p}^* + \mathbf{w}) &= g^T(\mathbf{p}^* + \mathbf{w}) + \frac{1}{2}(\mathbf{p}^* + \mathbf{w})^T B(\mathbf{p}^* + \mathbf{w}) \\ &= g^T\mathbf{p}^* + g^T\mathbf{w} + \frac{1}{2}(\mathbf{p}^* + \mathbf{w})^T(B\mathbf{p}^* + B\mathbf{w}) \\ &= g^T\mathbf{p}^* + g^T\mathbf{w} + \frac{1}{2}(\mathbf{p}^*)^T B\mathbf{p}^* + \frac{1}{2}(\mathbf{p}^*)^T B\mathbf{w} + \frac{1}{2}\mathbf{w}^T B\mathbf{p}^* + \frac{1}{2}\mathbf{w}^T B\mathbf{w} \end{aligned} \tag{4.33}$$

Since B is symmetric, $B^T = B$, which implies $(\mathbf{p}^*)^T B\mathbf{w} = (B\mathbf{p}^*)^T\mathbf{w}$, and

$$\mathbf{w}^T B\mathbf{p}^* = \mathbf{w}^T(B\mathbf{p}^*) = (B\mathbf{p}^*)^T\mathbf{w} = (\mathbf{p}^*)^T B\mathbf{w} \tag{4.34}$$

Hence, (4.33) becomes:

$$\begin{aligned} m(\mathbf{p}^* + \mathbf{w}) &= (g^T\mathbf{p}^* + \frac{1}{2}(\mathbf{p}^*)^T B\mathbf{p}^*) + g^T\mathbf{w} + (B\mathbf{p}^*)^T\mathbf{w} + \frac{1}{2}\mathbf{w}^T B\mathbf{w} \\ &= m(\mathbf{p}^*) + \frac{1}{2}\mathbf{w}^T B\mathbf{w} \\ &\geq m(\mathbf{p}^*). \end{aligned} \tag{4.35}$$

The last inequality is due to the fact that B is positive semidefinite and thus $\mathbf{w}^T B\mathbf{w} \geq 0$. Hence, $\mathbf{p}^*$ is a minimizer of $m(\mathbf{p})$.

Now assume $\mathbf{p}^*$ is a minimizer of m. It follows that $\nabla m(\mathbf{p}^*) = \mathbf{0}$ and $\nabla^2 m(\mathbf{p}^*)$ is positive semidefinite. From (4.32), note that $\nabla m(\mathbf{p}^*) = B\mathbf{p}^* + g = 0$, which implies that g is in the range of B. Moreover, $\nabla^2 m(\mathbf{p}^*) = B$, so B is positive semidefinite.

For statement (2), assume that B is positive definite. Also assume $\mathbf{p}$ and $\mathbf{q}$ are both minimizers of m. We want to show that $\mathbf{p} = \mathbf{q}$. Using statement (1), since

$\mathbf{p}$ and $\mathbf{q}$ are minimizers,

$$B\mathbf{p} = B\mathbf{q} = -g. \tag{4.36}$$

Since B is positive definite, B is invertible. So this leads to $B^{-1}B\mathbf{p} = B^{-1}B\mathbf{q}$ and therefore $\mathbf{p} = \mathbf{q}$. Therefore, m has a unique minimizer.

Now assume m has a unique minimizer, call it $\mathbf{p}^*$. We want to show that B is positive definite. Suppose B is not positive definite. Then there exists some $\mathbf{w} \neq 0$ such that $\mathbf{w}^T B\mathbf{w} = 0$. From (4.35), $m(\mathbf{p}^* + \mathbf{w}) = m(\mathbf{p}^*)$, indicating that both $\mathbf{p}^*$ and $\mathbf{p}^* + \mathbf{w}$ are minimizers of m, which is a contradiction. Therefore B must be positive definite. □

The following theorem [12] gives the conditions to the solution of trust region problem.

Theorem 4.2. *The vector* $\mathbf{p}^*$ *is a global solution to the trust region problem*

$$\min_{\mathbf{p}\in\mathbb{R}^n} m(\mathbf{p}) = f + g^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T B\mathbf{p}, \qquad \|\mathbf{p}\| \leq \Delta \tag{4.37}$$

if and only if $\mathbf{p}^*$ *is feasible and there exists some* $\lambda \geq 0$ *such that the following conditions are satisfied:*

(1) $(B + \lambda I)\mathbf{p}^* = -g$

(2) $\lambda(\Delta - \|\mathbf{p}^*\|) = 0$

(3) $(B + \lambda I)$ *is positive semidefinite.*

Proof. ($\Leftarrow$) Assume there exists $\lambda \geq 0$ satisfying the three conditions above. We want to show that $\mathbf{p}^*$ is a global minimizer of $m(\mathbf{p})$. By Theorem 4.1, $\mathbf{p}^*$ is the global

minimizer of the quadratic function:

$$\hat{m}(\mathbf{p}) = g^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T(B + \lambda I)\mathbf{p} = m(\mathbf{p}) + \frac{\lambda}{2}\mathbf{p}^T\mathbf{p} \tag{4.38}$$

Since $\hat{m}(\mathbf{p}) \geq \hat{m}(\mathbf{p}^*)$ for any $\mathbf{p}$,

$$m(\mathbf{p}) \geq m(\mathbf{p}^*) + \frac{\lambda}{2}\Big[(\mathbf{p}^*)^T\mathbf{p}^* - \mathbf{p}^T\mathbf{p}\Big] \tag{4.39}$$

From condition (2), $\lambda(\Delta - \|\mathbf{p}^*\|) = 0$ implies

$$\lambda(\Delta - \|\mathbf{p}^*\|)(\Delta + \|\mathbf{p}^*\|) = \lambda(\Delta^2 - \|\mathbf{p}^*\|^2) = \lambda(\Delta^2 - (\mathbf{p}^*)^T\mathbf{p}^*) = 0. \tag{4.40}$$

Thus,

$$\begin{aligned} m(\mathbf{p}) &\geq m(\mathbf{p}^*) + \frac{\lambda}{2}\Big[(\mathbf{p}^*)^T\mathbf{p}^* + \mathbf{p}^T\mathbf{p}\Big] \\ &= m(\mathbf{p}^*) + \frac{\lambda}{2}\Big[(\mathbf{p}^*)^T\mathbf{p}^* - \Delta^2 + \Delta^2 - \mathbf{p}^T\mathbf{p}\Big] \\ &= m(\mathbf{p}^*) + \frac{\lambda}{2}(\Delta^2 - \mathbf{p}^T\mathbf{p}) \end{aligned}$$

Since $\lambda \geq 0, m(\mathbf{p}) \geq m(\mathbf{p}^*)$ for all $\mathbf{p}$ satisfying $\|\mathbf{p}\| \leq \Delta$. Therefore, $\mathbf{p}^*$ is a global minimizer.

($\Rightarrow$) Assume $\mathbf{p}^*$ is a global solution to $m(\mathbf{p})$. We want to show there exists $\lambda \geq 0$ satisfying the three conditions.

Case 1: $\|\mathbf{p}^*\| < \Delta$, that is, $\mathbf{p}^*$ is an unconstrained minimizer of m.

Note that $\nabla m(\mathbf{p}^*) = B\mathbf{p}^* + g = \mathbf{0}$. It follows that $\lambda = 0$ satisfies condition (1). Also $\nabla^2 m(\mathbf{p}^*) = B$, where B is positive semidefinite. The choice $\lambda = 0$ satisfies condition (3). Condition (2) is automatically satisfied when $\lambda = 0$.

Case 2: $\|\mathbf{p}^*\| = \Delta$.

Note that condition (2) is immediately satisfied and the minimizer is within the trust region radius. Moreover, $\mathbf{p}^*$ also solves the constraint problem (4.37). Define the Lagrangian function:

$$\mathcal{L}(\mathbf{p}, \lambda) = m(\mathbf{p}) + \frac{\lambda}{2}(\mathbf{p}^T\mathbf{p} - \Delta^2). \tag{4.41}$$

By the optimality conditions for constrained optimization, there exists some λ for which $\mathbf{p}^*$ is a stationary point. Setting the partial derivative $\nabla_{\mathbf{p}}\mathcal{L}$ of $\mathcal{L}$ with respect to $\mathbf{p}$ to 0, we obtain

$$\nabla_{\mathbf{p}}\mathcal{L}(\mathbf{p}, \lambda) = g + B\mathbf{p} + \lambda\mathbf{p} = 0,$$

and it follows that

$$g + B\mathbf{p}^* + \lambda\mathbf{p}^* = 0 \implies (B + \lambda I)\mathbf{p}^* = -g. \tag{4.42}$$

So condition (1) is satisfied.

Since $\mathbf{p}^*$ is the minimizer of $m(\mathbf{p})$, $m(\mathbf{p}) \geq m(\mathbf{p}^*)$ for any $\mathbf{p}$ with $\mathbf{p}^T\mathbf{p} = (\mathbf{p}^*)^T\mathbf{p}^* = \Delta^2$ and $\mathbf{p} \neq \mathbf{p}^*$. We can write

$$m(\mathbf{p}) \geq m(\mathbf{p}^*) + \frac{\lambda}{2}((\mathbf{p}^*)^T\mathbf{p}^* - \mathbf{p}^T\mathbf{p}).$$

From (4.37),

$$m(\mathbf{p}) - m(\mathbf{p}^*) = (f + g^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T B\mathbf{p}) - (f + g^T\mathbf{p}^* + \frac{1}{2}(\mathbf{p}^*)^T B\mathbf{p}^*) \tag{4.43}$$

and from (4.42),

$$g^T = -(\mathbf{p}^*)^T(B + \lambda I)^T = -(\mathbf{p}^*)^T(B + \lambda I), \tag{4.44}$$

where $(B + \lambda I) = (B + \lambda I)^T$ because it is symmetric. Thus, combining (4.43) and (4.44),

$$
\begin{aligned}
& m(\mathbf{p}) - m(\mathbf{p}^*) \\
&= -(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}\mathbf{p}^T(B + \lambda I)\mathbf{p} + (\mathbf{p}^*)^T(B + \lambda I)(\mathbf{p}^*) - \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)(\mathbf{p}^*) \\
&= -(\mathbf{p}^*)^T B\mathbf{p} - (\mathbf{p}^*)^T\lambda\mathbf{p} + \frac{1}{2}\mathbf{p}^T B\mathbf{p} + \frac{1}{2}\mathbf{p}^T\lambda\mathbf{p} + (\mathbf{p}^*)^T B(\mathbf{p}^*) + (\mathbf{p}^*)^T\lambda(\mathbf{p}^*) \\
&\quad - \frac{1}{2}(\mathbf{p}^*)^T B(\mathbf{p}^*) - \frac{1}{2}(\mathbf{p}^*)^T\lambda(\mathbf{p}^*)
\end{aligned}
$$

Collect terms of B and λ:

$$
\begin{aligned}
&= \frac{1}{2}\mathbf{p}^T B\mathbf{p} - (\mathbf{p}^*)^T B\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T B\mathbf{p}^* + \frac{1}{2}\mathbf{p}^T\lambda\mathbf{p} - (\mathbf{p}^*)^T\lambda\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T\lambda\mathbf{p}^* \\
&= \frac{1}{2}\mathbf{p}^T(B + \lambda I)\mathbf{p} - (\mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p}^* \\
&= \frac{1}{2}\mathbf{p}^T(B + \lambda I)\mathbf{p} - \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p} - \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p}^* \\
&= \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)\mathbf{p} - \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)\mathbf{p}^* \\
&= \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}(\mathbf{p}^*)^T(B + \lambda I)(\mathbf{p}^* - \mathbf{p}) \\
&= \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)\mathbf{p} + \frac{1}{2}(\mathbf{p}^* - \mathbf{p})^T(B + \lambda I)(\mathbf{p}^*) \\
&= \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)\mathbf{p} - \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)(\mathbf{p}^*) \\
&= \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)(\mathbf{p} - \mathbf{p}^*)
\end{aligned}
$$

So,

$$\frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^T(B + \lambda I)(\mathbf{p} - \mathbf{p}^*) \geq 0 \tag{4.45}$$

which implies $(B + \lambda I)$ is positive semidefinite.

All three conditions are satisfied when $\mathbf{p}^*$ is a global minimizer. Now we need to show that $\lambda \geq 0$. We will show this by proof of contradiction. Suppose to the contrary that $\lambda < 0$ and satisfies conditions (1) and (2). Since $\mathbf{p}^*$ minimizes m, by

Theorem 4.1, B is positive semidefinite and $B\mathbf{p}^* = -g$. This implies $\lambda = 0$ in our theorem. This contradicts our supposition. Hence, $\lambda \geq 0$. □

4.5 The Levenberg-Marquardt Algorithm

4.5.1 Motivation behind Levenberg-Marquardt Algorithm

Before delving into the full details of the Levenberg-Marquardt (LM) algorithm, reviewing the motivation behind the algorithm will add clarity to how the algorithm works. The Gauss-Newton method, just like Newton's method, has rapid convergence, but is sensitive to the initial position. On the other hand, the gradient descent method is not sensitive to initial position even though convergence may be slow. Levenberg combines the advantages of gradient descent and Gauss-Newton by taking B_k in equation (4.17) as:

$$B_k = \nabla^2 f_k + \lambda I \tag{4.46}$$

where λ is a damping factor that is adjusted at each iteration.

As in the Gauss-Newton method, the approximation $J_k^T J_k$ is used instead of the actual Hessian $\nabla^2 f_k$, that is,

$$B_k = J_k^T J_k + \lambda I \tag{4.47}$$

and

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J_k^T J_k + \lambda I)^{-1} J_k^T \mathbf{r}_k \tag{4.48}$$

Recall that the Hessian of f is

$$\nabla^2 f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{4.49}$$

Along with the equation (4.48), Levenburg [10] defined the following rule to determine the damping factor λ at each iteration:

(1) Perform one iteration.

(2) Evaluate error at the given iterate.

(3) If error increases, increase λ. If error decreases, decrease λ.

A more precise algorithm for calculating λ in the LM algorithm can be given in trust-region framework and is often called the *trust-region subproblem* [12]:

4.5.2 Trust-Region Subproblem Algorithm

Given λ_1 and k-th time step of the LM algorithm.

for $n = 1, 2, 3, \ldots$

(1) Conduct a Cholesky factorization:

$$J_{k+1}^T J_{k+1} + \lambda_{k_n} I = L_n L_n^T, \tag{4.50}$$

where L_n is an $n \times n$ lower triangular matrix.

(2) Solve $\mathbf{p_n^{(\lambda)}}$ and $\mathbf{q_n^{(\lambda)}}$ in the following equations in sequence:

$$L_n L_n^T \mathbf{p_n^{(\lambda)}} = -J_{k+1}^T \mathbf{r}_{k+1} \tag{4.51}$$

$$L_n \mathbf{q}_\mathbf{n}^{(\lambda)} = \mathbf{p}_\mathbf{n}^{(\lambda)} \tag{4.52}$$

(3) Solve the equation:

$$\lambda_{n+1} = \lambda_n + \left(\frac{\|\mathbf{p}_\mathbf{n}^{(\lambda)}\|}{\|\mathbf{q}_\mathbf{n}^{(\lambda)}\|} \right)^2 \left(\frac{\|\mathbf{p}_\mathbf{n}^{(\lambda)}\| - \Delta_k}{\Delta_k} \right) \tag{4.53}$$

end

Given $\lambda_1 = 1$ as an initial guess. For $k > 1$, we calculate λ using the trust-region subproblem algorithm (Algorithm 4.2). For practical purposes, the algorithm will not be implemented until convergence is obtained because it is computationally expensive. Most will define a finite number of iterations n for the algorithm, or define a tolerance for $|\lambda_{n+1} - \lambda_n|$ and stop the algorithm.

Marquardt [11] noticed that if λ becomes too large, the term $J_k^T J_k$ becomes negligible and the algorithm (4.48) behaves similarly to the gradient descent algorithm. The gradient drop towards the minimum becomes very small for a given path $\mathbf{p}_k$. We want movement along smaller gradients to be larger, and vice versa. Marquardt eliminates this issue by replacing the identity matrix with the diagonal of $J_k^T J_k$ as follows

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[J_k^T J_k + \lambda \, \text{diag}(J_k^T J_k) \right]^{-1} J_k^T \mathbf{r}_k. \tag{4.54}$$

The above equation is the Levenberg-Marquardt algorithm.

4.5.3 Implementation of Levenberg-Marquardt Algorithm

Using the trust region framework, the goal of the LM algorithm is to solve the following minimization problem:

$$\min_{\mathbf{p}} \ \frac{1}{2}\|J_k\mathbf{p} + \mathbf{r}_k\|^2, \text{ subject to } \|\mathbf{p}\| \leq \Delta_k, \tag{4.55}$$

where $\Delta_k > 0$ is the trust-region radius. We define the model function m to be:

$$m_k(\mathbf{p}) = \frac{1}{2}\|\mathbf{r}_k\|^2 + \mathbf{p}^T J_k^T \mathbf{r}_k + \frac{1}{2}\mathbf{p}^T J_k^T J_k \mathbf{p}. \tag{4.56}$$

If the Gauss-Newton direction $\mathbf{p}^{GN}$ obtained from solving $J_k^T J_k \mathbf{p}^{GN} = -J_k^T \mathbf{r}_k$ satisfies the constraint $\|\mathbf{p}^{GN}\| < \Delta$, then $\mathbf{p}^{GN}$ also solves the trust-region subproblem. If this is not the case, then there exists $\lambda > 0$ for which $\mathbf{p}_k^{LM}$ solves

$$(J_k^T J_k + \lambda I)\mathbf{p}_k^{LM} = -J_k^T \mathbf{r}_k = -\nabla f_k, \tag{4.57}$$

and $\|\mathbf{p}^{LM}\| = \Delta$.

The following lemma [12] gives the conditions for the solution of minimization problem (4.55).

Lemma 4.3. *The vector $\mathbf{p}^{LM}$ is the solution to the minimization problem* (4.55) *if and only if $\mathbf{p}^{LM}$ is feasible and there exists $\lambda \geq 0$ such that*

$$(J_k^T J_k + \lambda I)\mathbf{p}^{LM} = -J_k^T \mathbf{r}_k \tag{4.58}$$

$$\lambda(\Delta - \|\mathbf{p}^{LM}\|) = 0 \tag{4.59}$$

Proof. Condition (3) in Theorem 4.2 is satisfied automatically since $J_k^T J_k$ is positive semidefinite and $\lambda \geq 0$. Equations (4.58) and (4.59) follow from condition (1) and condition (2) of Theorem 4.2. □

4.5.4 The Levenberg-Marquardt Algorithm

Given $\hat{\Delta} > 0$, $\Delta_1 \in (0, \hat{\Delta})$, and $\eta \in [0, \frac{1}{4})$

for $k = 1, 2, ...$

(1) If $k = 1$, calculate $\mathbf{p}_k^{GN}$:

$$\mathbf{p_k}^{GN} = -(J_k^T J_k)^{-1} J_k^T \mathbf{r}_k \tag{4.60}$$

if $\mathbf{p}_k^{GN} < \Delta_1$

Use the Gauss–Newton method to obtain convergence

else

Initiate the LM algorithm.

(2) Calculate λ_k using the trust-region subproblem (Algorithm 4.5.2).

(3) Approximate $\mathbf{p}_k$ by:

$$\mathbf{p}_k^{LM} = -(J_k^T J_k + \lambda I)^{-1} J_k^T \mathbf{r}_k \tag{4.61}$$

(4) Evaluate ρ_k using equation (4.56) for $m_k(\mathbf{x})$:

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{p}_k)}{m_k(\mathbf{0}) - m_k(\mathbf{p}_k)} \tag{4.62}$$

(5) Determine how to change trust region radius for the next iteration:

if $\rho_k < \frac{1}{4}$

$\Delta_{k+1} = \frac{1}{4}\Delta_k$

else

if $\rho_k > \frac{3}{4}$ and $||\mathbf{p_k}|| = \Delta_k$

$$\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$$

else

$$\Delta_{k+1} = \Delta_k$$

(6) Determine if after the step direction $\mathbf{p}_k$, ρ_k is small enough to reach an acceptable tolerance η.

if $\rho_k > \eta$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$$

else

$$\mathbf{x}_{k+1} = \mathbf{x}_k.$$

4.5.5 Convergence of The Levenberg-Marquardt Algorithm

Before proving the convergence of the LM algorithm, we have to prove the convergence of the trust region algorithm.

Theorem 4.4. *Let $\eta \in (0, \frac{1}{4})$ in the trust region algorithm (Algorithm 4.4.1). Suppose that $\|B_k\| \leq \beta$ for some constant β. Let g be bounded below on the set level set S defined by:*

$$S(R_0) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{y}\| < R_0, \text{ for some } \mathbf{y} \in S\}, \tag{4.63}$$

where $R_0 > 0$. Let g be a Lipschitz continuous function in $S(R_0)$ with Lipschitz constant β_1, that is $g \in LC_{\beta_1}(S(R_0))$. Suppose all approximate solution $\mathbf{p}_k$ in trust-region algorithm satisfies

$$m_k(\mathbf{0}) - m_k(\mathbf{p}) \geq c_1 ||g_k|| \min\left(\Delta_k, \frac{||g_k||}{B_k}\right) \tag{4.64}$$

and $||\mathbf{p}_k|| \leq \gamma\Delta_k$ for some constant $\gamma \geq 0$, $c_1 > 0$. Then $\{g_k\} \to 0$.

Proof. We consider a particular positive index m such that $g(\mathbf{x}_m) \neq 0$. Since $g \in LC_{\beta_1}(S)$, we have:

$$||g(\mathbf{x}) - g_(\mathbf{x}_m)|| \leq \beta_1 ||\mathbf{x} - \mathbf{x}_m||, \ \forall \mathbf{x}, \mathbf{x}_m \in S(R_0). \tag{4.65}$$

We define scalars $\epsilon = \frac{1}{2}||g_m||$ and $R = \min\left(\frac{\epsilon}{\beta_1}, R_0\right)$. Notice the R-ball around $\mathbf{x}_m$

$$B(\mathbf{x}_m, R) = \{\mathbf{x} \mid ||\mathbf{x} - \mathbf{x}_m|| \leq R\} \tag{4.66}$$

is contained in $S(R_0)$, so Lipschitz continuity of g holds inside $B(\mathbf{x}_m, R)$, that is,

$$\|g(\mathbf{x}) - g(\mathbf{y})\| \leq \beta_1 \|\mathbf{x} - \mathbf{y}\|, \ \forall \mathbf{x}, \mathbf{y} \in B(\mathbf{x}_m, R).$$

In particular,

$$\begin{aligned} \|g(\mathbf{x}) - g(\mathbf{x}_m)\| &\leq \beta_1 \|\mathbf{x} - \mathbf{x}_m\| \\ &\leq \beta_1 R \leq \beta_1(\epsilon/\beta_1) = \epsilon = \frac{1}{2}\|g(\mathbf{x}_m)\|. \end{aligned}$$

From the triangle inequality

$$||g(\mathbf{x}_m)|| - ||g(\mathbf{x})|| \leq ||g(\mathbf{x}_m) - g(\mathbf{x})|| \leq \frac{1}{2}\|g(\mathbf{x}_m)\| \tag{4.67}$$

which implies

$$||g(\mathbf{x})|| \geq \frac{1}{2}\|g(\mathbf{x}_m)\| = \epsilon. \tag{4.68}$$

Let $\{\mathbf{x}_k\}$ be a sequence generated by trust-region algorithm. If $\{\mathbf{x}_k\}_{k \geq m} \subset B(\mathbf{x}_m, R)$, then $\|g(\mathbf{x}_k)\| \geq \epsilon$ for all $k \geq m$. Hence, $\{g(\mathbf{x}_k)\} \nrightarrow 0$. Therefore, there must exist some index $l \geq m$ such that $\{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots\}$ lie outside the ball $B(\mathbf{x}_m, R)$, that is, $\mathbf{x}_{l+1}$ is the first iterate that escapes $B(\mathbf{x}_m, R)$. Note that $\|g(\mathbf{x}_k)\| \geq \epsilon$ for

$k = m, m+1, \ldots, l$. Thus,

$$f(\mathbf{x}_m) - f(\mathbf{x}_{l+1}) = f(\mathbf{x}_m) - f(\mathbf{x}_{m+1}) + f(\mathbf{x}_{m+1}) - \ldots - f(\mathbf{x}_{l+1}) \tag{4.69}$$

$$= \sum_{k=m}^{l} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}.) \tag{4.70}$$

If $\mathbf{x} = \mathbf{x}_{k+1}$, then $f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) = 0$. If $\mathbf{x} \neq \mathbf{x}_{k+1}$, then $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$ for some $\mathbf{p}_k \neq \mathbf{0}$ and this happens when $\rho_k > \eta$, that is,

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})}{m_k(\mathbf{0}) - m_k(\mathbf{p}_k)} > \eta$$

$$\Rightarrow f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) > \eta(m_k(\mathbf{0}) - m_k(\mathbf{p}_k))$$

From (4.70), we have

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{l+1}) &\geq \sum_{k=m, \mathbf{x}_k \neq \mathbf{x}_{k+1}}^{l} \eta(m_k(\mathbf{0}) - m_k(\mathbf{p}_k)) \\ &\geq \sum_{k=m, \mathbf{x}_k \neq \mathbf{x}_{k+1}}^{l} \eta c_1 \|g_k\| \min\left(\Delta_k, \frac{\|g_k\|}{B_k}\right) \text{ (by assumption)} \\ &\geq \sum_{k=m, \mathbf{x}_k \neq \mathbf{x}_{k+1}}^{l} \eta c_1 \epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta}\right). \end{aligned}$$

The last inequality comes from the fact that $\|g_k\| \geq \epsilon$ for all $k \geq m$ and $\|B_k\| \leq \beta$.

We consider two cases:

Case 1: If $\Delta_k > \epsilon/\beta$, then

$$f(\mathbf{x}_m) - f(\mathbf{x}_{l+1}) \geq \eta c_1 \epsilon \frac{\epsilon}{\beta}. \tag{4.71}$$

Case 2: If $\Delta_k \leq \epsilon/\beta$ for $k = m, m+1, \ldots, l$, then

$$f(\mathbf{x}_m) - f(\mathbf{x}_{l+1}) \geq \eta c_1 \epsilon \sum_{k=m, \mathbf{x}_k \neq \mathbf{x}_{k+1}}^{l} \Delta_k \tag{4.72}$$

$$\geq \eta c_1 \epsilon R \tag{4.73}$$

$$= \eta c_1 \epsilon \min\left(\frac{\epsilon}{\beta}, R_0\right). \tag{4.74}$$

Since $\{f(\mathbf{x}_k)\}$ is decreasing and bounded below, $\{f(\mathbf{x}_k)\} \to f(\mathbf{x}^*)$ and $f(\mathbf{x}^*) > -\infty$. Hence, combining both cases we obtain

$$\begin{aligned} f(\mathbf{x}_m) - f(\mathbf{x}^*) &\geq f(\mathbf{x}_m) - f(\mathbf{x}_{l+1}) \text{ (since } f(\mathbf{x}^*) \leq f(\mathbf{x}_{l+1})) \\ &\geq \eta c_1 \epsilon \min\Big(\frac{\epsilon}{\beta}, \frac{\epsilon}{\beta_1}, R_0\Big) \\ &= \frac{1}{2}\eta c_1 \|g(\mathbf{x}_m)\| \min\Big(\frac{\|g(\mathbf{x}_m)\|}{2\beta}, \frac{\|g(\mathbf{x}_m)\|}{2\beta_1}, R_0\Big). \end{aligned}$$

But as $m \to \infty$, $f(\mathbf{x}_m) - f(\mathbf{x}^*) \to 0$, and this forces $\|g(\mathbf{x}_m)\| \to 0$ as well. □

Now we use this theorem to show that the Levenberg-Marquardt algorithm converges [12].

Theorem 4.5. *Let $\eta \in (0, \frac{1}{4})$ in the trust region algorithm. Suppose the set level L as defined by* (4.63) *is bounded and the residual functions $\mathbf{r}_j$, where $j = 1, \ldots, m$ are Lipschitz continuous and differentiable in neighborhood N of L. Assume that for each k, the approximate solution for $\mathbf{p}_k$ in* 4.55 *satisfies:*

$$m_k(\mathbf{0}) - m_k(\mathbf{p}_k) \geq c_1 ||J_k^T \mathbf{r}_k|| \min\Big(\Delta_k, \frac{||J_k^T \mathbf{r}_k||}{||J_k^T J_k||}\Big) \tag{4.75}$$

for some constant $c_1 > 0$. In addition, $||\mathbf{p}_k|| \leq \gamma \Delta_k$ for some constant $\gamma \geq 1$. Then

$$\lim_{k \to \infty} J_k^T \mathbf{r}_k = 0 \tag{4.76}$$

Proof. From the smoothness of $\mathbf{r}_j$, i.e. $\mathbf{r}_j$ is infinitely differentiable. We can choose $M > 0$ such that $||J_k^T J_k|| \leq M$ for all k. f is bounded is bounded below by zero. Thus, Theorem 4.4 is satisfied. □

4.5.6 Computational Example

This example will illustrate the Levenberg-Marquardt (LM) algorithm 4.5.4. The following table shows the annual full-time student enrollment data from Califor-

nia State University, Los Angeles from 2005-2015 [21].

Table 4.1: California State University, Los Angeles full-time student enrollment data from 2005-2015

Year	Full-Time Student Enrollment
2005	15936
2006	16251
2007	16687
2008	16297
2009	15967
2010	16151
2011	17262
2012	17952
2013	18796
2014	20445
2015	23252

We fit the following nonlinear model function

$$p(t, \mathbf{x}) = x_2 \ln(x_1 t) + x_3 \tag{4.77}$$

using the LM algorithm 4.5.4.

The parameter vector changes after each k-th iterate:

$$\mathbf{x_k} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} \tag{4.78}$$

Our initial guess for $\mathbf{x_1}$ after a rough estimate will be:

$$\mathbf{x_1} = \begin{bmatrix} 100 \\ 50 \\ 100 \end{bmatrix} \tag{4.79}$$

The first step of the LM algorithm is to use the Gauss–Newton method.

$$r(\mathbf{x_1}) = \begin{bmatrix} 50\ln(100) + 100 - 15936 \\ 50\ln(200) + 100 - 16251 \\ 50\ln(300) + 100 - 16687 \\ 50\ln(400) + 100 - 16297 \\ 50\ln(500) + 100 - 15967 \\ 50\ln(600) + 100 - 16151 \\ 50\ln(700) + 100 - 17262 \\ 50\ln(800) + 100 - 17952 \\ 50\ln(900) + 100 - 18796 \\ 50\ln(1000) + 100 - 20445 \\ 50\ln(1100) + 100 - 23252 \end{bmatrix} = \begin{bmatrix} 15606 \\ 15886 \\ 16302 \\ 15897 \\ 15556 \\ 15731 \\ 16834 \\ 17518 \\ 18356 \\ 20000 \\ 22802 \end{bmatrix} \tag{4.80}$$

$||r(\mathbf{x_1})||^2 = 3.3510 * 10^9$, so $f(\mathbf{x_1}) = 1.6755 * 10^{13}$

Recall that the residual is defined as $r_j = |y_j - p(t_j, \mathbf{x})|$. Since the absolute function is not smooth, to ensure positivity by re-writing the residual r_j as a square function:

$$r_j^2 = (y_j - x_2 \ln(x_1 t_j) - x_3)^2 \tag{4.81}$$

The Jacobian is calculated:

$$
\begin{aligned}
J(\mathbf{x_1}) &= \begin{bmatrix} \frac{\partial r_j}{\partial x_1} & \frac{\partial r_j}{\partial x_2} & \frac{\partial r_j}{\partial x_3} \end{bmatrix} \\
&= \begin{bmatrix}
\frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \frac{\partial r_1}{\partial x_3} \\
\frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & \frac{\partial r_2}{\partial x_3} \\
\frac{\partial r_3}{\partial x_1} & \frac{\partial r_3}{\partial x_2} & \frac{\partial r_3}{\partial x_3} \\
\frac{\partial r_4}{\partial x_1} & \frac{\partial r_4}{\partial x_2} & \frac{\partial r_4}{\partial x_3} \\
\frac{\partial r_5}{\partial x_1} & \frac{\partial r_5}{\partial x_2} & \frac{\partial r_5}{\partial x_3} \\
\frac{\partial r_6}{\partial x_1} & \frac{\partial r_6}{\partial x_2} & \frac{\partial r_6}{\partial x_3} \\
\frac{\partial r_7}{\partial x_1} & \frac{\partial r_7}{\partial x_2} & \frac{\partial r_7}{\partial x_3} \\
\frac{\partial r_8}{\partial x_1} & \frac{\partial r_8}{\partial x_2} & \frac{\partial r_8}{\partial x_3} \\
\frac{\partial r_9}{\partial x_1} & \frac{\partial r_9}{\partial x_2} & \frac{\partial r_9}{\partial x_3} \\
\frac{\partial r_{10}}{\partial x_1} & \frac{\partial r_{10}}{\partial x_2} & \frac{\partial r_{10}}{\partial x_3} \\
\frac{\partial r_{11}}{\partial x_1} & \frac{\partial r_{11}}{\partial x_2} & \frac{\partial r_{11}}{\partial x_3}
\end{bmatrix} \\
&= \begin{bmatrix}
-326 & -185964 & -32604 \\
-318 & -190498 & -31795 \\
-311 & -193352 & -31113 \\
-315 & -201262 & -31462 \\
-337 & -220568 & -33669 \\
-350 & -234199 & -35036 \\
-367 & -249728 & -36712 \\
-400 & -276305 & -39999 \\
-456 & -319366 & -45604
\end{bmatrix}
\end{aligned}
\tag{4.82}
$$

Combining equation(4.13) and equation(4.28) from the Gauss–Newton method (GN), we get:

$$
\mathbf{p_k}^{GN} = -(J_k^T J_k)^{-1} J_k^T r_k \tag{4.83}
$$

Substituting our calculated values we get $\mathbf{p_1}^{GN} = (-36.9018, -2.6100, 0.4891)$

Once we go through one step of the GN algorithm, we compare $\mathbf{p_1}^{GN}$ to Δ_1. The trust regions acts as an indicator to see if we are within an acceptable range of the minimum of the minimization function $f(\mathbf{x})$ from equation (4.10). For illustrative purposes, let $\Delta_1 = 0.1$. In this case, $||\mathbf{p_1}^{GN}|| = 36.9972 > 0.1$. Because of this, we switch to the LM algorithm.

We can now initialize the LM algorithm. Going back to our initial guess $\mathbf{x_1}$,

$||r(\mathbf{x_1})||^2 = 3.3510 * 10^9$, so $f(\mathbf{x_1}) = 1.6755 * 10^{13}$, same as the initialization step of GN.

Let $\lambda_1 = 1$ as an initial guess. For the purposes of this illustration, we will use this algorithm only once.

So using $\lambda_1 = 1$ and equation (4.61), $\mathbf{p_1}^{LM} = (0.0050, -5.5109 * 10^{-10}, 0.5000)$.

Following the trust region algorithm (4.4.1), we now calculate ρ_k (4.84).

$$\rho_k = \frac{f(\mathbf{x_k}) - f(\mathbf{x_k} + \mathbf{p_k})}{m_k(\mathbf{0}) - m_k(\mathbf{p_k})} \tag{4.84}$$

(1)

$$f(\mathbf{x_1}) = 1.6755 * 10^{13} \tag{4.85}$$

(2)

$$f(\mathbf{x_1} + \mathbf{p_1}) = f(\mathbf{x_2}) = \frac{1}{2}||r(\mathbf{x_2})||^2 = 1.6754 * 10^9 \tag{4.86}$$

(3)

$$m_1(\mathbf{0}) = \frac{1}{2}||r(\mathbf{x_1})||^2 = f(\mathbf{x_1}) = 1.6755 * 10^{13} \tag{4.87}$$

(4)

$$m_k(\mathbf{p_k}) = 8.7897 * 10^{25} \tag{4.88}$$

Combining terms, we end up with:

$$\rho_1 = \frac{f(\mathbf{x_1}) - f(\mathbf{x_1} + \mathbf{p_1})}{m_1(\mathbf{0}) - m_1(\mathbf{p_1})} = 5.2461 * 10^{16} \tag{4.89}$$

For the purpose of illustration, let $\Delta_1 = 0.1$ and $\eta = 0.001$ From the trust algorithm (4.4.1), we keep the same trust region value, so $\Delta_2 = \Delta_1$. Since $\rho_1 > \eta$, $\mathbf{x_2} = \mathbf{x_1} + \mathbf{p_1}$

We can now update our parameter values:

$$\mathbf{x_2} = \mathbf{x_1} + \mathbf{p_1}^{LM} = \begin{bmatrix} 100 + .0050 \\ 50 + (-5.5109 * 10^{-10}) \\ 100 + .5000 \end{bmatrix} = \begin{bmatrix} 100.0050 \\ 50.0000 \\ 100.5000 \end{bmatrix} \tag{4.90}$$

For k = 2, we need to calculate λ_2 first with the trust region subproblem (4.4.1). When $k > 1$, λ in equation (4.61) is calculated using the trust region subproblem algorithm (4.5.2):

$$J_2^T J_2 + \lambda_1 I = \begin{bmatrix} 1340177.876 & 847098339.3 & 134024387.8 \\ 847098339.3 & 5.41941 * 10^{11} & 84714069006 \\ 134024387.8 & 84714069006 & 13403108838 \end{bmatrix} \tag{4.91}$$

We take the Cholesky Decomposition to get:

$$L_1 L_1^T = \begin{bmatrix} 1157.6605 & 0 & 0 \\ 731732.9440 & 80673.0616 & 0 \\ 115771.7532 & 0.7835 & 100.0069 \end{bmatrix} \begin{bmatrix} 1157.6605 & 731732.9440 & 115771.7532 \\ 0 & 80673.0616 & 0.7835 \\ 0 & 0 & 100.0069 \end{bmatrix} \tag{4.92}$$

Solving $\mathbf{p_1}^{(\lambda)}$ from equation (4.51):

$$\mathbf{p_1}^{(\lambda)} = \begin{bmatrix} 3350.1457 \\ -5.5320 * 10^{-5} \\ -32.9994 \end{bmatrix} \tag{4.93}$$

Solving $\mathbf{q_1}^{(\lambda)}$ from equation (4.52):

$$\mathbf{q_1}^{(\lambda)} = \begin{bmatrix} 2.8939 \\ -26.2486 \\ -3350.2041 \end{bmatrix} \tag{4.94}$$

Using the equation (4.53) we get:

$$\begin{aligned} \lambda_2 &= \lambda_1 + \left(\frac{||\mathbf{p_1}^{(\lambda)}||}{||\mathbf{q_1}^{(\lambda)}||}\right)^2 \left(\frac{||\mathbf{p_1}^{(\lambda)}|| - \Delta}{\Delta}\right) \\ &= 1 + \left(\frac{3.3503 * 10^4}{3.3503 * 10^4}\right)^2 \left(\frac{3.3503 * 10^4 - 0.1}{0.1}\right) = 3.3503 * 10^4 \end{aligned} \tag{4.95}$$

Using to calculate (4.61) to calculate $\mathbf{p_2}^{LM}$, we end up with:

$$\mathbf{p_2} = \begin{bmatrix} 0.0050 \\ 1.6264 * 10^{-5} \\ 0.5000 \end{bmatrix} \tag{4.96}$$

This implies:

$$\mathbf{x_3} = \mathbf{x_2} + \mathbf{p_2}^{LM} = \begin{bmatrix} 100.0100 \\ 50 \\ 100.9998 \end{bmatrix} \tag{4.97}$$

The following graph provides an illustration of the LM algorithm after a successive number of iterations:

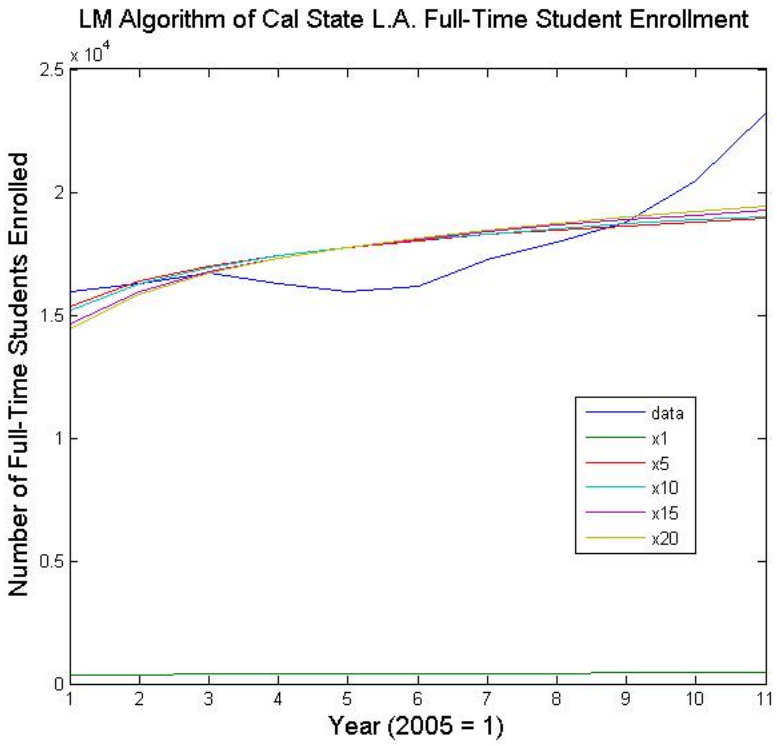


Figure 4.1: LM Algorithm fitting on Annual Cal State LA Full-Time Enrollment Data from 2005 - 2015

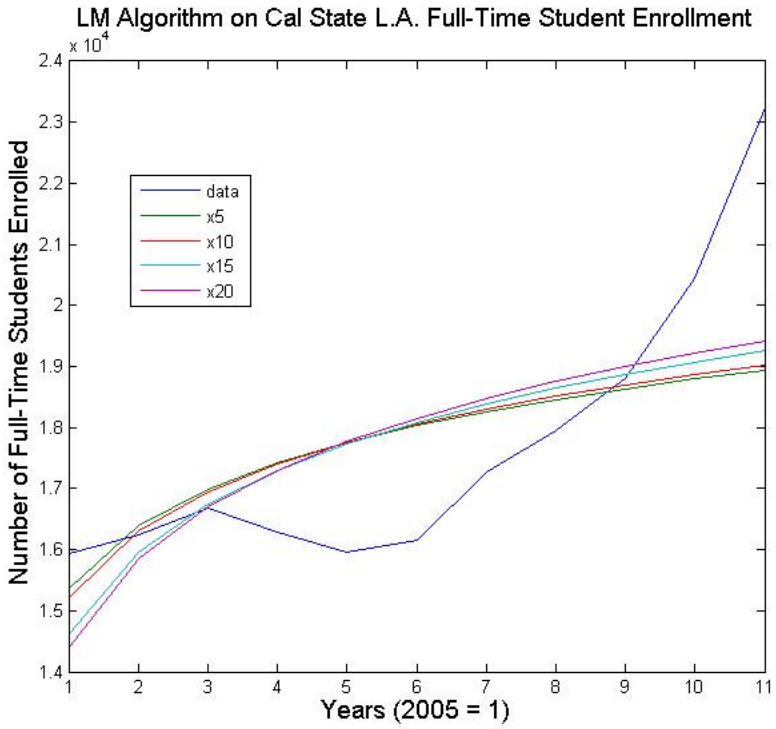


Figure 4.2: LM Algorithm fitting on Annual Cal State LA Full-Time Enrollment Data from 2005 - 2015

The LM algorithm ends once $\rho_k < \eta$.

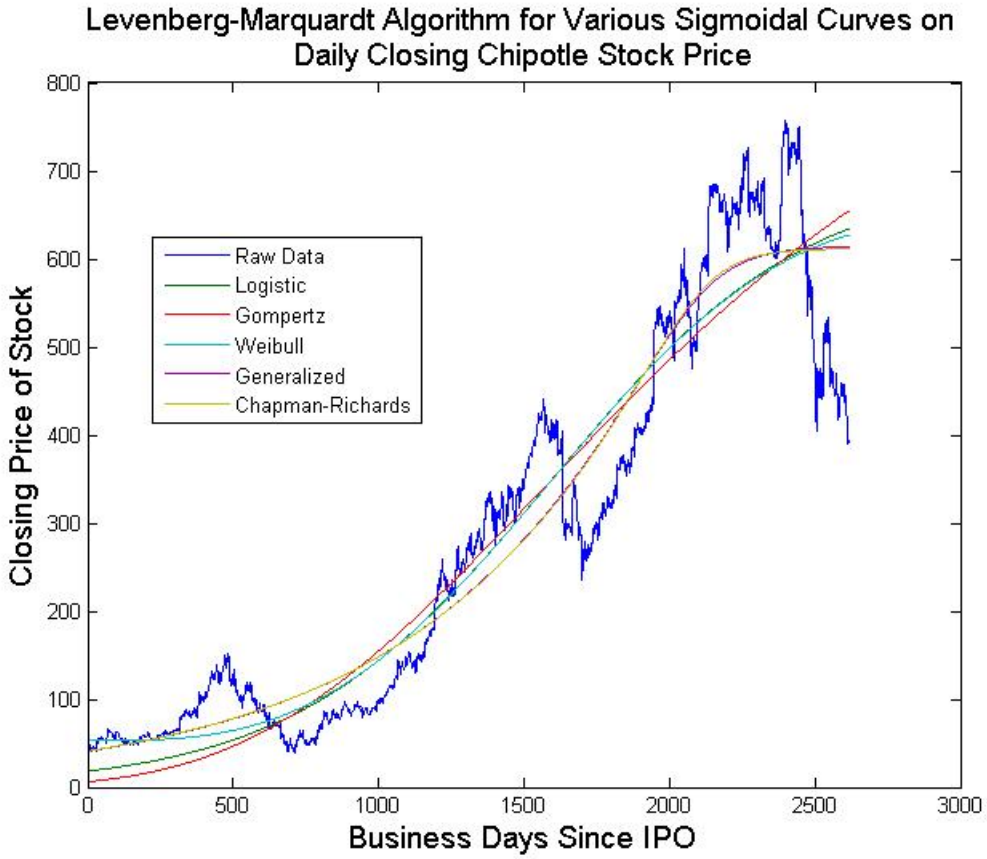


Figure 4.3: LM Algorithm of various sigmoidal curves and their respective mean square error (MSE).

Table 4.2: LM algorithm of various sigmoidal curves and their respective MSE

Curve Name	MSE
Logistic	4835.38127595731
Gompertz	5409.55782739912
Weibull	4548.42018423027
Generalized	4060.92655664517
Chapman-Richards	4005.64641784122

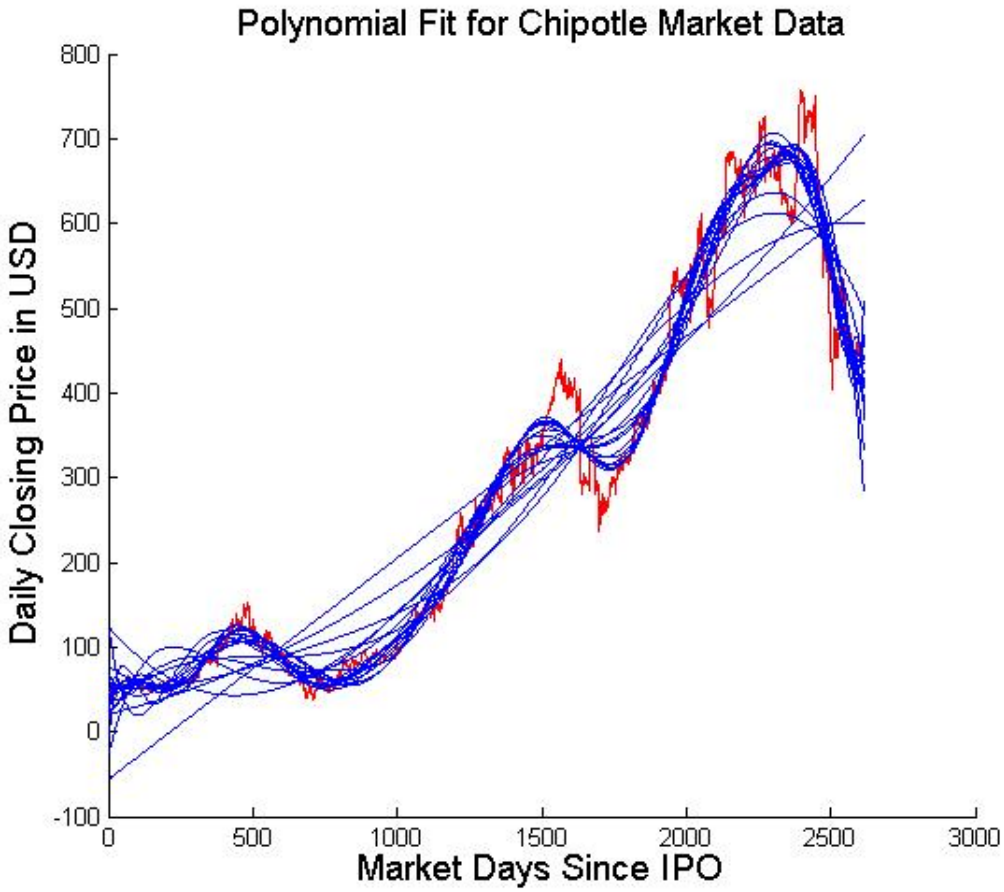


Figure 4.4: Polynomial algorithms of various degrees and their respective mean square error (MSE).

Table 4.3: Polynomial algorithms of various degrees and their respective mean square average (MSE)

Polynomial Degree	MSE
1	7362.6697517347902
2	6168.5780648502696
3	4615.8348964957704
4	3407.5441301470801
5	3107.53868716131
6	2235.1476172573798
7	2070.1495434897602
8	1433.1560713026099
9	1257.1509207751301
10	1191.5658148058201
11	1179.1434984611301
12	1178.4457355050699
13	1006.92989762918
14	924.50777729245601
15	868.82744941962801
16	833.82532793095197
17	829.35627632649903
18	823.47416471310203
19	822.90489838668702
20	780.12966874691404

CHAPTER 5

Forecasting Data

5.1 Methodology

This chapter will demonstrate the use of the Levenberg-Marquardt (LM) algorithm to fit data and forecast stock market prices. We filter the data with the Hodrick–Prescott (HP), exponential smoothing, and moving average techniques. Data without a filter applied is our standard of comparison. We will use the Logistic, Gompertz, Weibull, Chapman-Richards, and the Generalized Logistic equations after application of each respective filter.

All data fitted starts at the closing price of the initial public offering (IPO) to variable amounts of days chosen forward in time. The raw data is the daily closing prices of Vanguard Energy Fund Investor Shares (VGENX) [31]. It starts from May 23rd, 1984 to November 11th, 2016. The fund invests in US energy and foreign securities. The composition of the fund as of December 31st, 2016 is shown in this data table:

Table 5.1: Composition of VGENX Mutual Fund

Energy Fund Investor	as of 12/31/2016
Coal & Consumable Fuels	0.00%
Consumer Discretionary	0.10%
Consumer Staples	0.10%
Financials	0.20%
Health Care	0.10%
Industrials	0.20%
Information Technology	0.20%
Integrated Oil & Gas	36.10%
Oil & Gas Drilling	1.60%
Oil & Gas Equipment & Services	9.00%
Oil & Gas Exploration & Production	37.90%
Oil & Gas Refining & Marketing	7.20%
Oil & Gas Storage & Transportation	3.70%
Utilities	3.50%

From this data set, we start with the IPO to a certain number of days we assume to be known data. We call this "prior data." The prior data consists of 1000, 2000, 3000, 4000, 5000, 6000, and 7000 data points. From the prior data, we attempt to forecast a set number of days after the last prior data point. We attempt to forecast stock prices 100, 300, 1000, and 3000 trading days into the future. Prior to fitting the data with the LM algorithm, we either leave the prior data unfiltered, apply the Hodrick-Prescott filter, the moving average filter, or the exponential smoothing filter. The moving average filter is arbitrarily 300 trading days, which approximates one year's worth of trading. The weight factor α for the exponential average was chosen by taking the lowest mean square error value between the prior data and filtered data set in 0.1 intervals between 0 and 1. The forecast difference is defined as the actual data at the forecasted time point minus the fitted data at the forecast time point.

Positive values correspond to forecast underestimates, and negative values correspond to forecast overestimates.

5.2 Results

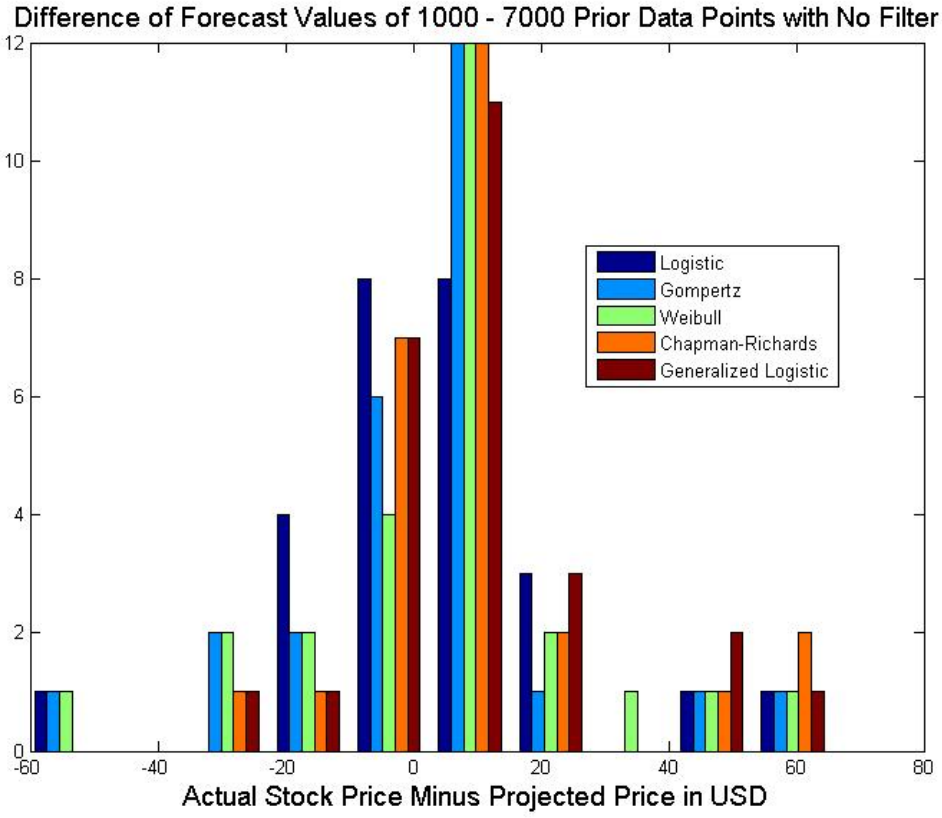
Since the raw data set is large, only 1000, 5000, and 7000 prior data points are provided with more detailed analysis. Their respective forecast plots, forecast difference bar graph, and MSE bar graph are shown in section 5.4. The reason for these choices is because 1000 prior data points is representative of initial behavior of a sigmoidal curve, 5000 prior points is representative of behavior immediately before inflection behavior, and 7000 prior data points is representative of behavior of a sigmoidal curve inclusive of the inflection point. In other words, these prior data points are representative of emergent, inflection, and saturation phases. The inflection point occurs roughly between 5000 - 6000 days after IPO. Histograms of forecast differences display all prior data sets from 1000 - 7000 prior data points. Data tables of each forecast differences and their mean square error (MSE) are located in appendix D.1.

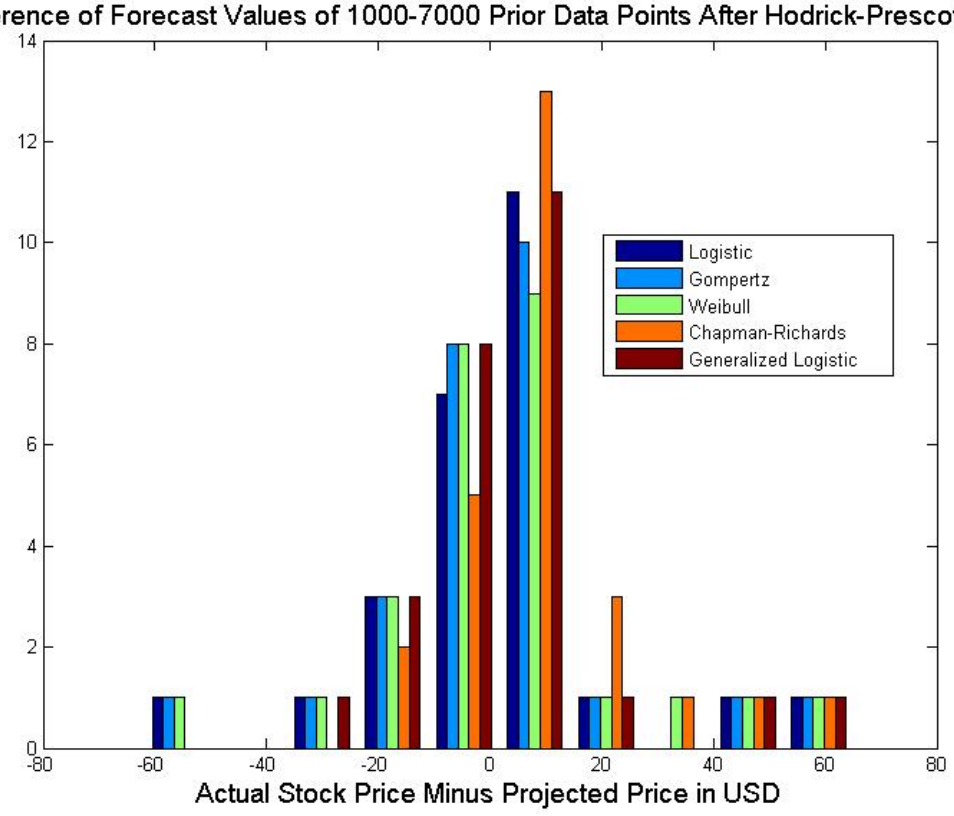
From section 5.4, the data set shows the MSE and forecast difference magnitude increases as the number of forecast days increases. For 1000 prior data points, the all MSE are less than 100 $\$^2$, which implies the mean error is within the square root of the MSE, or \$10. But if we look at 1000 forecast days or less, the MSE is generally less than $10\$^2$, or error that is roughly \$3.

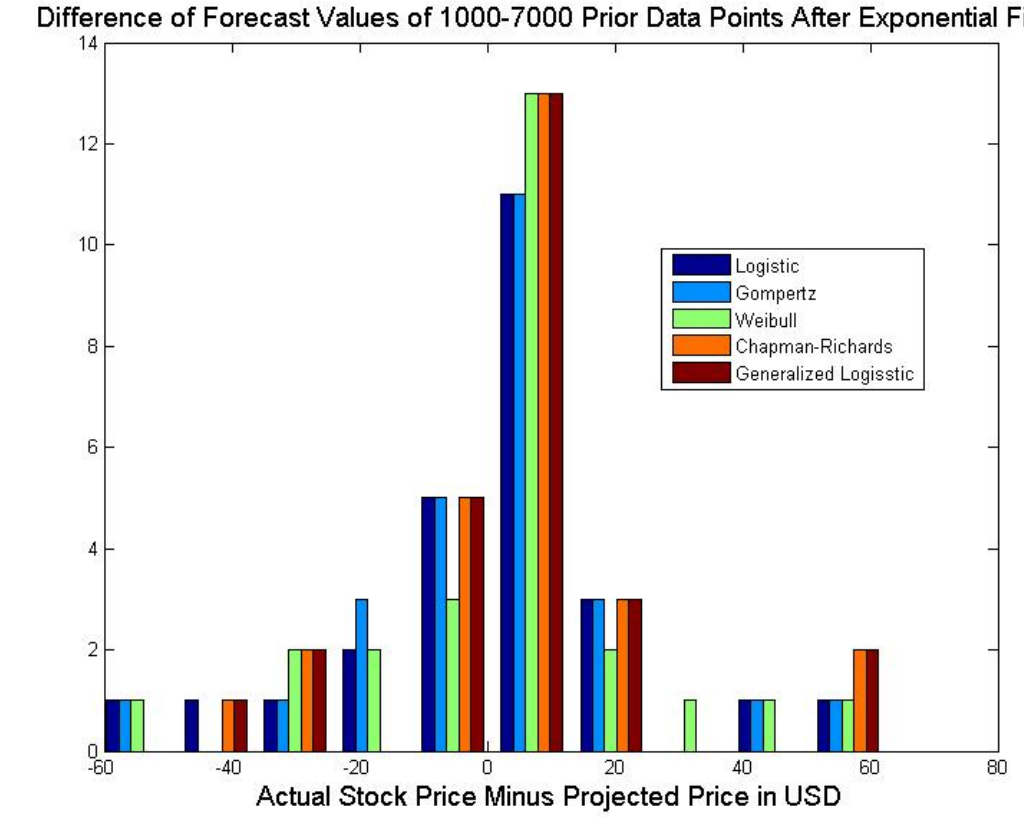
For 5000 prior data points, the MSE are generally less than 200 $\$^2$ for 100 and 300 forecast days, and range from 800 - 1800 $\$^2$ for 1000 to 3000 forecast days, which

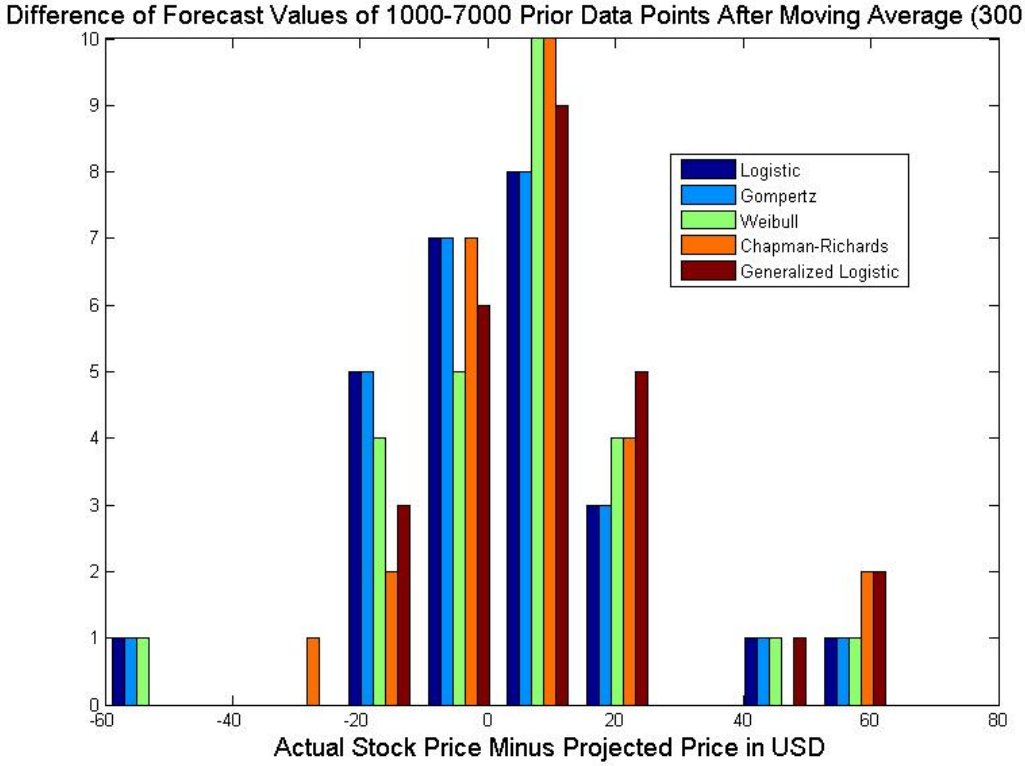
implies a mean error of about $\$10 - \$40$. Forecasting near or past the inflection point produces higher uncertainty. The forecast differences 300 and 1000 forecast days were generally high, with a range of approximately $\$10 - \$50$. The HP filter was able to keep 100 to 1000 forecast days all under \$20. But forecasting past the inflection point at 3000 forecast days, the forecast difference drops, but the MSE remains relatively high. This suggests there is much more volatility after the inflection point.

For 7000 prior data points, the forecast values are generally negative, meaning the sigmoidal curves generally overestimates the actual stock value. But once the maximum value for the stock price is known and the approximate location of the inflection point, the MSE are all below 1000 $\$^2$. The behavior of stock prices near the carrying capacity is less volatile than it is near the inflection point.


Difference of Forecast Values of 1000 - 7000 Prior Data Points with No Filter
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized Logistic
Actual Stock Price Minus Projected Price in USD


Difference of Forecast Values of 1000-7000 Prior Data Points After Hodrick-Prescott Filter
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized Logistic
Actual Stock Price Minus Projected Price in USD


Difference of Forecast Values of 1000-7000 Prior Data Points After Exponential Filter
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized Logisstic
Actual Stock Price Minus Projected Price in USD


Difference of Forecast Values of 1000-7000 Prior Data Points After Moving Average (300 Days)
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized Logistic
Actual Stock Price Minus Projected Price in USD

When plotting the histogram of forecast differences, positive values correspond to underestimates, and negative values correspond to overestimates. Ideally, we would like to have our difference curves skew to the right and be as close to zero as possible. This translates to accurate forecasts that are slightly underestimated. The standard deviation is generally within $30 for all filtration techniques and curves. The exponential filter appears to be worse than our control. A filter should provide a greater contrast in moment values so that it is easier to distinguish how each sigmoidal curve member is behaving. By inspection of minimums and maximums of teach statistical moment with each type of filtration and curve, the difference between the minimums and maximums of the exponential filter is the smallest. The minimum and maximum differences are smaller than unfiltered data. So the exponential filter performs worse than the control.

A normal distribution itself still does not demonstrate a particular sigmoidal curve is better at forecasting than another. A normal distribution near a mean of zero simply implies that the forecasts have a 50 % chance of forecasting above or below the actual price. We want the normal distributions to have a mean past zero and have a positive skew to that the probability of underestimating accurate data is greater than 50 %.

Table 5.2: Average of Forecast Differences

	Unfiltered	Hodrick-Prescott	Exponential	Moving Average
Logistic	2.966	1.263	1.757	1.508
Gompertz	1.982	1.604	1.931	2.331
Weibull	4.930	1.911	3.468	3.607
Chapman-Richards	9.161	8.542	5.312	6.072
Generalized	10.406	4.207	5.179	10.163

Note: Red highlighted are the lowest values of a column, and blue values the highest.

Table 5.3: Standard Deviation of Forecast Differences

	Unfiltered	Hodrick-Prescott	Exponential	Moving Average
Logistic	22.059	22.832	22.706	22.498
Gompertz	22.837	22.496	22.860	22.020
Weibull	22.313	22.051	22.482	22.102
Chapman-Richards	19.478	17.899	20.444	19.127
Generalized	19.433	18.358	20.515	20.334

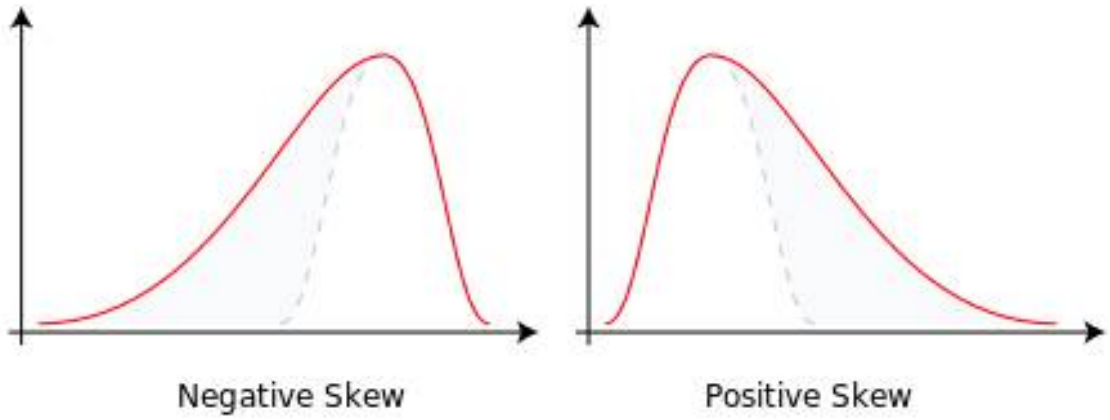
Note: Red highlighted are the lowest values of a column, and blue values the highest.

Table 5.4: Histogram of Skews of Forecast Differences

	Unfiltered	Hodrick-Prescott	Exponential	Moving Average
Logistic	0.056	0.035	0.161	0.224
Gompertz	0.036	0.025	0.032	0.181
Weibull	0.247	0.255	0.179	0.296
Chapman-Richards	1.252	1.600	0.732	1.374
Generalized	1.259	1.279	0.718	1.274

Note: Red highlighted are the lowest values of a column, and blue values the highest.

The skew is the measure of asymmetry about the mean. Positive skew values means the data is more spread out to the right of the mean, and negative skew values means the data is more spread out to the left of the mean. Here is a general schematic of skew [29]:



The following is the equation to calculate skew [23]:

$$s = \frac{E(x-\mu)^3}{\sigma^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}\right)^3} \tag{5.1}$$

For the applications of this thesis, positive skew values correspond to higher frequency of positive forecast differences. The Chapman-Richards and Generalized Logistic equations have the largest positive skew values, meaning forecast values tend to underestimate actual values. The Logistic and Gompertz equations have the lowest skew values for all filtration techniques, implying that their distributions are close to symmetry about the mean.

Table 5.5: Kurtosis

	Unfiltered	Hodrick-Prescott	Exponential	Moving Average
Logistic	5.856	5.259	4.961	5.543
Gompertz	5.262	5.492	5.283	5.850
Weibull	4.352	5.003	4.874	5.448
Chapman-Richards	4.947	6.131	5.120	5.636
Generalized	4.970	5.725	5.091	4.383

Note: Red highlighted are the lowest values of a column, and blue values the highest.

The kurtosis is the measure of "tailedness" of a normal distribution, not the sharpness nor relative height of a normal distribution's peak [8]. In other words,

kurtosis is the measure outliers normalized to the standard deviation. The kurtosis does not directly imply the shape of a peak for a given distribution.

The following is the equation to calculate kurtosis [22]:

$$k = \frac{E(x-\mu)^4}{\sigma^4} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^4} \tag{5.2}$$

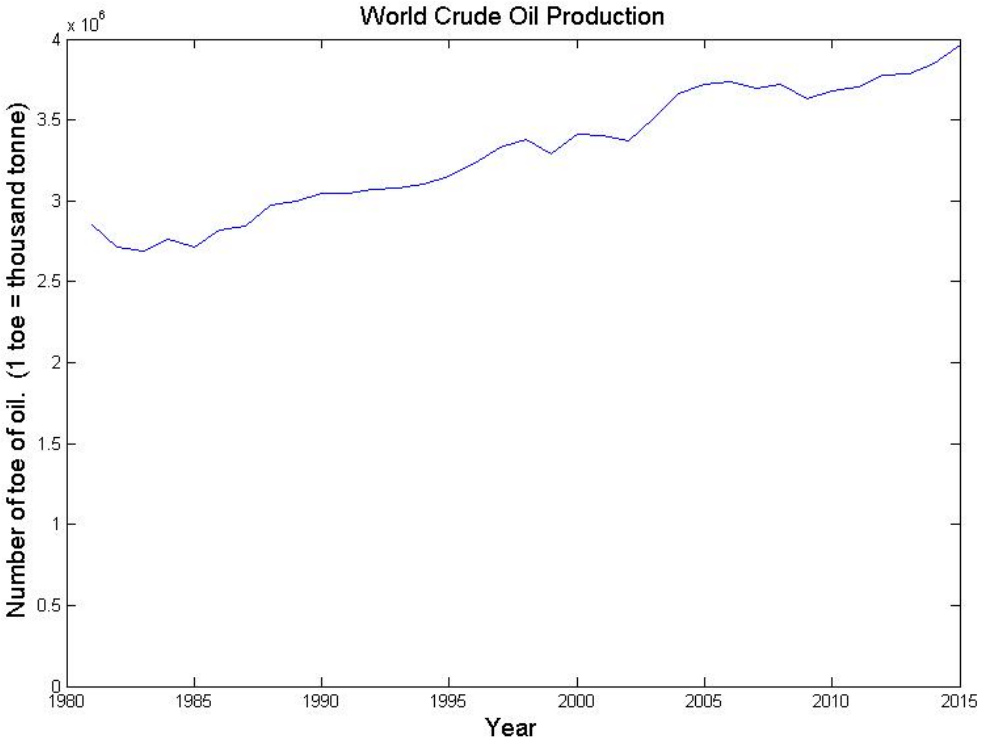
The forecast distributions should ideally show low kurtosis, which would imply the data would have few outliers and be close to the mean. Low kurtosis implies that the data would have forecasts far from the mean. The Logistic equation generally has the lowest kurtosis, which implies fewest outliers relative to the mean. But our data also shows that for the logistic equation, the mean is close to zero, and the standard deviation is very high, so we end up with a low kurtosis. The Gompertz has high kurtosis because the it has many outliers, so the numerator has a high value. The Chapman-Richards with the Hodrick–Prescott filter have high kurtosis because it has the lowest standard deviation. The Generalized Logistic with the moving average has a low kurtosis because is contains few outliers relative to the mean.

Another important feature of the data is the value of the fitted inflection point. The inflection point defines where the growth rate transitions from acceleration to deceleration, meaning the change of growth rate goes from positive to negative. From visual inspection, the inflection point occurs between 5000-6000 points after IPO. When trying to fit data with prior data near the inflection point, the fits forecast an extremely high carrying capacity, generally to the order of $\$10^3 - \$10^4$. This is because the fitting algorithm assumes that the high growth rate will last for a long time. The growth rate is the highest near the inflection point because the growth

rate is approaching an extrema value with respect to the second derivative of the closed form solution. Since we are fitting sigmoidal curves, the curve to the left of the inflection point exhibits positive concavity, and the curve to the right of the inflection point exhibits negative concavity. For all forecast differences, most of the differences were positive before encountering the inflection point and negative after the inflection point. This implies the sigmoidal curve generally underestimates the actual data prior to the inflection point, and overestimates after the inflection point. If the inflection point is known, an investor can take advantage of this behavior and determine whether to by or sell stock.

The MSE before and after the inflection point increases dramatically. If we focus our attention to 1000 and 3000 forecast days, for 1000 prior data points, all the MSE were less than 40 $\$^2$. 3000 forecast days implies 4000 days after IPO, so this time point is not past the inflection point. The increase is apparent for 5000 prior data points. The MSE for 1000 and 3000 forecast days are around $1000\$^2$. This implies stock prices have greater volatility after the inflection point.

Macroeconomic reasons can explain why stock prices have greater volatility. Baumeister and Peersman [1] and Robe and Wallen [17] show fluctuations in physical crude oil inventories are the biggest factor that determine oil prices. Crude oil production reported by the OECD has been fairly stable since VGENX went into IPO [27]:



But what has changed is the oil price elasticity. Price elasticity of is percentage change of the quantity demanded/supplied to the percent change in price [3]. Baumeister et al. [1] shows the price elasticity of crude oil decreases from 1985 to 2010. Low price elasticity implies small fluctuations in supply results in large changes in price.

Based on the data and analysis results, the Hodrick-Prescott (HP) filter used in tandem with the Chapman-Richards equation provide the best forecast results for this particular data set. The HP filter provides the largest difference in extreme values for the standard deviation, skew, and kurtosis. This allows for a bigger contrast when looking at the differences in moments between each type of equation. The Chapman-Richards provided consistent high average forecast differences for 2 out of 3 filtering techniques, which demonstrates its tendency to underestimate forecasts.

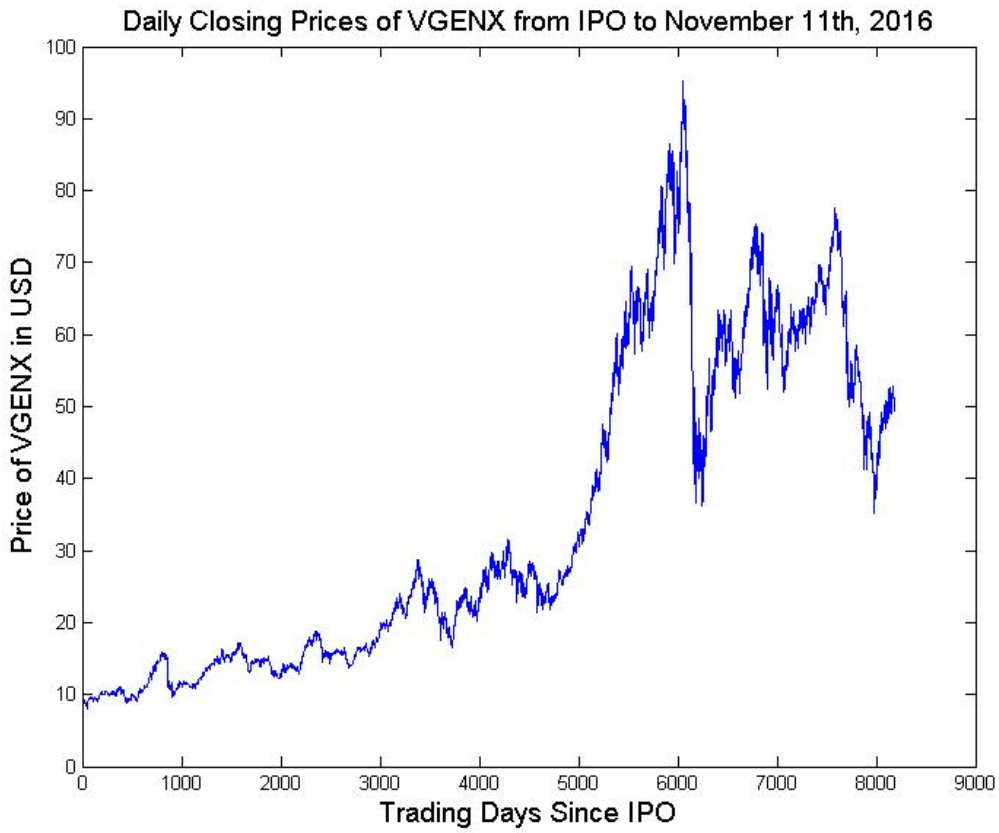
The Chapman-Richards consistently demonstrated the smallest standard deviation for all filtering techniques, so most differences are least likely to produce extreme values. The Chapman-Richards produces the highest positive skew values, so the forecast differences tend to underestimate actual values. The combination of the HP filter and Chapman-Richards has the highest kurtosis, which implies most of the data is concentrated near the mean.

5.3 Future Research

This paper used a deterministic approach to forecasting stock market data. Another path to explore could explore correlations with other stock market indicators, such as oil rig counts or the VIX indicator, to see how it would influence forecasting with sigmoidal curves as suggested by Baumeister et al [1]. One can also explore how to incorporate sigmoidal curves into agent based modeling [4]. One can also explore applications to mean reversion theory and how to determine the best conditions to sell stock [18].

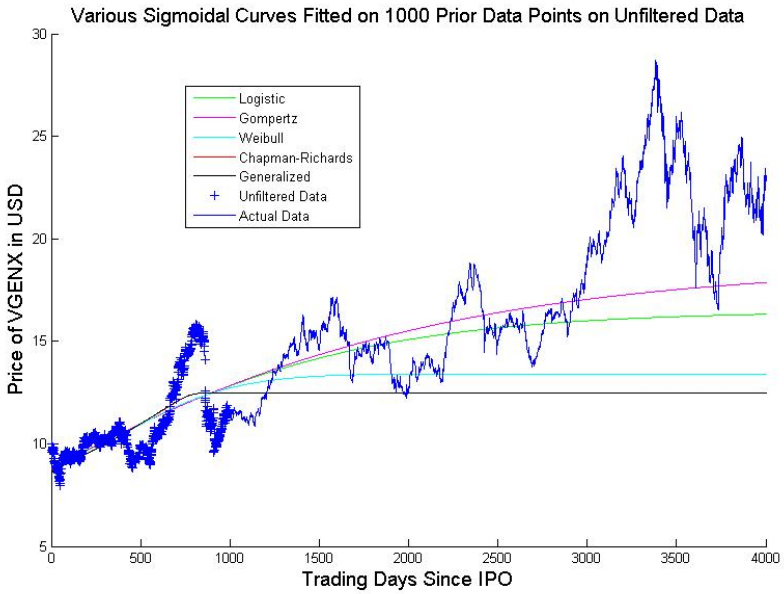
5.4 Data

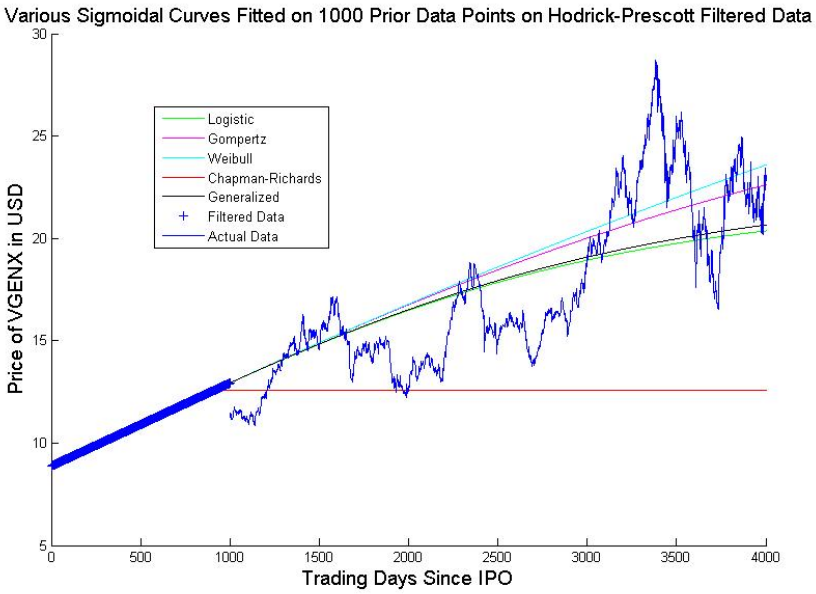
5.4.1 Raw Data

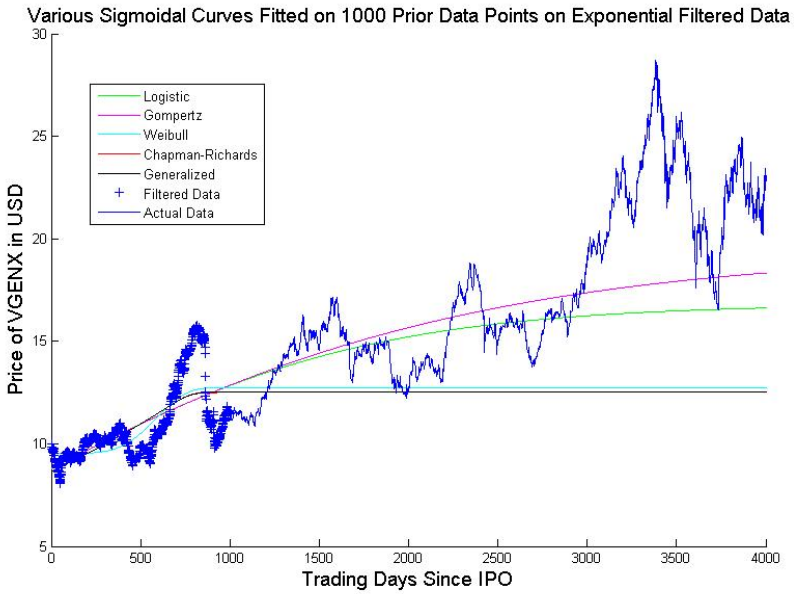


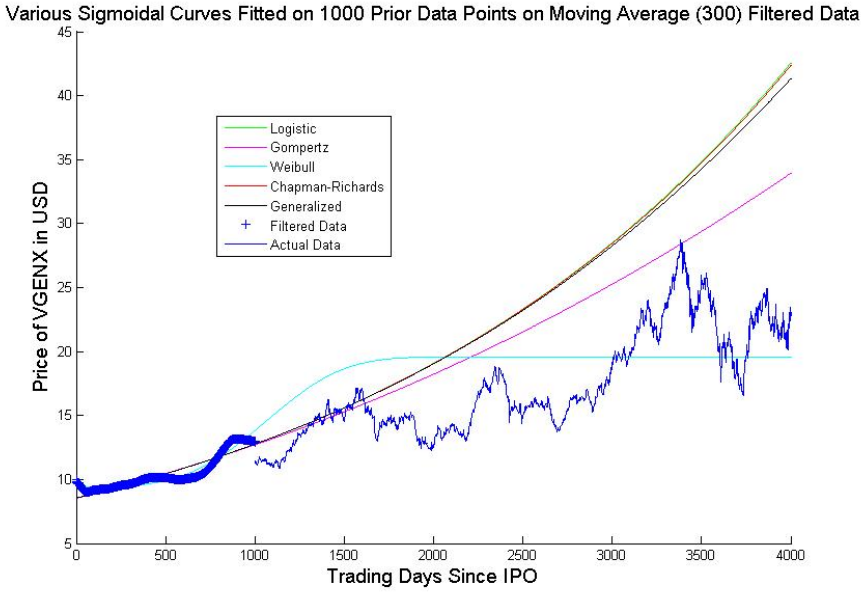
5.4.2 Fit of Various Sigmoidal Curves

1000 Prior Data Points

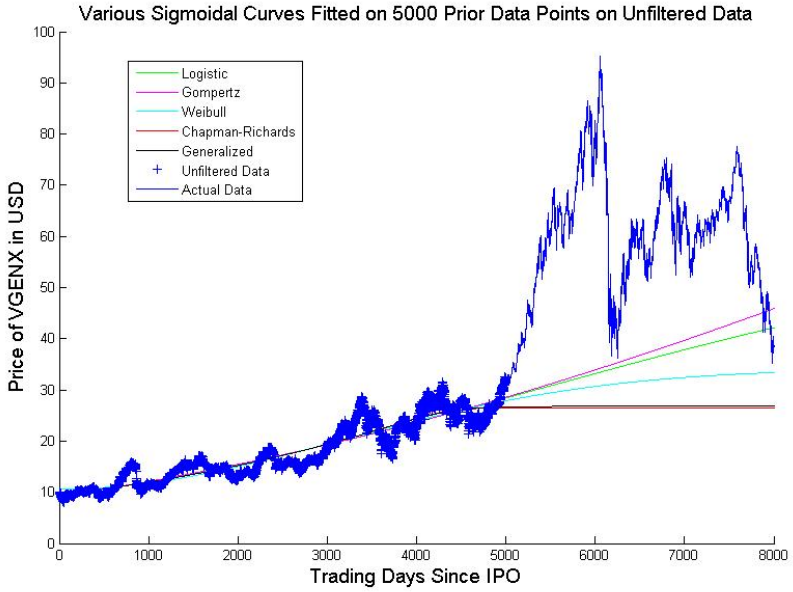


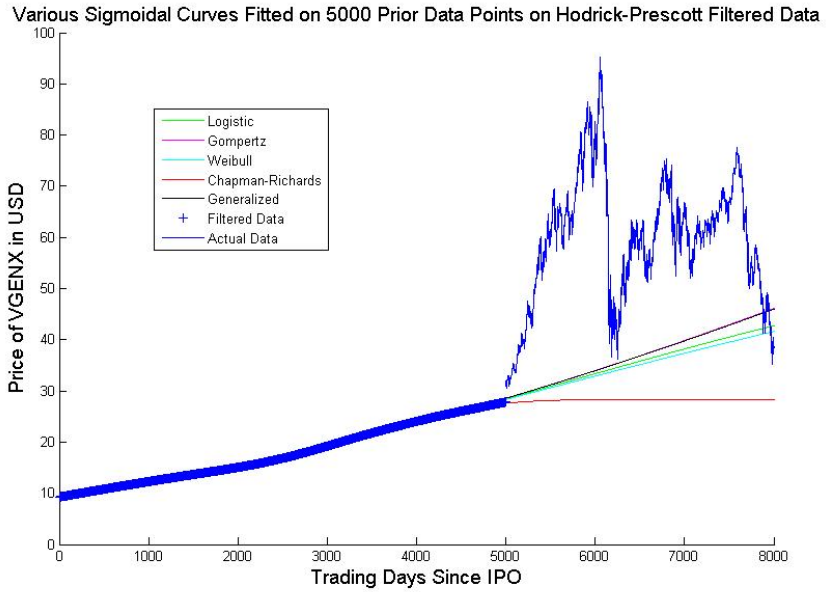


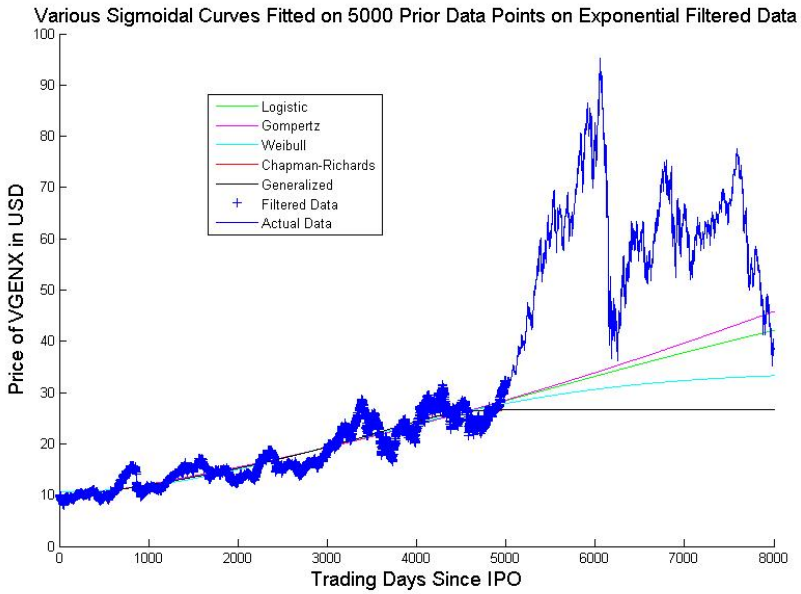

Various Sigmoidal Curves Fitted on 1000 Prior Data Points on Exponential Filtered Data
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized
Filtered Data
Actual Data
Price of VGENX in USD
Trading Days Since IPO

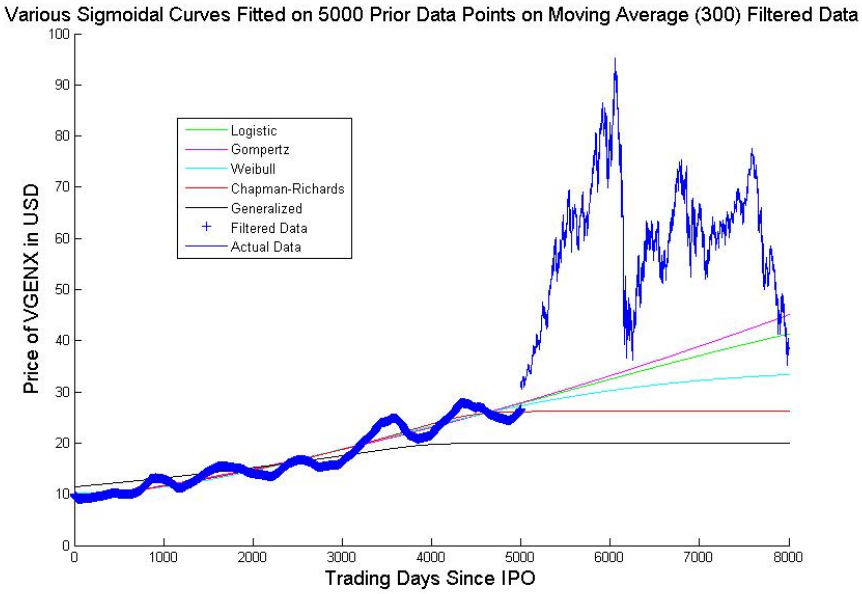

Various Sigmoidal Curves Fitted on 1000 Prior Data Points on Moving Average (300) Filtered Data
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized
Filtered Data
Actual Data
Price of VGENX in USD
Trading Days Since IPO

5000 Prior Data Points

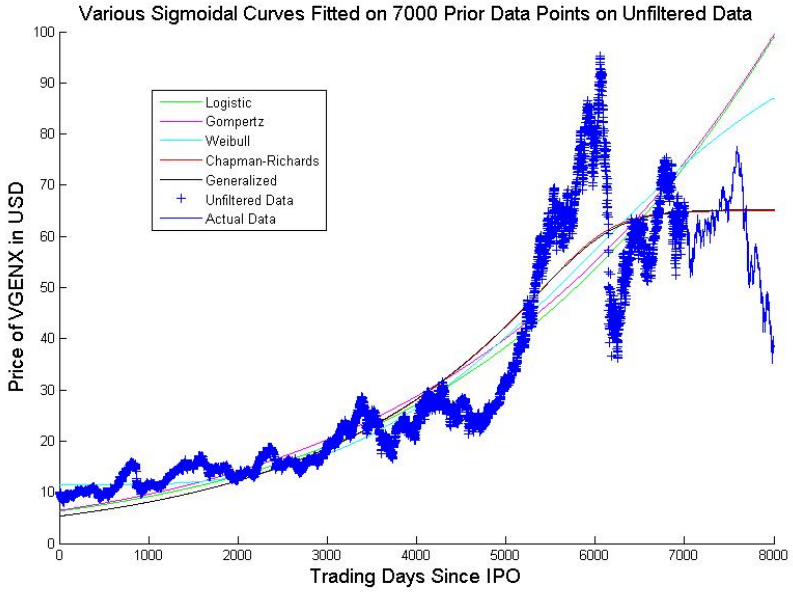


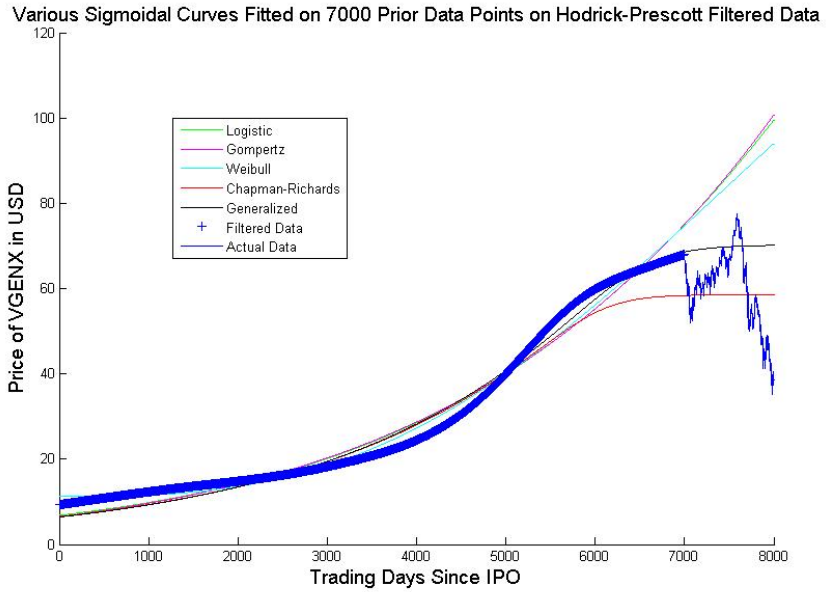


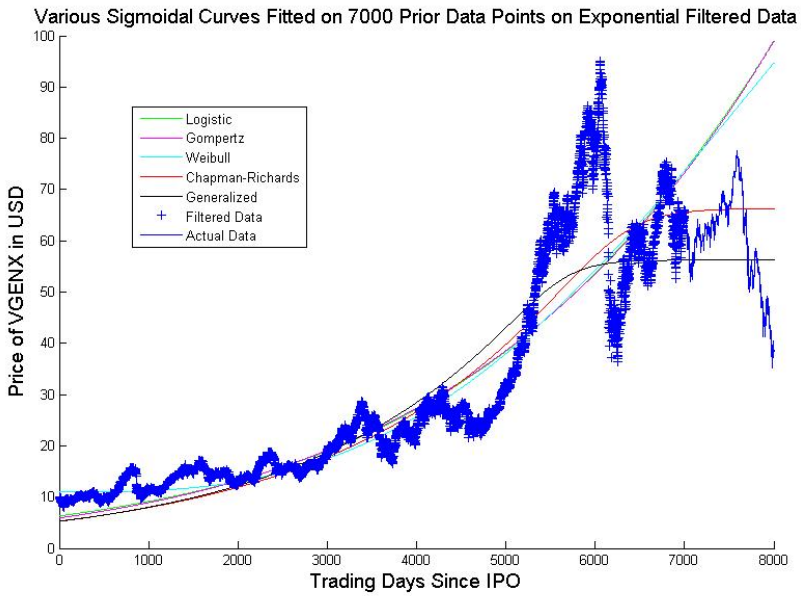

Various Sigmoidal Curves Fitted on 5000 Prior Data Points on Exponential Filtered Data
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized
Filtered Data
Actual Data
Price of VGENX in USD
Trading Days Since IPO
0
10
20
30
40
50
60
70
80
90
100
0
1000
2000
3000
4000
5000
6000
7000
8000

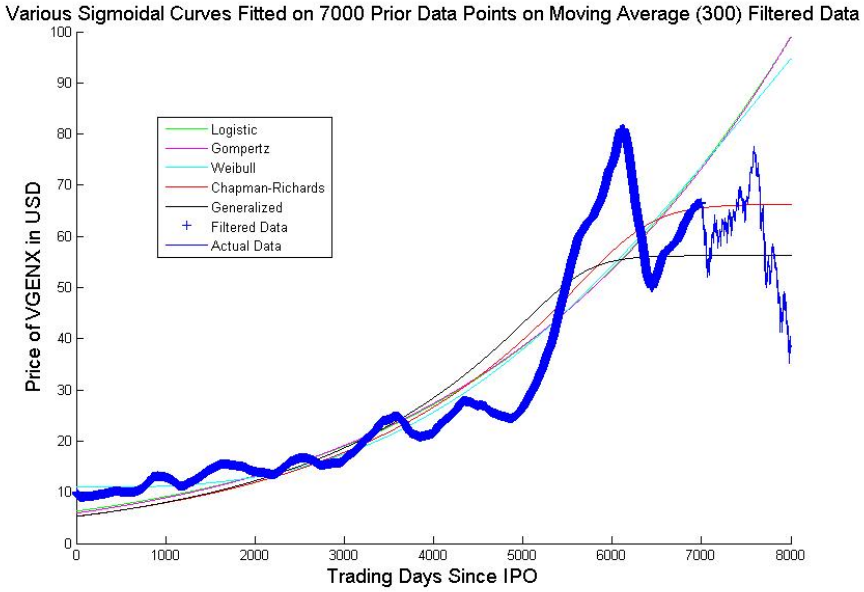

Various Sigmoidal Curves Fitted on 5000 Prior Data Points on Moving Average (300) Filtered Data
Logistic
Gompertz
Weibull
Chapman-Richards
Generalized
Filtered Data
Actual Data
Price of VGENX in USD
Trading Days Since IPO
0
10
20
30
40
50
60
70
80
90
100
0
1000
2000
3000
4000
5000
6000
7000
8000

7000 Prior Data Points



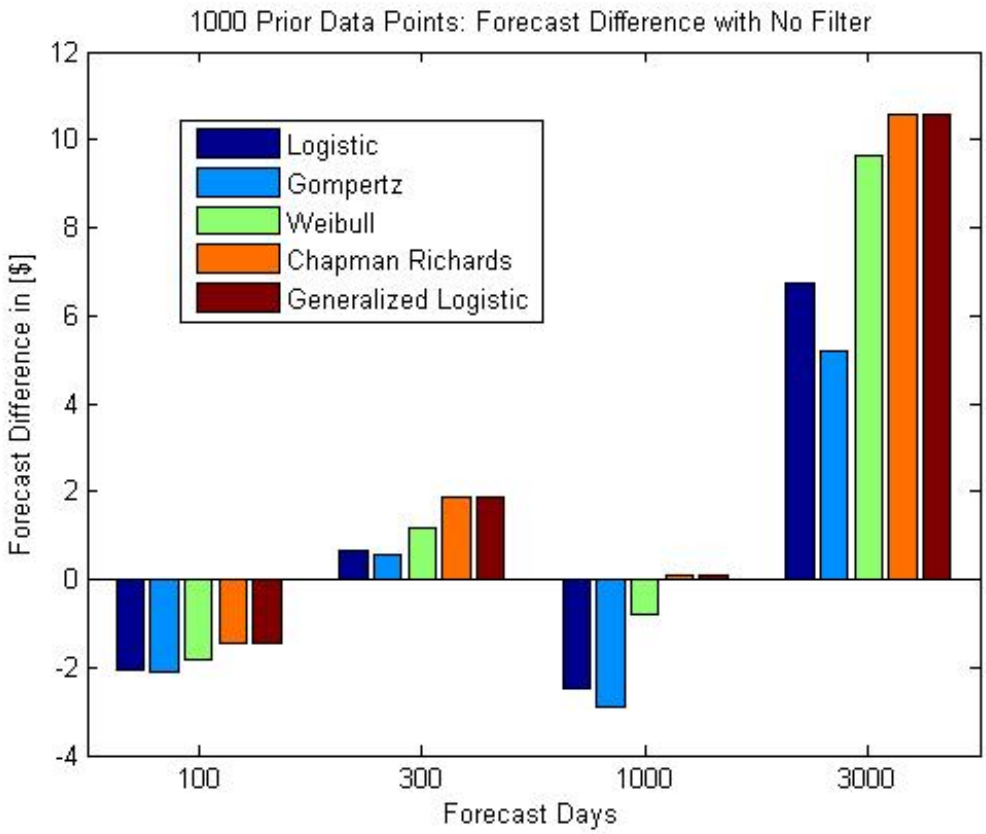


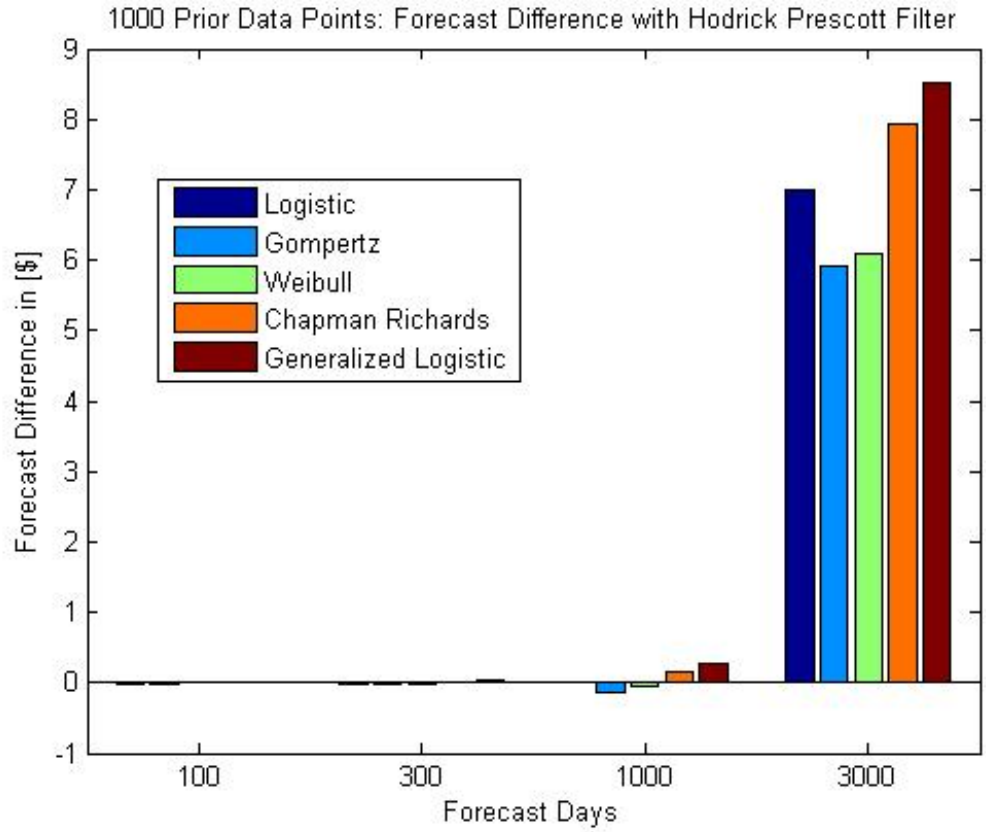


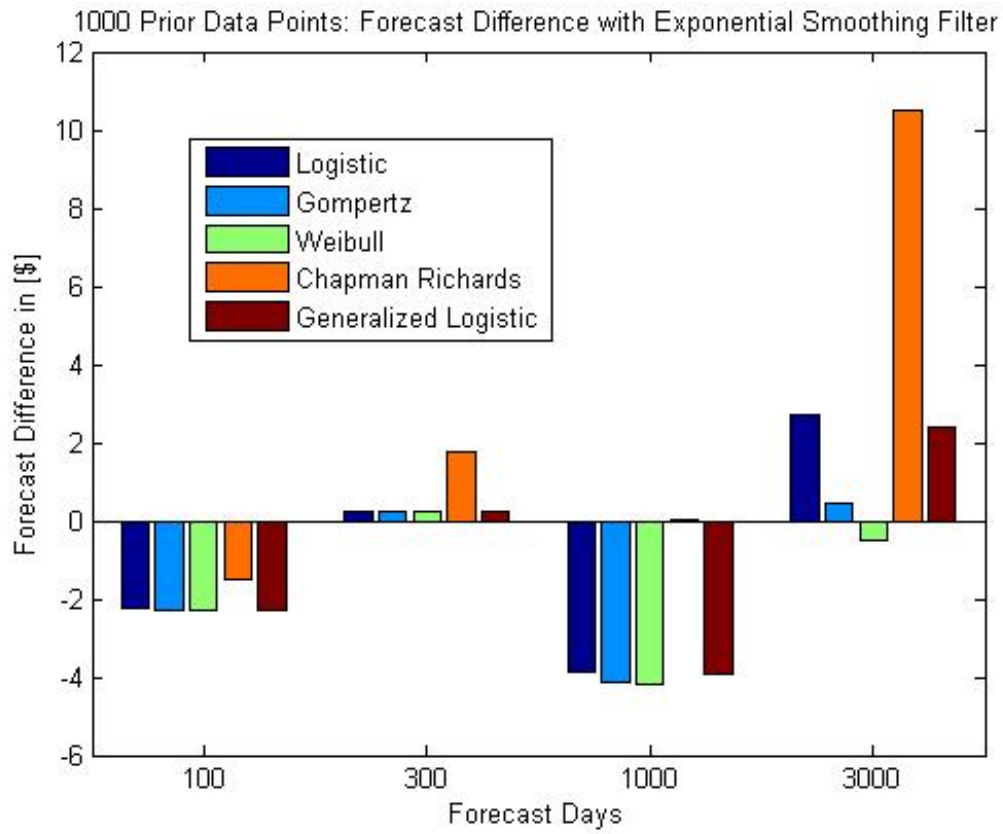


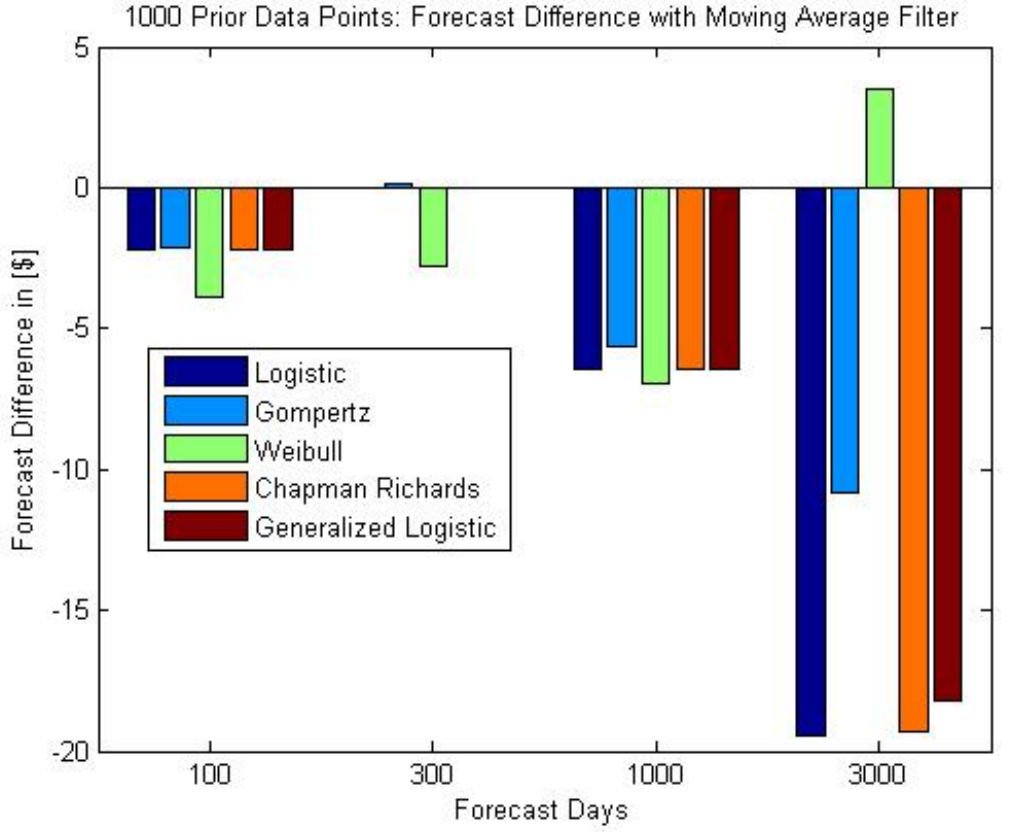
5.4.3 Forecast Difference with 1000 Prior Known Days

Note: Negative correspond to overestimation in forecasts, while positive numbers correspond to underestimation of forecasts. Units are in U.S. dollars.

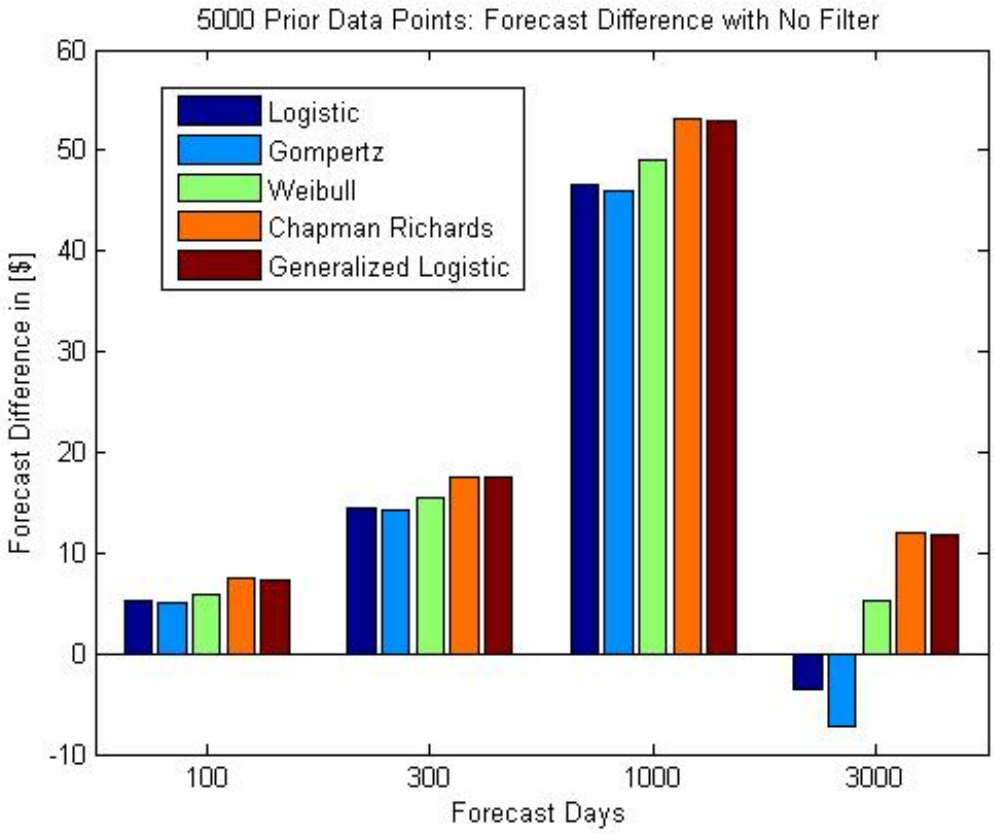

1000 Prior Data Points: Forecast Difference with No Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
Forecast Difference in [$]
Forecast Days
12
10
8
6
4
2
0
-2
-4
100
300
1000
3000

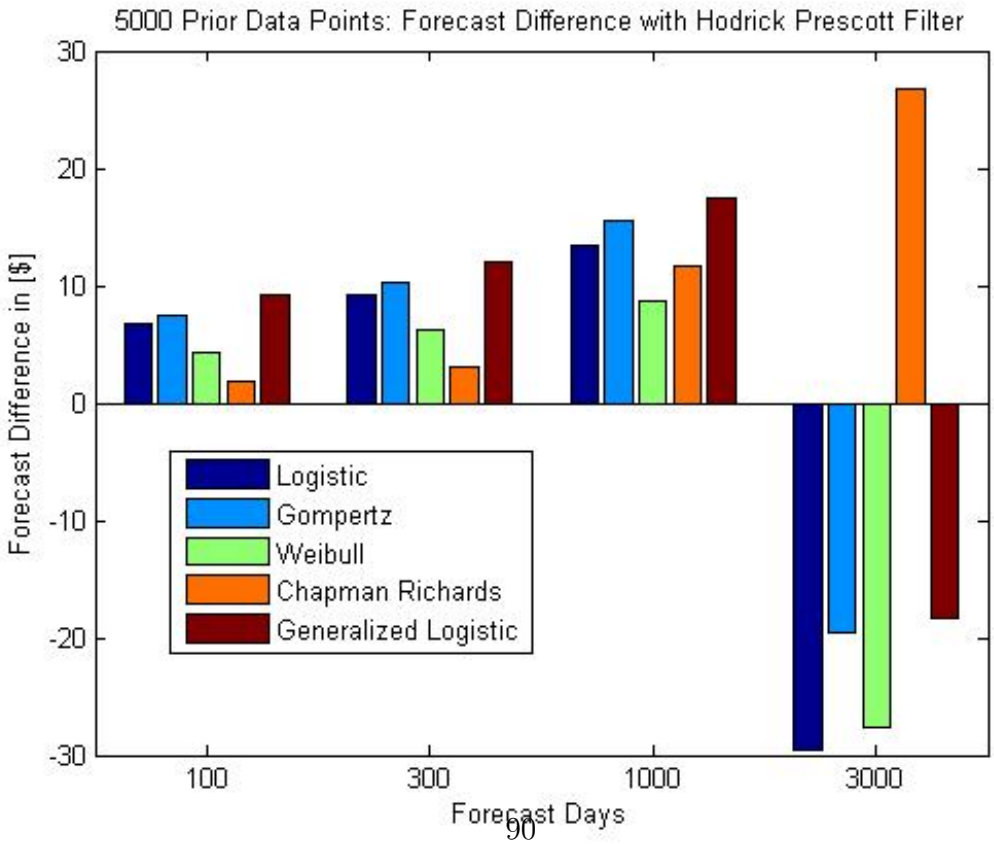

1000 Prior Data Points: Forecast Difference with Hodrick Prescott Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
Forecast Difference in [$]
Forecast Days
9
8
7
6
5
4
3
2
1
0
-1
100
300
1000
3000

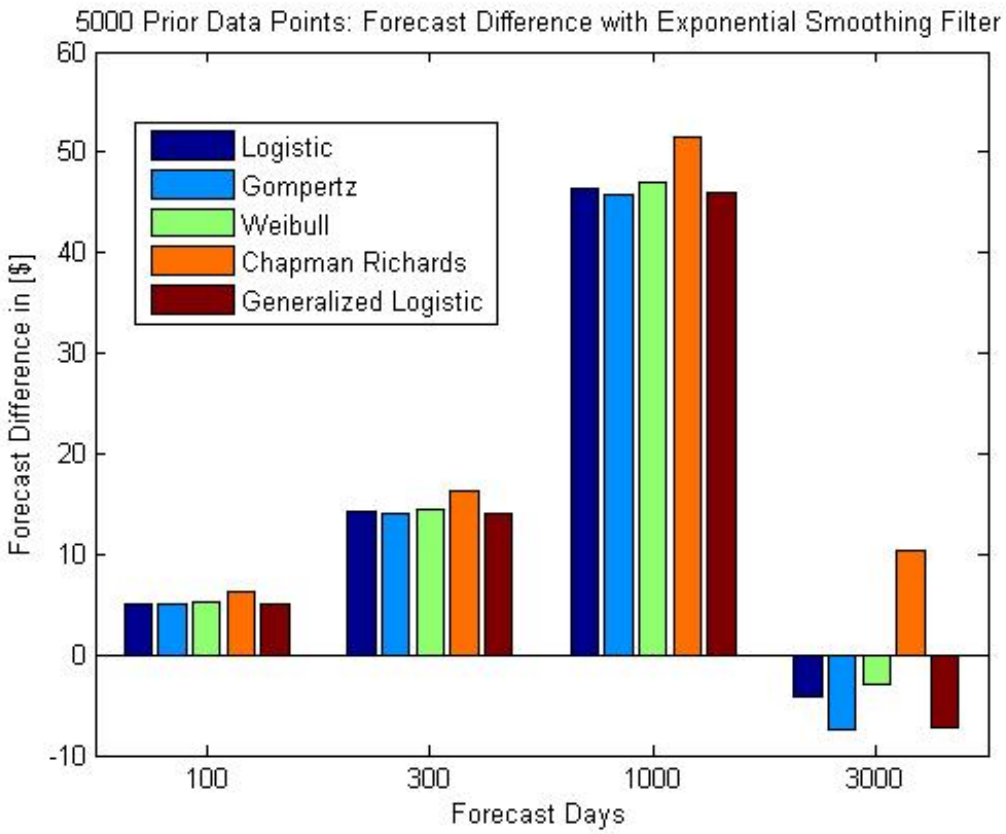

1000 Prior Data Points: Forecast Difference with Exponential Smoothing Filter
Forecast Difference in [$]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
12
10
8
6
4
2
0
-2
-4
-6
100
300
1000
3000

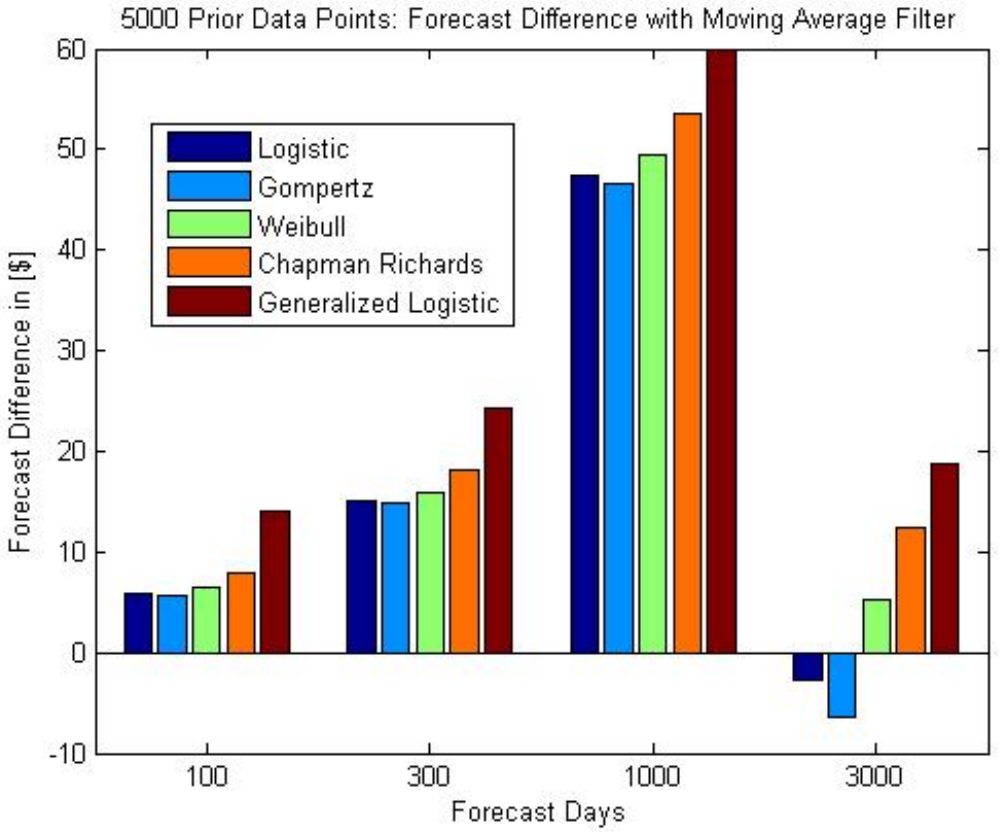

1000 Prior Data Points: Forecast Difference with Moving Average Filter
Forecast Difference in [$]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
5
0
-5
-10
-15
-20
100
300
1000
3000

5.4.4 Forecast Difference with 5000 Prior Known Days

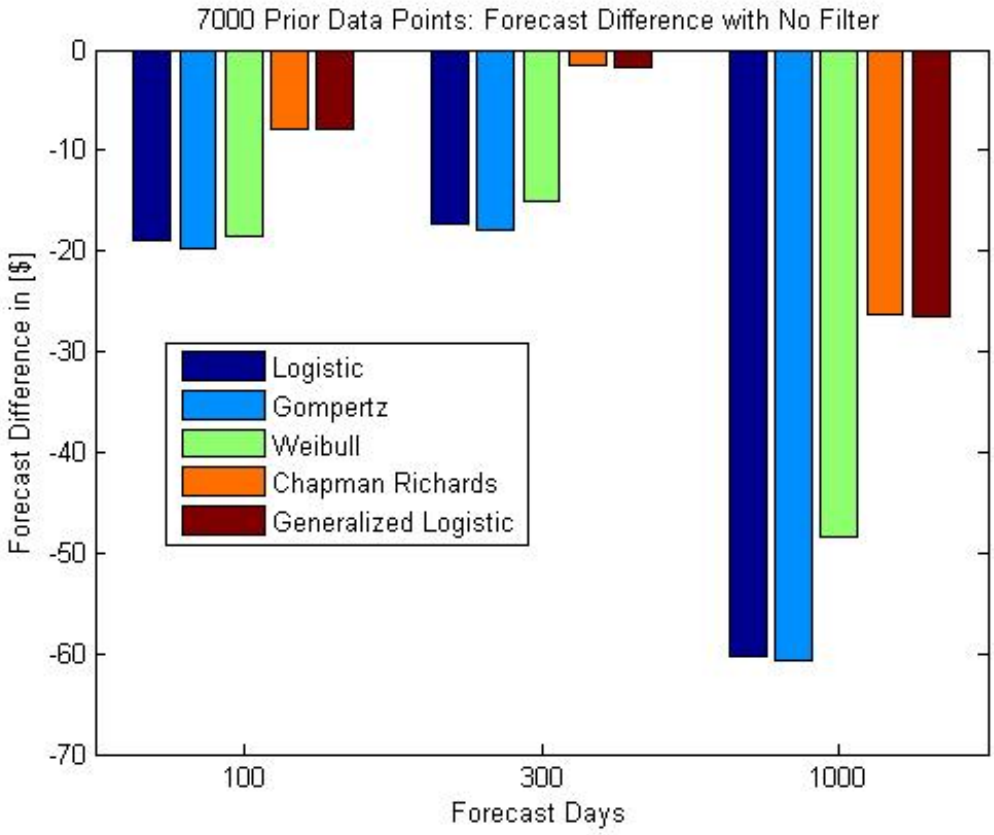


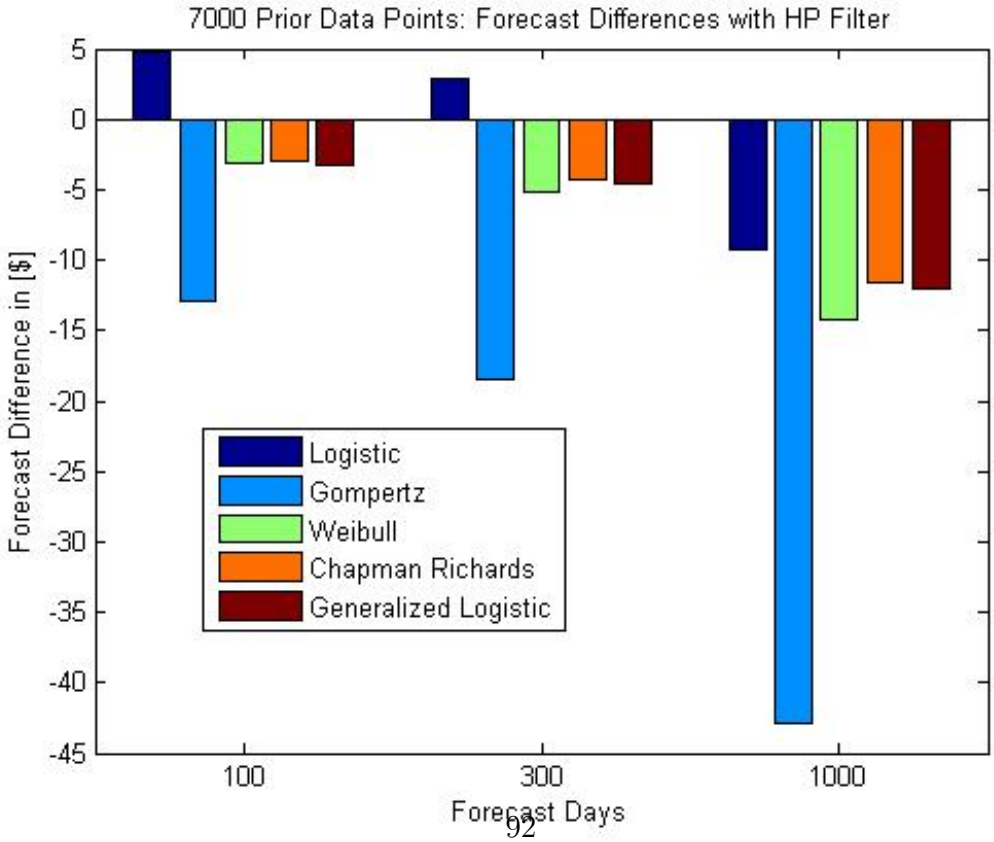


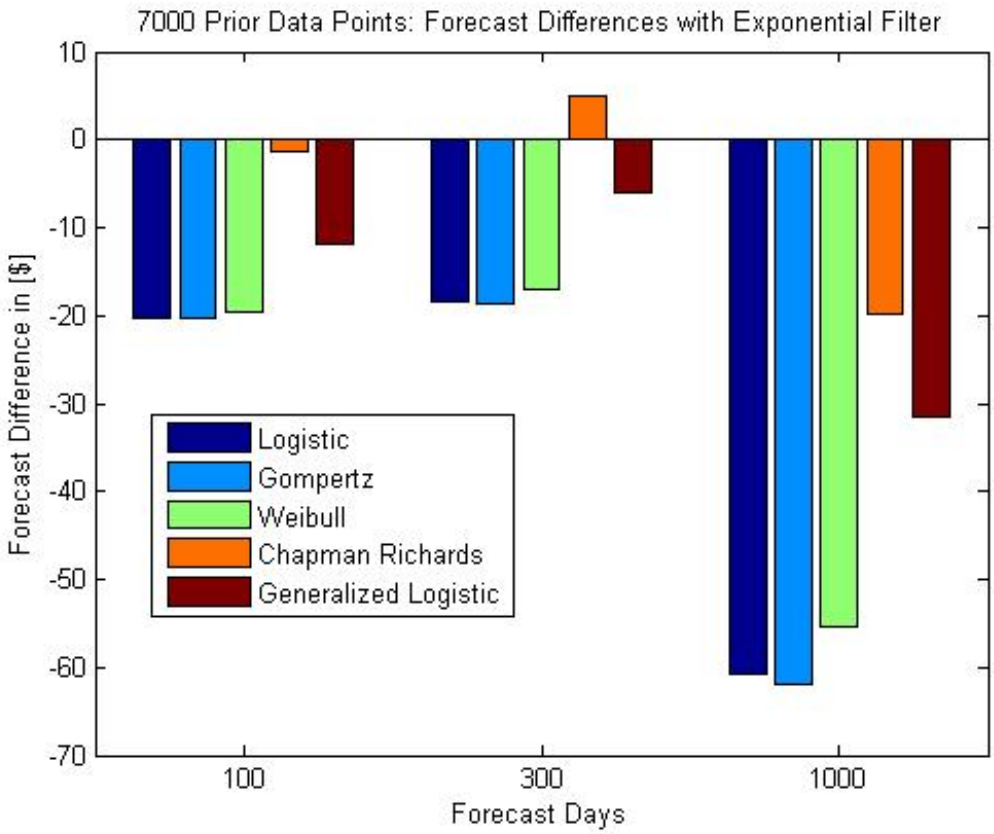

5000 Prior Data Points: Forecast Difference with Exponential Smoothing Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
Forecast Difference in [$]
60
50
40
30
20
10
0
-10
100
300
1000
3000
Forecast Days

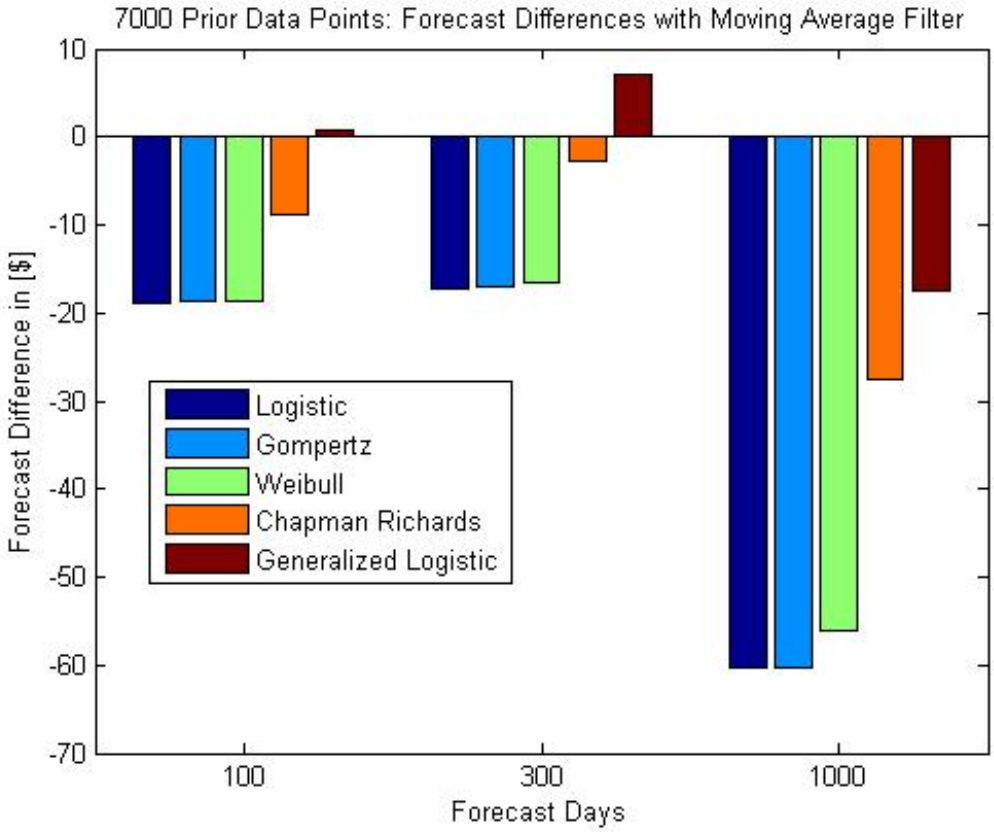

5000 Prior Data Points: Forecast Difference with Moving Average Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
Forecast Difference in [$]
60
50
40
30
20
10
0
-10
100
300
1000
3000
Forecast Days

5.4.5 Forecast Difference with 7000 Prior Known Days

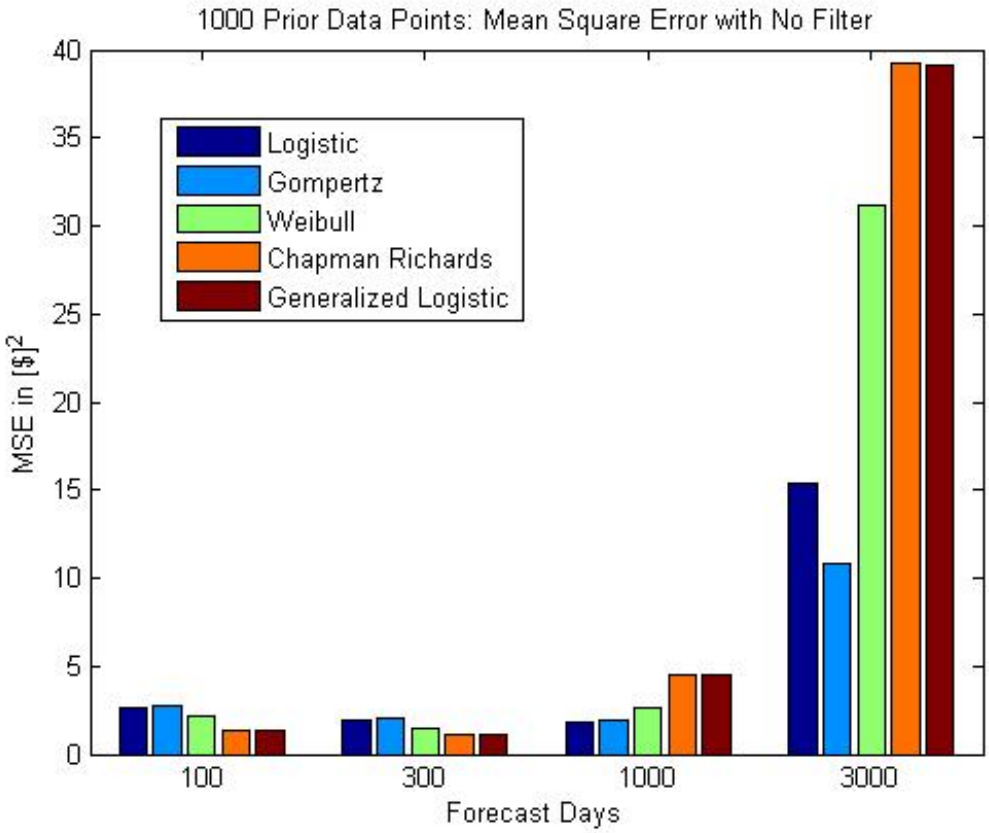


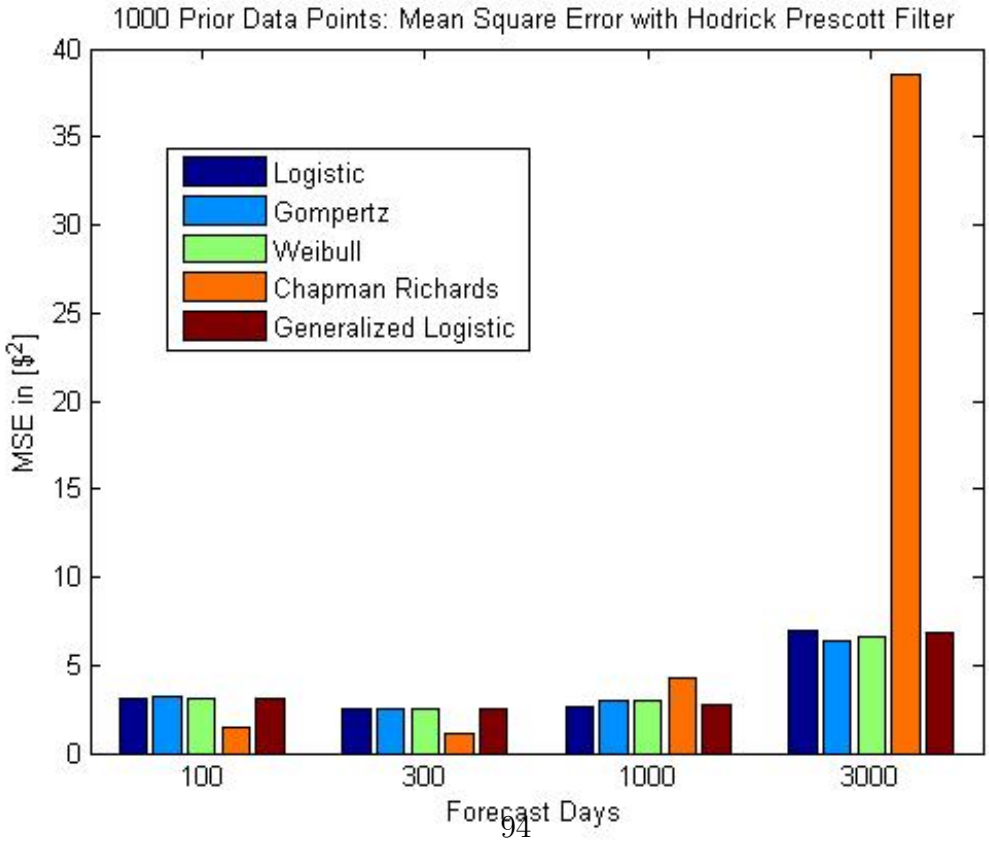


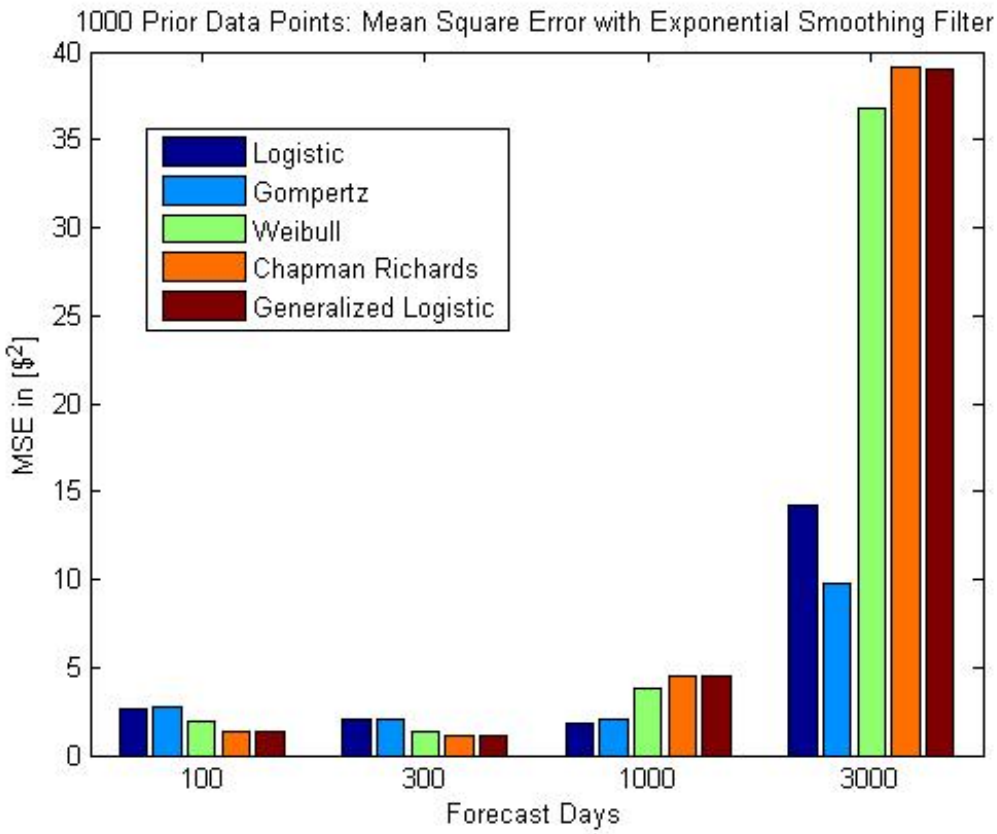

7000 Prior Data Points: Forecast Differences with Exponential Filter
Forecast Difference in [$]
10
0
-10
-20
-30
-40
-50
-60
-70
100
300
1000
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic

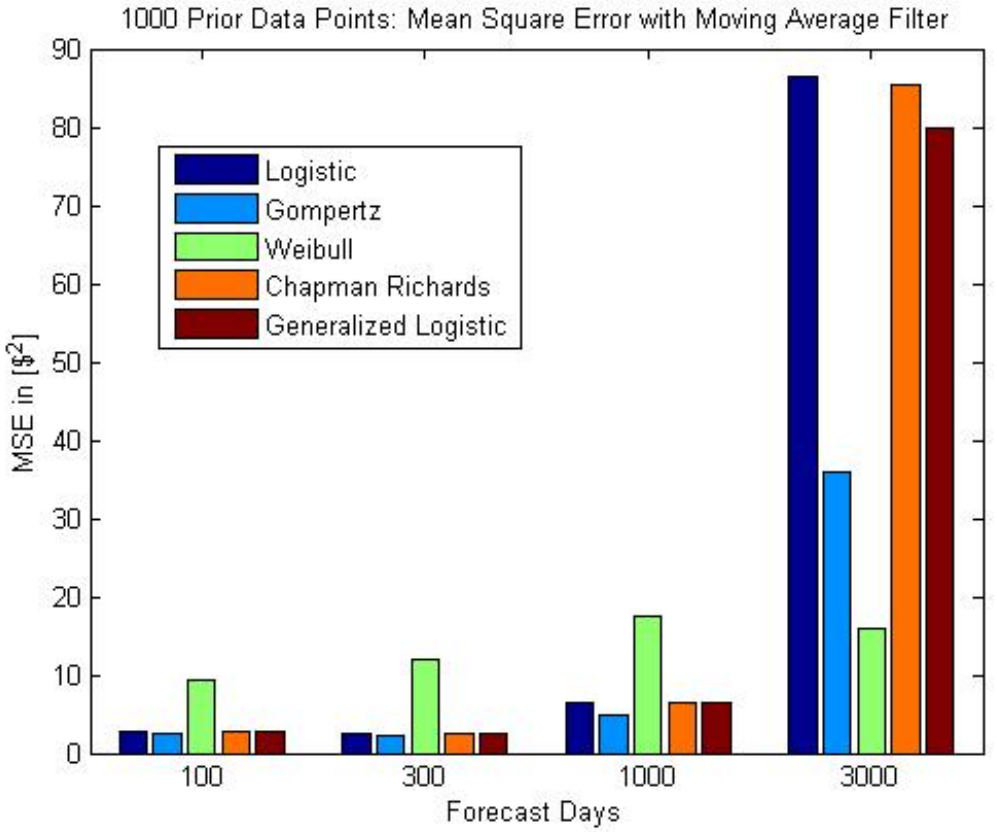

7000 Prior Data Points: Forecast Differences with Moving Average Filter
Forecast Difference in [$]
10
0
-10
-20
-30
-40
-50
-60
-70
100
300
1000
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic

5.4.6 MSE with 1000 Prior Known Days

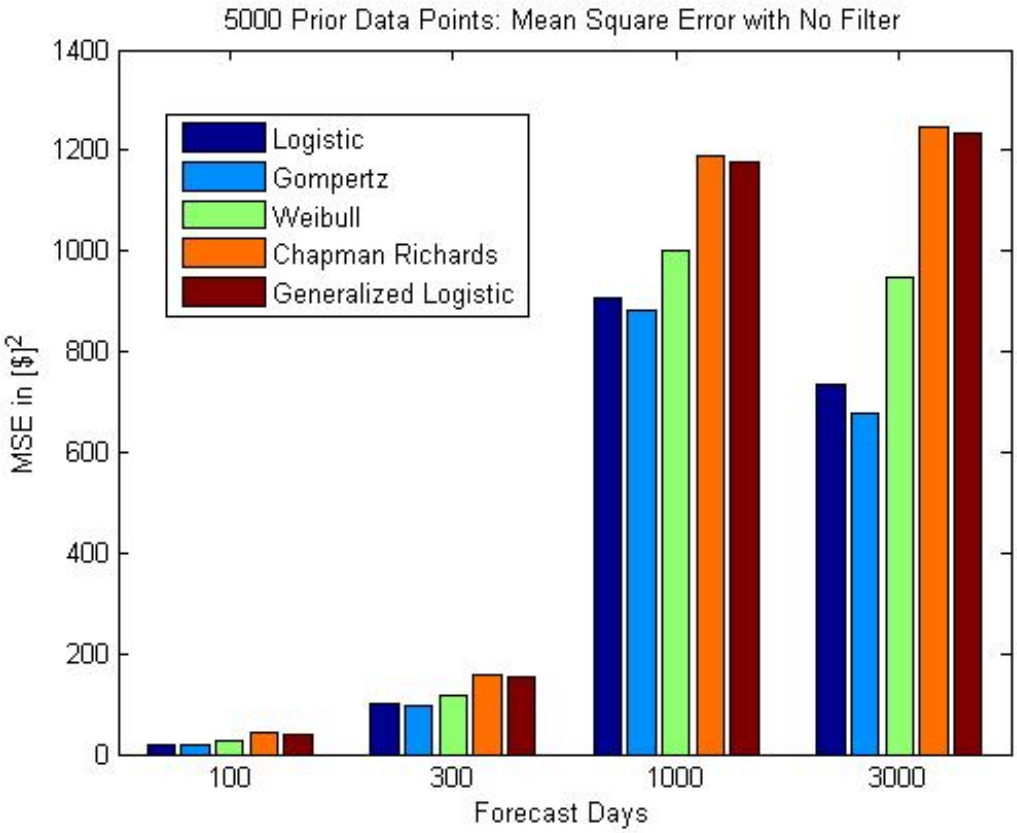


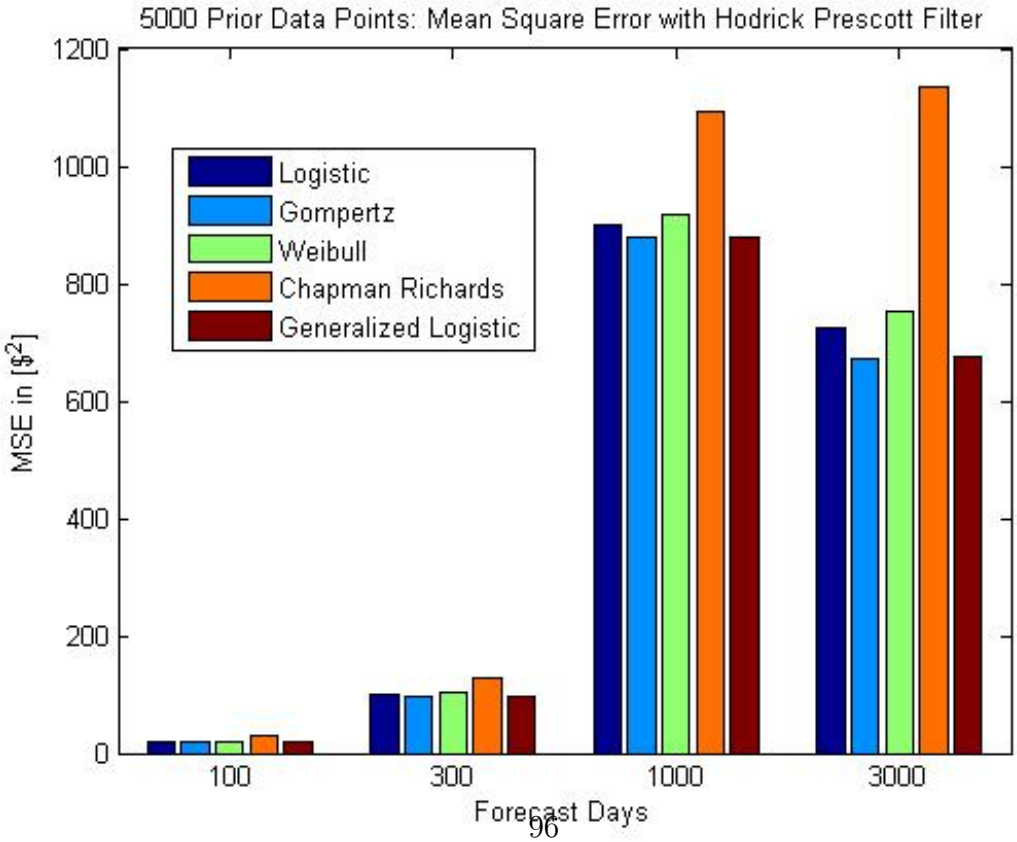


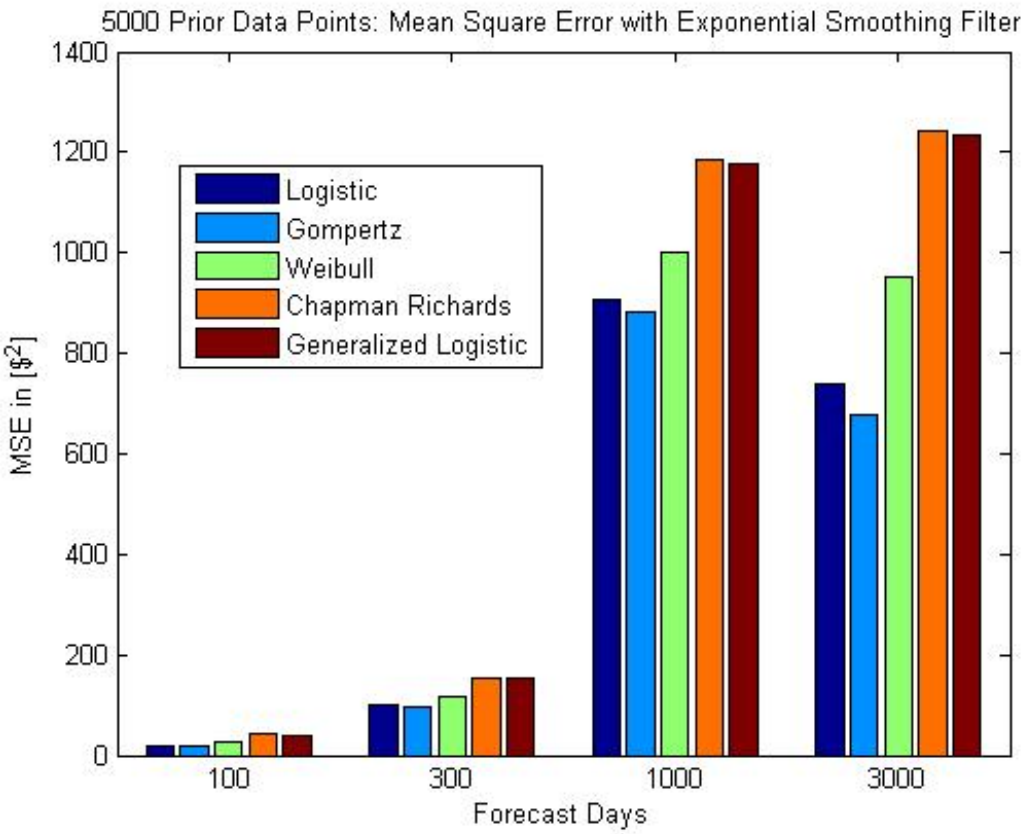

1000 Prior Data Points: Mean Square Error with Exponential Smoothing Filter
MSE in [$^2]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
0
5
10
15
20
25
30
35
40
100
300
1000
3000

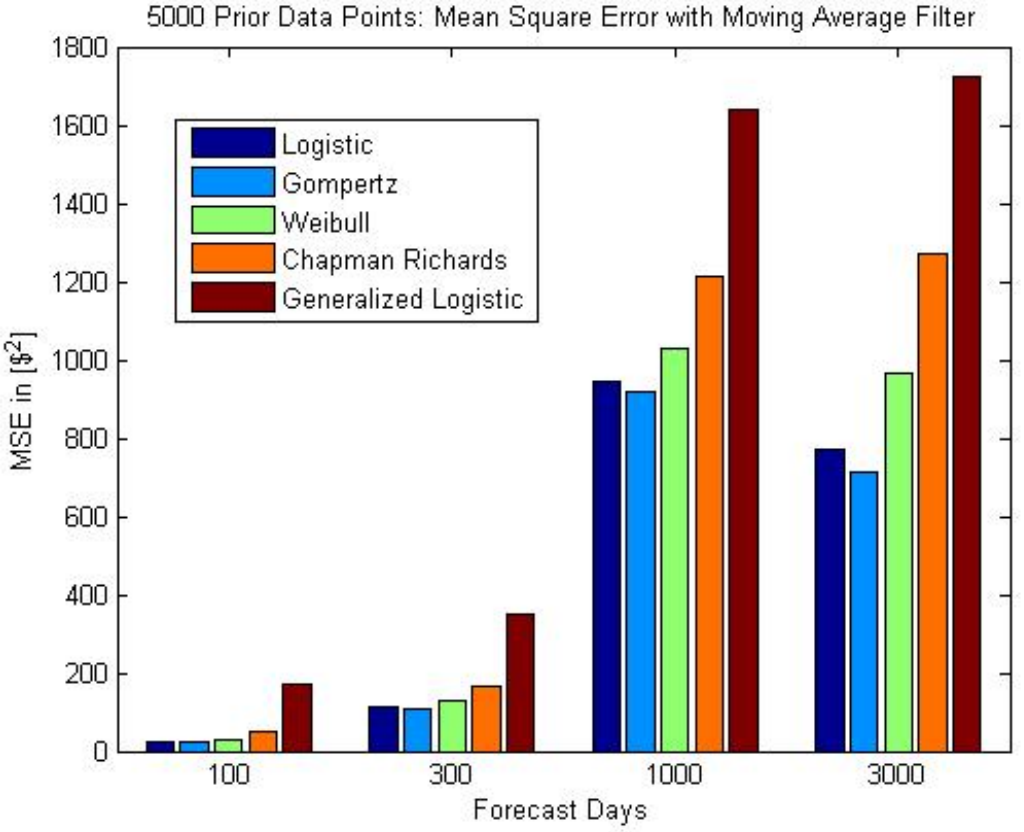

1000 Prior Data Points: Mean Square Error with Moving Average Filter
MSE in [$^2]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
0
10
20
30
40
50
60
70
80
90
100
300
1000
3000

5.4.7 MSE with 5000 Prior Known Days

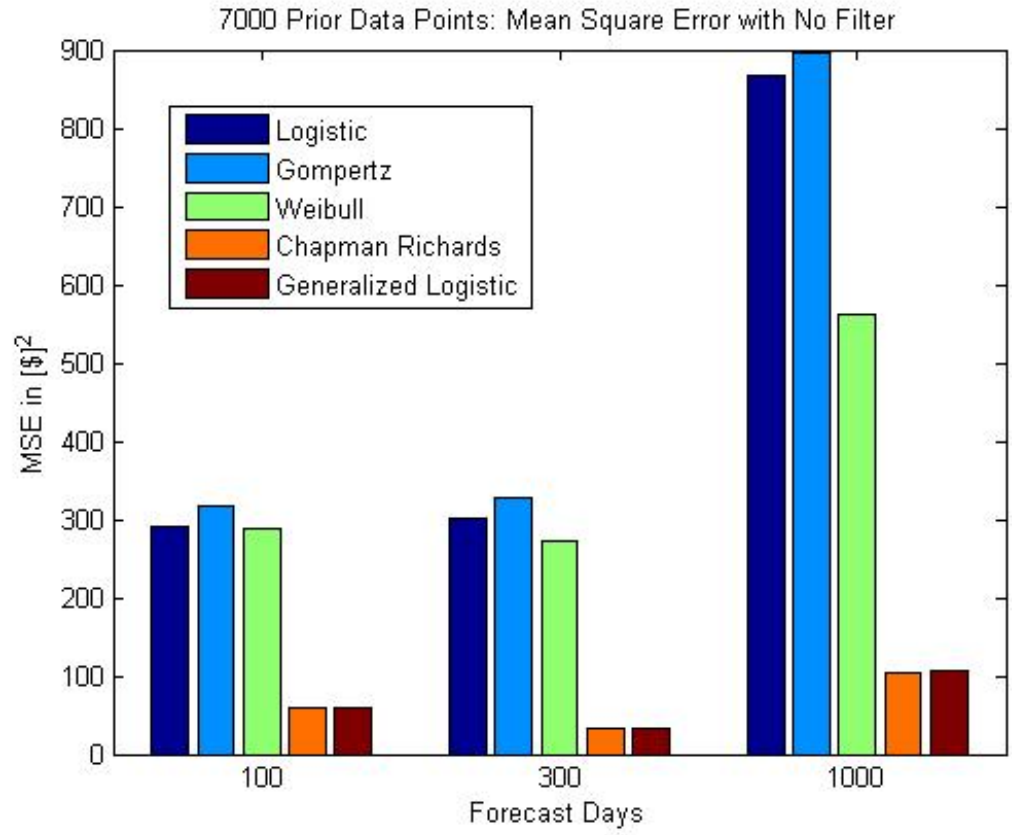


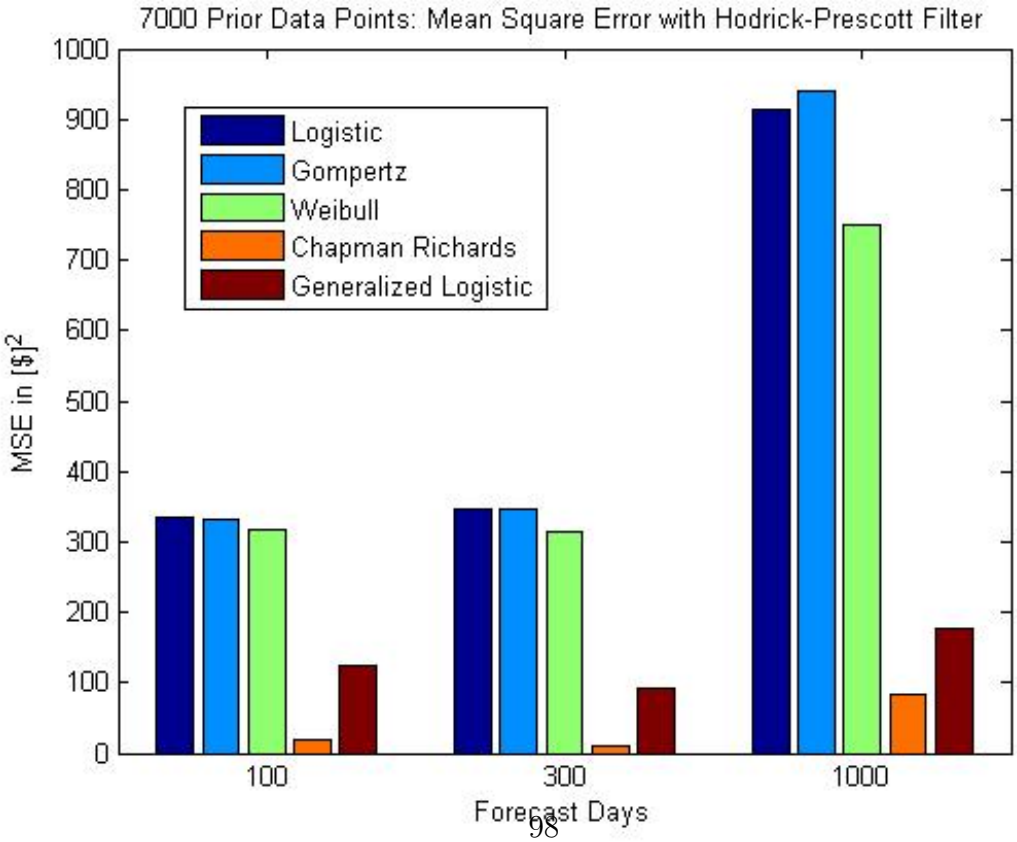


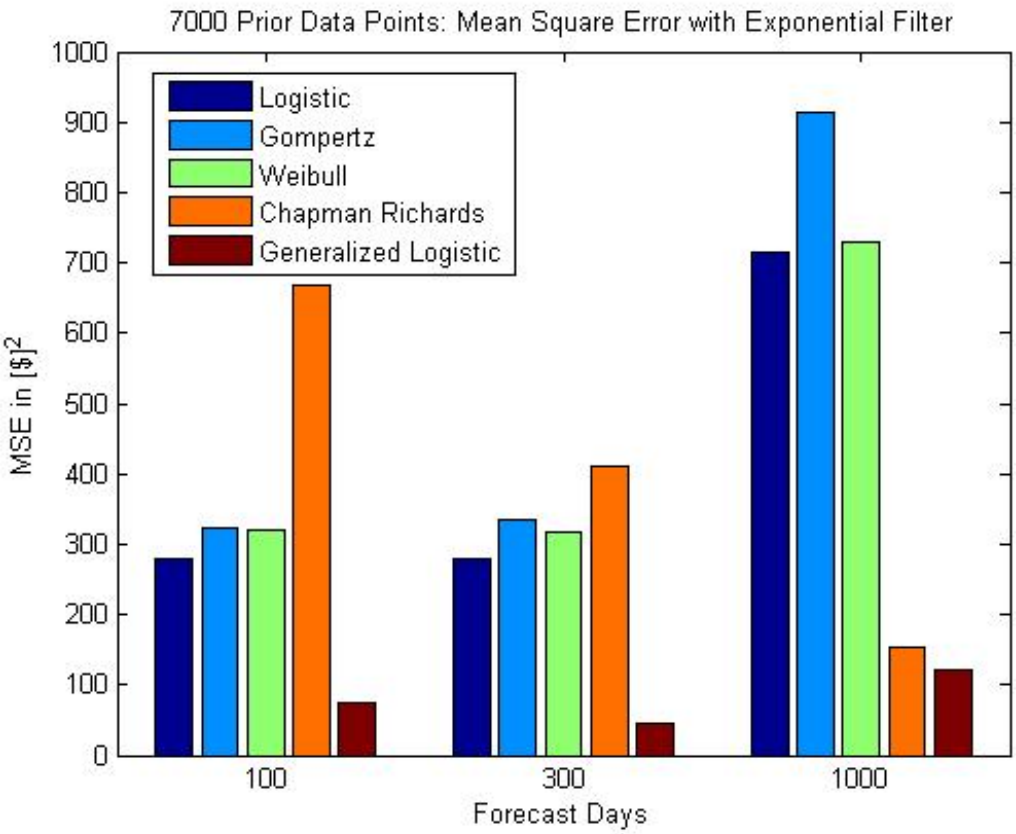

5000 Prior Data Points: Mean Square Error with Exponential Smoothing Filter
MSE in [$^2]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
0
200
400
600
800
1000
1200
1400
100
300
1000
3000

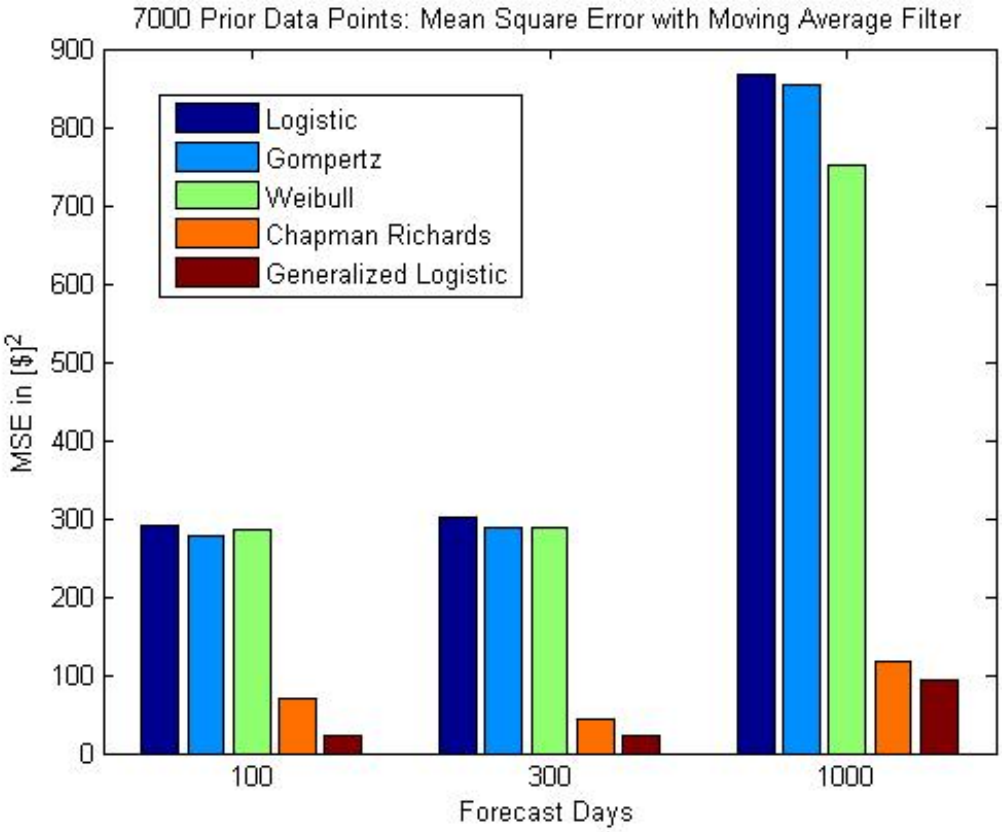

5000 Prior Data Points: Mean Square Error with Moving Average Filter
MSE in [$^2]
Forecast Days
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
0
200
400
600
800
1000
1200
1400
1600
1800
100
300
1000
3000

5.4.8 MSE with 7000 Prior Known Days






7000 Prior Data Points: Mean Square Error with Exponential Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
MSE in [$]^2
0
100
200
300
400
500
600
700
800
900
1000
100
300
1000
Forecast Days


7000 Prior Data Points: Mean Square Error with Moving Average Filter
Logistic
Gompertz
Weibull
Chapman Richards
Generalized Logistic
MSE in [$]^2
0
100
200
300
400
500
600
700
800
900
100
300
1000
Forecast Days

REFERENCES

[1] B. Baumeister, G. Peersman, *The Role of Time–Varying Price Elasticities in Accounting for Volatility Changes in the Crude Oil Market* Journal of Applied Econommetrics,**28**, Wiley Online Library, (2013), pp. 1087-1109.

[2] G. Jarne, J. Sanchez - Choliz, F. Fatas–Villafranca, *S-shaped Curves in Economic Growth. A Theoretical Contribution and an Application* Evolutionary and Institutional Economics Review,**3**, 2, Springer Japan, (2007), pp. 239-259.

[3] W. E. Griffiths, R. C. Hill, G. G. Judge, *Learning and Practice Econometrics*, John Wiley and Sons, Inc., 1993.

[4] V. Grimm, E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H. Thulke, J. Weiner, T. Wiegand, D. L. DeAngelis,*Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology*, Science, **310**, 5750, (2005), pp. 987–991.

[5] W. Henao, https://www.mathworks.com/matlabcentral/fileexchange/3972-hodrick-prescott-filter (Visited November 11, 2016)

[6] R. J. Hodrick, E. C. Prescott, *Postwar U.S. Business Cycles: An Empirical Investigation*, Ohio State University , **29**, 1, (1997), pp. 1–16.

[7] H. Kim, *Hodrick-Prescott Filter*, http://www.auburn.edu/ hzk0001/hpfilter.pdf, March 12th, 2004.

[8] P. H. Westfall, *Kurtosis as Peakedness, 1905-2014. R.I.P*, Am Stat. 2014 , 68(3), pp. 191195.

[9] S. G. Markridakis, S. C. Wheelwright, R. J. Hyndman *Forecasting: Methods and Applications 3rd Edition*, John Wiley and Sons, Inc., 1997

[10] K.Levenberg, *A Method for the Solution of Certain Problems In Least Squares*, Quart. Appl. Math. , **2**, pp.164–168.

[11] D. Marquardt, *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, SIAM J. Appl. Math. , Vol. 11, (1944), pp. 431-441.

[12] J. Nocedal, S. J. Wright *Numerical Optimization 2nd Edition*, Springer Science+Business Media, 2000.

[13] M. J. Panik, *Growth Curve Modeling: Theory and Application*, John Wiley and Sons, Inc., 2014.

[14] C. C. Pegels, *Exponential Forecasting: Some New Variations*, Theory Series, INFORMS,**15**, 5, (1969), pp. 311–315.

[15] A. Ranganathan, http://www.ananth.in/docs/lmtut.pdf, June 8th, 2004.

[16] M. O. Ravn, H. Uhlig. *On adjusting the Hodrick - Prescott filter for the frequency of observations*, Review of Economics and Statistics, **84**, (2002), pp. 371–376.

[17] M. A. Robe, J. Wallen. *Fundamentals, Derivatives Market Information and Oil Price Volatility*, Vol.36 (4), (2016), pp. 317 344.

[18] J. J. Siegel, *Stocks for the Long Run: The Definitive Guide to Financial Market Returns and Long-Term Investment Strategies*,McGraw–Hill, 1998.

[19] J. A. Schumpeter, *Business Cycles: A Theoretical, Historical and Statistical Analysis of the Capitalist Process*,McGraw–Hill, 1939.

[20] P. H. Westfall, *Kurtosis as Peakedness, 1905–2014, R.I.P* ,Am Stat., 2014 , 68(3): 191195.

[21] http://www.calstatela.edu/institutionalresearch/quick-facts (Visited April 1st, 2017)

[22] https://www.mathworks.com/help/stats/kurtosis.html (Visited March 8th, 2017)

[23] https://www.mathworks.com/help/stats/skewness.html (Visited March 8th, 2017)

[24] https://en.wikipedia.org/wiki/Gaussian_elimination#Computational_efficiency (Visited April 7th, 2017)

[25] https://en.wikipedia.org/wiki/Hessian_matrix (Visited November 18, 2016)

[26] https://en.wikipedia.org/wiki/Mean_squared_error (Visited November 9, 2016)

[27] OECD (2017), Crude oil production (indicator). doi: 10.1787/4747b431-en (Accessed on 13 April 2017)

[28] https://en.wikipedia.org/wiki/Polynomial_interpolation (Visited November 15, 2016)

[29] https://en.wikipedia.org/wiki/Skewness (Visited March 9th, 2017)

[30] https://www.google.com/finance?cid=701829 (Visited June 17th, 2016)

[31] https://finance.yahoo.com/quote/VGENX?p=VGENX (Visited November 12th, 2016)

APPENDIX A

The Logistic Model

We start with definition of the logistic differential equation [13]:

$$\frac{dY_t}{dt} = \beta Y_t \left(1 - \frac{Y_t}{Y_\infty}\right), \tag{A.1}$$

where β is the maximal growth rate when Y_t is much smaller than Y_∞, and Y_∞ is the upper limit fo the sigmoidal curve, also known as the *carrying capacity*.

We nondimensionalize (A.1) by dividing Y_t by its max value Y_∞:

$$\frac{dY_t/dt}{Y_\infty} = \beta \frac{Y_t}{Y_\infty}\left(1 - \frac{Y_t}{Y_\infty}\right)$$

and then substitute $x = Y_t/Y_\infty$ to get

$$\frac{dx}{dt} = \beta x(1-x). \tag{A.2}$$

Using the method of separation of variables, we have

$$\frac{dx}{x(1-x)} = \beta dt. \tag{A.3}$$

Now we use partial fractions decomposition to write

$$\frac{1}{x(1-x)} = \frac{A}{x} + \frac{B}{(1-x)}$$

$$A(1-x) + Bx = 1 \Rightarrow A - Ax + Bx = 1$$

which gives us a system of equations

$$\begin{cases} -A + B = 0 \\ A = 1 \end{cases}$$

and it follows that $B = 1$. Integrating the left hand side of (A.3), we obtain

$$\int \frac{1}{x(1-x)} dx = \int \frac{1}{x} dx + \int \frac{1}{1-x} dx = \ln(x) - \ln(1-x)$$

and therefore

$$\ln(x) - \ln(1-x) = \beta t + C$$

$$\ln\left(\frac{1-x}{x}\right) = -\beta t - C$$

$$\frac{1}{x} - 1 = \alpha e^{-\beta t}, \text{ where } \alpha = e^{-C}.$$

Solving for x,

$$x = \frac{1}{1+\alpha e^{-\beta t}}.$$

When we substitute $x = Y_t/Y_\infty$, we finally obtain

$$Y_t = \frac{Y_\infty}{1+\alpha e^{-\beta t}}, \quad t \geq 0. \tag{A.4}$$

APPENDIX B

The Gompertz Model

We start with definition of the Gompertz differential equation from [13]:

$$\frac{dY_t}{dt} = \beta Y_t \ln\left(\frac{Y_\infty}{Y_t}\right), \tag{B.1}$$

where β is a positive constant, and Y_∞ is the upper limit for the curve. We multiply $1/Y_\infty$ on both sides:

$$\frac{dY_t/dt}{Y_\infty} = \beta \frac{Y_t}{Y_\infty} \ln\left(\frac{Y_\infty}{Y_t}\right)$$

and then substitute $x = Y_t/Y_\infty$ to get

$$\frac{dx}{dt} = \beta x \ln\left(\frac{1}{x}\right). \tag{B.2}$$

Using the separation of variables, we solve for x as follows:

$$\begin{aligned}
\int \frac{dx}{x \ln(\frac{1}{x})} &= \int \beta dt \\
-\ln\left(\ln\frac{1}{x}\right) &= \beta t + C, \text{ where } C \text{ is a constant} \\
\ln\left(\ln\frac{1}{x}\right) &= -\beta t - C \\
\ln\frac{1}{x} &= \alpha e^{-\beta t}, \text{ and } \alpha = e^{-C} \\
\ln x &= -\alpha e^{-\beta t} \\
x &= e^{-\alpha e^{-\beta t}}
\end{aligned}$$

Finally, the substitution $x = Y_t/Y_\infty$ gives the closed form solution of the Gompertz equation:

$$Y_t = Y_\infty e^{-\alpha e^{-\beta t}}. \tag{B.3}$$

APPENDIX C

The Generalized Logistic Equation

The Generalized Logistic Equation (or sometimes called the Richard's model) is given by:

$$\frac{dY_t}{dt} = \beta Y_t \left[1 - \left(\frac{Y_t}{Y_\infty}\right)^r\right] = \beta Y_t - \frac{\beta}{Y_\infty^r} Y_t^{r+1} \tag{C.1}$$

Let $\eta = -\frac{\beta}{Y_\infty^r}$. Then

$$\frac{dY_t}{dt}\left(\frac{1}{Y_t^{r+1}}\right) = \frac{\beta}{Y_t^r} + \eta$$
$$\frac{dY_t}{dt}\left(\frac{1}{Y_t^{r+1}}\right) - \frac{\beta}{Y_t^r} = \eta$$

We use the substitution $w = \frac{1}{Y_t^r} = Y_t^{-r}$ to get

$$w' = \frac{-rY_t^{r-1}\frac{dY_t}{dt}}{(Y_t^r)^2} = \frac{-r\frac{dY_t}{dt}}{(Y_t^{(r+1)})}$$

Note that

$$-\frac{w'}{r} - \beta w = \eta$$
$$w' + r\beta w = -r\eta.$$

Using the method of integrating factor, we let $\mu(t) = e^{\int r\beta dt} = e^{r\beta t}$ and multiply both sides of the equation with $\mu(t)$:

$$e^{r\beta t}w' + r\beta e^{r\beta t}w = -r\eta e^{r\beta t}.$$

Note that the left hand side of the above equation is the derivative of $e^{r\beta t}w$. Integrating both sides of the equation with respect to t gives us

$$
\begin{aligned}
e^{r\beta t}w &= -\int r\eta e^{r\beta t}dt \\
we^{r\beta t} &= -\frac{r\eta}{r\beta}e^{r\beta t} + k, \text{ where } k \text{ is the integration constant} \\
we^{r\beta t} + \frac{\eta}{\beta}e^{r\beta t} &= k \\
e^{r\beta t}\left(w + \frac{\eta}{\beta}\right) &= k \\
w &= \frac{k}{e^{r\beta t}} - \frac{\eta}{\beta} \\
\frac{1}{Y_t^r} &= \frac{k}{e^{r\beta t}} - \frac{\eta}{\beta} = \frac{k\beta - \eta e^{r\beta t}}{e^{r\beta t}\beta}
\end{aligned}
$$

We then obtain

$$Y_t^r = \frac{e^{r\beta t}\beta}{k\beta - \eta e^{r\beta t}} \tag{C.2}$$

Letting $Y(0) = Y_0$ we have

$$
\begin{aligned}
Y_0^r &= \frac{\beta}{k\beta - \eta} \\
k\beta - \eta &= \frac{\beta}{Y_0^r} \\
k\beta &= \frac{\beta}{Y_0^r} + \eta \\
k &= \frac{1}{Y_0^r} + \frac{\eta}{\beta} = \frac{1}{Y_0^r} - \frac{1}{Y_\infty^r}.
\end{aligned}
$$

Substituting these values into (C.2):

$$
\begin{aligned}
Y_t^r &= \frac{1}{\left(\dfrac{e^{-r\beta t}}{\beta}\right)(k\beta - \eta e^{r\beta t})} \\
&= \frac{1}{e^{-r\beta t}k - \left(\dfrac{\eta}{\beta}\right)} \\
&= \frac{1}{e^{-r\beta t}\left(\dfrac{1}{Y_0^r} - \dfrac{1}{Y_\infty^r}\right) + \frac{1}{Y_\infty^r}} \\
&= \frac{1}{\dfrac{e^{-r\beta t}}{Y_0^r} - \dfrac{e^{-r\beta t}}{Y_\infty^r} + \dfrac{1}{Y_\infty^r}} \\
&= \frac{Y_\infty^r}{e^{-r\beta t}\left(\dfrac{Y_\infty^r}{Y_0^r}\right) - e^{-r\beta t} + 1} \\
&= \frac{Y_\infty^r}{1 + \alpha e^{-\beta r t}}, \text{ where } \alpha = \frac{Y_\infty^r}{Y_0^r} - 1.
\end{aligned}
$$

By taking the rth root of both sides, we obtain the generalized logistic growth function:

$$
Y_t = \frac{Y_\infty}{(1 + \alpha e^{-\beta r t})^{\frac{1}{r}}}, \ t \geq 0 \tag{C.3}
$$

APPENDIX D

The Chapman-Richards Model

The von Bertalanffy proposed the following ODE to model the growth of a biological organism [13]:

$$\frac{dY_t}{dt} = \eta Y_t^k - \gamma Y_t \tag{D.1}$$

Letting $x = Y_t^{1-k}$, its derivative is given by

$$\frac{dx}{dt} = (1-k)Y_t^{-k}\frac{dY_t}{dt}$$

which gives

$$\frac{dY_t}{dt} = \frac{1}{1-k}Y_t^k\frac{dx}{dt}. \tag{D.2}$$

By equating both forms of $\frac{dY_t}{dt}$ in (D.1) and (D.2), and dividing by Y_t^k, we solve for the closed form of von Bertalanffy equation:

$$\frac{1}{1-k}\frac{dx}{dt} = \eta - \gamma Y_t^{1-k}.$$

Substituting $x = Y_t^{1-k}$, we obtain

$$\int \frac{dx}{\eta - \gamma x} = \int (1-k)dt$$

$$-\gamma^{-1}\ln(\eta - \gamma x) = (1-k)t + C$$

$$\ln(\eta - \gamma x) = -\gamma(1-k)t - \gamma C$$

$$\eta - \gamma x = \alpha e^{-\gamma(1-k)t}, \text{ where } \alpha = e^{-\gamma C}$$

$$x = \frac{\eta}{\gamma} - \frac{\alpha}{\gamma}e^{-\gamma(1-k)t}.$$

We then substitute Y_t^{1-k} for x to get

$$Y_t = \left[\frac{\eta}{\gamma} - \frac{\alpha}{\gamma}e^{-\gamma(1-k)t}\right]^{1/(1-k)}. \tag{D.3}$$

For $t = 0$,

$$Y_0 = \left[\frac{\eta}{\gamma} - \frac{\alpha}{\gamma}\right]^{1/(1-k)} \Rightarrow \frac{\alpha}{\gamma} = \frac{\eta}{\gamma} - Y_0^{1-k} \tag{D.4}$$

Substituting (D.4) into (D.3) we get:

$$Y_t = \left[\frac{\eta}{\gamma} - \left(\frac{\eta}{\gamma} - Y_0^{1-k}\right)e^{-\gamma(1-k)t}\right]^{1/(1-k)}$$

Since $\lim_{t\to\infty} Y_t = Y_\infty = \left(\frac{\eta}{\gamma}\right)^{1/(1-k)}$,

$$Y_t = [Y_\infty^{1-k} - (Y_\infty^{1-k} - Y_0^{1-k})e^{-\gamma(1-k)t}]^{1/(1-k)}.$$

We arrive at the von Bertalanffy growth equation:

$$Y_t = Y_\infty\left[1 - \beta e^{-\gamma(1-k)t}\right]^{1/(1-k)}, \text{ where } \beta = 1 - \left(\frac{Y_0}{Y_\infty}\right)^{1-k}. \tag{D.5}$$

By rewriting the parameters of von Bertalanffy, Chapman and Richards arrive at the final equation:

$$Y_t = Y_\infty[1 - ae^{-\lambda t}]^m. \tag{D.6}$$

D.1 Data

D.1.1 No filter

Mean Square Error (MSE)

Table D.1: MSE with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.61135103	1.938620929	1.831250179	15.33841508
Gompertz	2.701222206	2.034266155	1.96325777	10.77792298
Weibull	2.120946758	1.503765706	2.59596849	31.20901334
Chapman Richards	1.293579949	1.095170952	4.522710583	39.18503707
Generalized Logistic	1.299437909	1.09723158	4.515052626	39.15932587

Table D.2: MSE with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	1.800483147	2.08533946	1.995083273	54.33160091
Gompertz	1.910437072	2.117080353	1.869612992	50.10425501
Weibull	1.194105121	2.018420201	3.063977076	68.50349107
Chapman Richards	0.156830314	2.662841411	5.957570279	82.734415
Generalized Logistic	0.886154736	5.149701411	10.71258562	98.6432342

Table D.3: MSE with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	10.69470413	24.91988161	37.49973957	664.5941422
Gompertz	10.25766667	24.11311424	36.03035963	652.1851722
Weibull	10.28566453	24.15709457	36.08489618	652.3824135
Chapman Richards	15.51644259	32.73336354	48.97689301	716.9550356
Generalized Logistic	31.33035885	53.89153253	73.56245124	785.2451826

Table D.4: MSE with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.726135938	8.74613049	8.077473188	428.4271285
Gompertz	4.742461287	10.85126113	7.297735	516.4951064
Weibull	111.8297312	136.6688715	84.47549707	866.0758109
Chapman Richards	33.23430693	55.81688673	45.82072379	1181.878319
Generalized Logistic	51.71422851	79.05724671	66.48795532	1266.361606

Table D.5: MSE with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	19.92395364	100.0127934	906.3739881	736.3018321
Gompertz	18.56187854	96.21082464	881.786391	676.3090507
Weibull	25.48713249	115.2253061	999.8426897	946.5744979
Chapman Richards	42.9318586	156.6388477	1190.154764	1248.220562
Generalized Logistic	40.51928163	151.8599956	1175.75574	1232.410253

Table D.6: MSE with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	691.8738066	450.8928624	305.0963175
Gompertz	397.0122517	501.646422	524.9902366
Weibull	2986.086057	1348.620977	1031.69712
Chapman Richards	381.4951379	425.3962578	198.942821
Generalized Logistic	748.0353833	423.6319938	157.6057573

Table D.7: MSE with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	290.3010775	301.8115534	867.3404451
Gompertz	316.3715881	327.0463339	895.4991004
Weibull	287.4328534	271.0701173	562.2934959
Chapman Richards	58.34841661	32.12436709	105.1386588
Generalized Logistic	59.39813492	33.11909998	106.9589824

Forecast Difference

Note: Negative correspond to overestimation in forecasts, while positive numbers correspond to underestimation of forecasts. Units are in U.S. dollars.

Table D.8: Forecast Difference with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-2.066310904	0.662142318	-2.480770687	6.740436053
Gompertz	-2.103747199	0.569037486	-2.910795642	5.194857981
Weibull	-1.847781119	1.18507066	-0.801632266	9.647463173
Chapman Richards	-1.438128878	1.851865219	0.10186521	10.56186421
Generalized Logistic	-1.440795843	1.849181744	0.09918166	10.55918066

Table D.9: Forecast Difference with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-0.727687614	2.075221929	3.222525593	15.40425759
Gompertz	-0.778357031	1.988727232	3.00033098	14.93770252
Weibull	-0.417377295	2.557194625	4.13370569	16.64318908
Chapman Richards	0.51198514	3.52198614	5.12198414	17.63198414
Generalized Logistic	1.1075219	4.1175229	5.7175209	18.2275209

Table D.10: Forecast Difference with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.357168478	6.86100949	6.2245945	62.65587074
Gompertz	3.282331626	6.757179044	6.017809144	62.24497855
Weibull	3.286859424	6.762092192	6.022852113	62.25121457
Chapman Richards	4.066483983	7.706484637	7.3964826	64.1364866
Generalized Logistic	5.72866624	9.36866724	9.05866524	65.79866924

Table D.11: Forecast Difference with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.385771094	4.771539117	1.542554833	19.74543194
Gompertz	2.713315502	5.229037776	2.640628169	25.04711562
Weibull	11.3661223	13.7656101	10.89390101	34.15105328
Chapman Richards	6.4856222	9.993897312	11.12329382	46.33328735
Generalized Logistic	8.209253432	11.71551963	12.84401299	48.05399221

Table D.12: Forecast Difference with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	5.191962884	14.41322567	46.62142119	-3.500302616
Gompertz	5.015369782	14.15218463	45.91605572	-7.214083317
Weibull	5.859695626	15.41079745	49.12448701	5.245876076
Chapman Richards	7.450869562	17.61110106	53.15909336	11.9986534
Generalized Logistic	7.339379866	17.4825779	53.01503836	11.85341717

Table D.13: Forecast Difference with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	16.44554473	-15.67002212	-20.19462441
Gompertz	9.755636043	-22.8203652	-28.68270755
Weibull	8.044992968	-24.57737186	-28.60646251
Chapman Richards	10.10905934	-18.38267036	-3.107529918
Generalized Logistic	17.90783962	-11.93527626	0.069921092

Table D.14: Forecast Difference with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	-18.94603776	-17.3070845	-60.31517522
Gompertz	-19.70018713	-17.95053452	-60.76854887
Weibull	-18.5230645	-15.03829552	-48.35886035
Chapman Richards	-7.84916078	-1.55461871	-26.35174228
Generalized Logistic	-7.942010079	-1.697978502	-26.56022142

D.1.2 Hodrick-Prescott Filter

Mean Square Error (MSE)

Table D.15: MSE with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.14	2.482836339	2.687134216	6.986980062
Gompertz	3.16812579	2.522960154	2.960802858	6.409337644
Weibull	3.15	2.51E+00	2.985237236	6.603449025
Chapman Richards	1.45	1.15E+00	4.326874982	38.51581804
Generalized Logistic	3.14	2.488612783	2.725313696	6.789343938

Table D.16: MSE with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.268255972	2.841676677	2.073973497	29.27206475
Gompertz	3.358160819	2.90571996	2.226282854	24.10895201
Weibull	3.341454951	2.894599571	2.226008255	23.27260654
Chapman Richards	0.145267343	2.799319487	6.285097605	84.00143871
Generalized Logistic	3.286002181	2.854144311	2.101803306	28.23959813

Table D.17: MSE with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	8.46968511	21.06393166	31.43812992	623.8057854
Gompertz	8.202362778	20.51400416	30.3008658	608.0868897
Weibull	8.06225778	20.21942417	29.67089433	597.6596448
Chapman Richards	25.30586286	46.02906226	64.51944499	761.2256188
Generalized Logistic	8.24400015	20.59944641	30.4766176	610.6552474

Table D.18: MSE with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.982988865	9.283457336	7.780115081	453.0562864
Gompertz	4.872692766	11.09661809	7.287253988	523.0280666
Weibull	5.117064623	11.7049181	7.161920067	575.2289679
Chapman Richards	3.799803948	8.893864838	7.965383031	532.8350959
Generalized Logistic	4.814877291	10.98188876	7.297876108	518.9677351

Table D.19: MSE with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	19.49550529	98.88928381	900.1640677	724.0282654
Gompertz	18.43511692	95.86178546	879.721134	672.2705621
Weibull	20.71078937	102.1319052	918.0005185	753.8243247
Chapman Richards	29.51304598	127.9355016	1091.71416	1134.78763
Generalized Logistic	18.47832367	95.98176214	880.4812841	674.030009

Table D.20: MSE with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	381.4076452	516.9280736	602.749032
Gompertz	468.706295	472.1715354	418.3656329
Weibull	360.7075314	514.0727826	523.4484477
Chapman Richards	619.0322864	439.1713901	228.3569864
Generalized Logistic	503.6905178	453.9548338	290.8366148

Table D.21: MSE with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	334.5253618	345.9903777	914.6566041
Gompertz	332.4030947	345.7409865	941.0796625
Weibull	316.0754505	314.7064835	749.4005854
Chapman Richards	17.95971655	11.3953223	81.94574406
Generalized Logistic	125.2538798	90.66611107	176.5710753

Forecast Difference

Note: Negative correspond to overestimation in forecasts, while positive numbers correspond to underestimation of forecasts. Units are in U.S. dollars.

Table D.22: Forecast Difference with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-0.008415298	-0.014301092	0.013383698	6.995860441
Gompertz	-0.015109631	-0.034434975	-0.127789896	5.921448366
Weibull	-0.004551068	-0.011185207	-0.043801143	6.089955726
Chapman Richards	-0.001395194	0.007408208	0.167018255	7.940645832
Generalized Logistic	0.004006005	0.024305323	0.285719445	8.524301828

Table D.23: Forecast Difference with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	0.091456871	0.263338377	1.764616031	22.5438809
Gompertz	0.065465285	0.212582267	1.573874101	21.71695979
Weibull	0.066122699	0.210534936	1.543841054	21.45731051
Chapman Richards	0.083246532	0.247250255	1.704444771	22.29852757
Generalized Logistic	0.077709916	0.236388956	1.663513139	22.12318071

Table D.24: Forecast Difference with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	0.616459482	1.073309389	4.055010712	35.56383512
Gompertz	0.606327059	1.048157958	3.925942575	34.56101197
Weibull	0.709596252	1.180538704	4.163899581	34.99633086
Chapman Richards	0.606763492	1.050242314	3.939958177	34.68263727
Generalized Logistic	0.607685884	1.052200636	3.94908127	34.75365463

Table D.25: Forecast Difference with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	1.743689131	3.024696582	11.5614518	17.87814296
Gompertz	2.025912464	3.410911831	12.45724421	22.10166844
Weibull	1.765377871	3.116396744	12.134019	22.97501495
Chapman Richards	1.754728259	3.039064065	11.63594002	26.73289831
Generalized Logistic	10.07404053	12.43016539	25.12719578	47.57826876

Table D.26: Forecast Difference with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	6.705109637	9.274409414	13.38242541	-29.64868752
Gompertz	7.395251896	10.19510614	15.44813183	-19.49862191
Weibull	4.234554445	6.225852605	8.681303607	-27.60409232
Chapman Richards	1.754728259	3.039064065	11.63594002	26.73289831
Generalized Logistic	9.150442648	12.03160676	17.46575222	-18.25470527

Table D.27: Forecast Difference with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	-0.188315936	-3.62324445	-22.92033175
Gompertz	1.892872138	-0.821910946	-16.40615778
Weibull	-1.777275786	-4.880709853	-20.09416812
Chapman Richards	0.13842415	-2.965114941	-17.39558626
Generalized Logistic	0.739168347	-2.021867697	-13.8706468

Table D.28: Forecast Difference with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	4.822883311	2.809086553	-9.247084424
Gompertz	-12.9152326	-18.52542613	-42.92180762
Weibull	-3.114465025	-5.227639673	-14.25369377
Chapman Richards	-2.978530658	-4.350017956	-11.59722922
Generalized Logistic	-3.210986067	-4.647619835	-11.98960708

D.1.3 Exponential Smoothing

Mean Square Error (MSE)

Table D.29: MSE with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.669944362	1.994311526	1.859967218	14.23897108
Gompertz	2.762401552	2.094862134	2.036655355	9.714757291
Weibull	1.94167264	1.366310537	3.85164855	36.77015075
Chapman Richards	1.312027382	1.101692763	4.499461711	39.10669887
Generalized Logistic	1.329418502	1.107920586	4.477478975	39.03241513

Table D.30: MSE with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.365195256	2.316449902	1.748999339	44.45182348
Gompertz	2.010209922	2.156101118	1.825256389	48.56344111
Weibull	1.250578778	2.024860438	2.999704891	68.05853326
Chapman Richards	1.111565046	2.042198386	3.303800886	70.08249952
Generalized Logistic	1.22667632	2.054372709	3.182756702	69.30544503

Table D.31: MSE with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	10.73385612	24.97979318	37.5764782	664.9106837
Gompertz	10.29517381	24.17068621	36.1037297	652.5044372
Weibull	10.36463391	24.29087237	36.29633479	653.9414169
Chapman Richards	10.37475404	24.31837733	36.37521187	654.9505867
Generalized Logistic	10.35296661	24.27712385	36.29729237	654.2356193

Table D.32: MSE with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.745947608	8.779146568	8.076462059	428.3595572
Gompertz	4.798205752	10.93308602	7.319144661	514.916165
Weibull	4.726824011	10.92323258	7.153879283	550.1415442
Chapman Richards	3.838510766	8.984984786	7.922412924	442.433408
Generalized Logistic	3.822858374	8.949865613	7.948129622	440.0709937

Table D.33: MSE with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	20.03521194	100.2625705	907.3350164	737.6272655
Gompertz	18.67207357	96.46006556	882.7315455	677.5116702
Weibull	25.80770342	115.9158129	1002.485035	950.6577789
Chapman Richards	41.97263962	154.7956104	1184.825706	1242.432728
Generalized Logistic	40.71034937	152.2234075	1176.802899	1233.548805

Table D.34: MSE with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	308.4063305	564.8859472	769.7007274
Gompertz	398.0235938	501.2133996	523.7149649
Weibull	348.3991981	521.6327645	546.9904746
Chapman Richards	308.7844532	564.5352978	768.229178
Generalized Logistic	306.4900621	566.0357431	772.4602557

Table D.35: MSE with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	280.6887896	279.3134455	715.4719307
Gompertz	324.4042263	335.9050986	912.850504
Weibull	320.3188007	316.8772496	730.27608
Chapman Richards	667.8927667	409.4925154	153.720457
Generalized Logistic	74.66805199	45.49988629	121.2489878

Forecast Difference

Note: Negative correspond to overestimation in forecasts, while positive numbers correspond to underestimation of forecasts. Units are in U.S. dollars.

Table D.36: Forecast Difference with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-2.262918868	0.262658636	-3.874716821	2.694651597
Gompertz	-2.274469883	0.226360757	-4.149266786	0.450379975
Weibull	-2.269373774	0.233368263	-4.182065623	-0.518034138
Chapman Richards	-1.508252049	1.781515189	0.031514465	10.49151346
Generalized Logistic	-2.264591926	0.257344768	-3.914986438	2.404411214

Table D.37: Forecast Difference with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-1.267141611	1.296587923	1.601835493	12.19785897
Gompertz	-1.298707999	1.234099357	1.359918685	11.09998192
Weibull	-1.29375167	1.239142102	1.346978759	10.85696325
Chapman Richards	0.593925679	3.603923635	5.203921536	17.71392154
Generalized Logistic	-1.273431493	1.284056306	1.553519077	11.99542495

Table D.38: Forecast Difference with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.970745734	6.377748151	5.433157246	61.34159653
Gompertz	2.917629416	6.295763722	5.226217406	60.74600871
Weibull	2.889211568	6.25069238	5.103922385	60.32833679
Chapman Richards	5.160097401	8.799282862	8.489150847	65.22915472
Generalized Logistic	2.925944574	6.308553712	5.258774108	60.84562596

Table D.39: Forecast Difference with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.473843683	4.893205285	1.838675494	21.42038939
Gompertz	2.750559053	5.276063558	2.731944806	25.3788115
Weibull	2.826655137	5.415999675	3.224445145	28.25414299
Chapman Richards	2.410966313	4.804488294	1.677645871	30.21645533
Generalized Logistic	2.733772539	5.253137994	2.680286686	25.16638851

Table D.40: Forecast Difference with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	5.138138884	14.33941019	46.45675916	-4.096021363
Gompertz	4.998690285	14.12833484	45.85929082	-7.445213018
Weibull	5.290685464	14.55338452	46.9179543	-3.004556288
Chapman Richards	6.327918276	16.26147526	51.48929617	10.26514451
Generalized Logistic	5.004385544	14.1366081	45.88111895	-7.327423602

Table D.41: Forecast Difference with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	9.256473656	-23.76144198	-32.31856511
Gompertz	11.64395473	-20.48367338	-24.29960552
Weibull	8.778315117	-23.6875907	-26.99182023
Chapman Richards	15.11971468	-16.08340281	-9.501655126
Generalized Logistic	12.56241707	-18.86380352	-13.63784019

Table D.42: Forecast Difference with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	-20.22548227	-18.46927024	-60.87177879
Gompertz	-20.18462503	-18.55216347	-61.92458382
Weibull	-19.55873347	-17.04703306	-55.28220878
Chapman Richards	-1.451166873	4.865345005	-19.9106586
Generalized Logistic	-11.97192188	-6.13246367	-31.49059789

D.1.4 Moving average

Mean Square Error (MSE)

Table D.43: MSE with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.783553787	2.479959358	6.512169118	86.52237742
Gompertz	2.538221027	2.18670505	4.823457284	35.82214713
Weibull	9.278385675	12.00970097	17.55032478	15.8358527
Chapman Richards	2.781715323	2.477492061	6.491756386	85.34189969
Generalized Logistic	2.777375901	2.471967544	6.441968335	79.92021265

Table D.44: MSE with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	4.412226784	3.711932122	3.627867348	14.43022009
Gompertz	4.569694085	3.856002982	4.177404381	9.543539458
Weibull	3.337214839	2.866321033	1.862662718	45.71176139
Chapman Richards	2.043808258	2.26078458	2.596196972	64.80701341
Generalized Logistic	2.121283323	2.287190281	2.548615182	64.37108156

Table D.45: MSE with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	12.16433469	26.98542795	39.71706267	670.0284523
Gompertz	11.63760329	26.02763744	37.96275961	655.1270863
Weibull	13.73913945	29.72743468	44.38069737	698.7482607
Chapman Richards	13.75559166	29.77766389	44.49996193	699.1445167
Generalized Logistic	13.75537277	29.77725868	44.49923213	699.1406427

Table D.46: MSE with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	5.142166985	11.01945988	8.061311933	433.0551156
Gompertz	6.265782064	13.16580912	7.729422718	514.7802866
Weibull	5.665775838	12.20221223	7.552815115	528.3374692
Chapman Richards	5.166742234	11.07522786	8.026588796	437.5257159
Generalized Logistic	46.75435489	72.82795307	60.83293425	1244.298544

Table D.47: MSE with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	26.2453239	112.7899884	944.7882963	772.9001994
Gompertz	24.78999089	108.9637432	920.4360407	711.3290176
Weibull	31.56192177	126.5662578	1027.877097	964.5900265
Chapman Richards	47.76848652	164.9888151	1211.615384	1270.993694
Generalized Logistic	169.9131162	351.0156976	1639.711449	1725.989964

Table D.48: MSE with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	691.8738066	450.8928624	305.0963175
Gompertz	807.7989969	446.1882789	216.6611512
Weibull	653.4003697	447.9109943	285.4299857
Chapman Richards	769.1785331	439.856513	185.4372621
Generalized Logistic	740.814953	433.2490506	174.6319294

Table D.49: MSE with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	290.3010775	301.8115534	867.3404451
Gompertz	276.7096807	287.5778911	854.2818288
Weibull	285.1606003	287.815095	750.71414
Chapman Richards	70.31857887	42.19105593	118.0810847
Generalized Logistic	22.82271923	23.42658721	93.45590279

Forecast Difference

Note: Negative correspond to overestimation in forecasts, while positive numbers correspond to underestimation of forecasts. Units are in U.S. dollars.

Table D.50: Forecast Difference with 1000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-2.219922966	-0.040314981	-6.460394281	-19.48797949
Gompertz	-2.127300056	0.145172059	-5.616302599	-10.86211149
Weibull	-3.90147552	-2.790527773	-6.972131628	3.473705776
Chapman Richards	-2.219209188	-0.038662229	-6.450667057	-19.29816221
Generalized Logistic	-2.21758517	-0.035083929	-6.425831193	-18.24393338

Table D.51: Forecast Difference with 2000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	-1.611344652	0.761044237	0.325405529	8.97215896
Gompertz	-1.6578925	0.670213133	-0.047254275	6.722920973
Weibull	-1.266574622	1.410430382	2.374244468	14.65125837
Chapman Richards	-0.756759184	2.251666205	3.851551279	16.36155127
Generalized Logistic	-0.787265749	2.218110837	3.817564271	16.32756407

Table D.52: Forecast Difference with 3000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	3.57237535	7.064050431	6.395321345	62.78925595
Gompertz	3.487488344	6.944972134	6.15322809	62.29488477
Weibull	3.813352741	7.384042533	6.954058839	63.65207079
Chapman Richards	3.815514235	7.390558355	6.967696968	63.66090836
Generalized Logistic	3.815480143	7.390509924	6.967612977	63.66079905

Table D.53: Forecast Difference with 4000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	2.792258756	5.164854658	1.871579972	19.66136388
Gompertz	3.102266284	5.595267605	2.897681306	24.65748832
Weibull	2.930810017	5.413530447	2.768938023	25.82768393
Chapman Richards	2.799558686	5.177488234	1.914514119	20.0053434
Generalized Logistic	7.847033254	11.34706546	12.47274528	47.68267735

Table D.54: Forecast Difference with 5000 Prior Known Days

Forecast Days:	100	300	1000	3000
Logistic	5.873120348	15.10541143	47.34919267	-2.693802561
Gompertz	5.701733247	14.84984782	46.64949736	-6.41807983
Weibull	6.446343308	15.97029499	49.56036761	5.264815663
Chapman Richards	7.898629238	18.05334892	53.59800879	12.43745346
Generalized Logistic	14.08575591	24.26514699	59.82483848	18.66482709

Table D.55: Forecast Difference with 6000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	16.44548575	-15.67008809	-20.19472133
Gompertz	18.79869837	-12.52372989	-12.94842988
Weibull	15.70444051	-16.16262837	-17.36206608
Chapman Richards	18.08976838	-12.91439742	-5.731746716
Generalized Logistic	17.61827939	-13.00579795	-3.356993845

Table D.56: Forecast Difference with 7000 Prior Known Days

Forecast Days:	100	300	1000
Logistic	-18.94607399	-17.30712349	-60.31522455
Gompertz	-18.56316153	-16.93658111	-60.36809333
Weibull	-18.73304387	-16.52367769	-56.1086117
Chapman Richards	-8.768779743	-2.632819952	-27.59889037
Generalized Logistic	0.760684731	7.132579374	-17.59355243

APPENDIX E

MATLAB Code

E.1 Filters

E.1.1 Moving Average

If the time point has less data points than the moving average length, take the average of the prior data points.

data: Raw data. MALength: Length of moving average.

```
function [ result ] = meanAvg( data, MALength )


t = length(data);

for i = 1:t

if i == 1

result(1)= data(1);

end


if i <= MALength

A = data(1:i);

result(i) = 1/i * sum(A);

end


if i > MALength

k = i-MALength;
```

```
A = data(k:i);
result(i) = 1/MALength * sum(A);
end
end
```

*Note: MATLAB has its own built in function, "movmean."

E.2 Exponential Filter

This MATLAB code outputs columns of various weight parameters. The number of columns and the size of the weight parameter can easily be changed as per the user's specifications.

```
function [ result ] = alpha( data, PriorDataLength)

for i = 1:PriorDataLength
for k = 1:9
if i == 1
result(1, k)= data(1);
end

if i ~= 1
result(i, k) = result(i-1, k) + k * 0.1* (data(i-1) - result(i-1, k)) ;
end
end
end
```

*Note: MATLAB has its own built in function, "tsmovavg" in the Financial Toolbox.

E.2.1 The Hodrick-Prescott Filter

y : Unfiltered data w: lambda constant for Hodrick-Prescott Filter

```
function [s,desvabs] = hpfilter(y,w,plotter)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Author: Wilmer Henao    wi-henao@uniandes.edu.co
%   Department of Mathematics
%   Universidad de los Andes
%   Colombia
%
%   Hodrick-Prescott filter extracts the trend of a time series, the
%   output is not a formula but a new filtered time series.  This
%   trend can be  adjusted with parameter w; values for w lie usually
%   in the interval  [100,20000], and it is up to you to use the one
%   you like, As w approaches infty, H-P will approach a line.  If
%   the series doesn't have a trend p.e.White Noise, doing H-P is
%   meaningles
%   [s] = hpfilter(y,w)
%   w = Smoothing parameter (Economists advice: "Use w = 1600 for
%   quarterly data")
%   y = Original series
```

```
%   s = Filtered series
%   This program can work with several series at a time, as long
%   as the number of series you are working with doesn't exceed
%   the number of elements in the series + it uses sparse matrices
$   which improves speed and performance in the longest series
%
%   [s] = hpfilter(y,w,'makeplot')
%   'makeplot' in the input, plots the graphics of the original
%   series against the filtered series, if more than one series
%   is being considered the program will plot all of them in
%   different axes
%
%   [s,desvabs] = hpfilter(y,w)
%   Gives you a mesure of the standardized differences in absolute
%   values  between the original and the filtered series.  A big
%   desvabs means  that the series implies a large relative
%   volatility.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
if nargin < 2
error('Requires at least two arguments.');
end
```

```
[m,n] = size (y);

if m < n

y = y';     m = n;

end

d = repmat([w -4*w ((6*w+1)/2)], m, 1);

d(1,2) = -2*w;      d(m-1,2) = -2*w;

d(1,3) = (1+w)/2;   d(m,3) = (1+w)/2;

d(2,3) = (5*w+1)/2; d(m-1,3) = (5*w+1)/2;

B = spdiags(d, -2:0, m, m);    %I use a sparse version of B,

because when m is large, B will have many zeros

B = B+B';

s = B\y;


if nargin == 3

t = size(y,2);

for i = 1:t

figure(i)

plot(s(:,i),'r');   grid on;   hold on;   plot(y(:,i));

title(['Series #',num2str(i)]);

end

end

if nargout == 2

desvabs = mean(abs(y-s)./s);
```

```
end

toc
```

*Note: MATLAB has its own built in function, "hpfilter" in the Econometrics Toolbox.

E.3 Fitting

E.3.1 Polynomial Fit

parameter = polyfit(x value,y value raw data,polynomial order)

y values of fit = polyval(parameter, x value)

E.3.2 The Levenburg-Marquart Algorithm

[beta,R,J] = nlinfit(days,raw data, function ,test parameters);

E.4 MSE and Difference of Forecast

This function calculates the mean square error (MSE) and difference between the forecasted final time point and actual data point.

a : parameters from LM algorithm func: function PriorDataLength: Number of known time points ydata: Entire raw data set

```
function [ mse,  diff ] = results( a, func, PriorDataLength , ydata)


for i = 1:4

%% change i = 1:3 if forecast data exceeds dataset

if i == 1 %%MSE and difference calculation for 100 future points
```

```
A1 = ydata(1: (PriorDataLength + 100));

B2 = transpose (func(a,(1: (PriorDataLength + 100))));

A =  A1( (PriorDataLength + 1):(PriorDataLength + 100) );

B =  B2( (PriorDataLength + 1):(PriorDataLength + 100) );

C = (minus(A,B)).^2;

mse(i) = sum( C )/100;

diff(i) =  ydata(PriorDataLength + 100)

- func(a,(PriorDataLength + 100)) ;

end


if i == 2 %%MSE and difference calculation for 300 future points

A1 = ydata(1: (PriorDataLength + 300));

B2 = transpose (func(a,(1: (PriorDataLength + 300))));

A =  A1( (PriorDataLength + 1):(PriorDataLength + 300) );

B =  B2( (PriorDataLength + 1):(PriorDataLength + 300) );

C = (minus(A,B)).^2;

mse(i) = sum( C )/300;

diff(i) =  ydata(PriorDataLength + 300)

- func(a,(PriorDataLength + 300)) ;

end


if i == 3  %%MSE and difference calculation for 1000 future points

A1 = ydata(1: (PriorDataLength + 1000));
```

```
B2 = transpose (func(a,(1: (PriorDataLength + 1000))));

A =  A1( (PriorDataLength + 1):(PriorDataLength + 1000) );

B =  B2( (PriorDataLength + 1):(PriorDataLength + 1000) );

C = (minus(A,B)).^2;

mse(i) = sum( C )/1000;

diff(i) =  ydata(PriorDataLength + 1000)

- func(a,(PriorDataLength + 1000)) ;

end


if i == 4  %%MSE and difference calculation for 3000 future points

A1 = ydata(1: (PriorDataLength + 3000));

B2 = transpose (func(a,(1: (PriorDataLength + 3000))));

A =  A1( (PriorDataLength + 1):(PriorDataLength + 3000) );

B =  B2( (PriorDataLength + 1):(PriorDataLength + 3000) );

C = (minus(A,B)).^2;

mse(i) = sum( C )/3000;

diff(i) =  ydata(PriorDataLength + 3000)

- func(a,(PriorDataLength + 3000)) ;

end



end
```

```
mse = transpose(mse);

diff = transpose(diff);


end
```