Multi-Input Modeling using Sparsity Constraints with Global-Local Approximation

Christopher R. Wirz

crwirz@gmail.com

Masters of Science Project

Department of Mechanical & Aerospace Engineering

University at Buffalo

Buffalo, NY 14260

August 2008

Abstract

This research demonstrates effective methods in achieving a sparse solution to model development. Sparse solutions can greatly aid in signal approximation and prediction by capturing the dominant characteristics of the signal - even for higher dimensionality. This means a signal and its prediction can be reconstructed with less noise, so long as only a limited selection of functions are used to model the signal. The enforcement of this limit is known as sparsity - and it is achieved through various optimization routines in convex programming. This research begins by introducing the Least Squares technique to curve fitting and model development. This approach can adequately approximate a signal without understanding the underlying dynamics which produced that signal. The Least Squares solution provides the minimal 2-norm of the residual error both analytically and as an optimization routine. The optimization routine is further investigated to not only minimize the error of the model, but also the number of functions used to build the model. The algorithm may select from a wide array of functions and combinations of functions as found in a dictionary. By posing the problem in this manner, a tradeoff is seen between number of functions in the model, and the error of the approximation and prediction. Once a sufficient number of functions has been included in the model, the error minimization is improved only marginally by including any more basis functions. At this point, the tradeoff has occurred and an adequate reconstruction of the signal has been formed. Further, the ability to form a sparse solution is enhanced through the exploration of Global-Local Approximation Mapping. This research uses many different examples, varying in complexity and dimensionality - with real applications in output reconstruction, terrain mapping, and even financial analysis. Each example has its advantages and limitations utilizing model development with sparsity constraints.

Acknowledgements

I would like to thank my advisor, Dr. Puneet Singla for helping me to explore my interests in systems identification through this research. I would also like to thank Dr. Venkat Krovi for his influence early on in my graduate career. Also, I would like to thank the rest of my advisory committee. We are very fortunate to have such talented faculty at the University at Buffalo. Their guidance and support has encouraged me greatly, and has taught me so much.

Further, I would like to thank the other members of the LAIRS lab, and those who have graduated before me. Special thanks to the following groups that played a part in my graduate education: the MAE-GSA and its executive board, the ARM Lab, UB-ASME, NAVEODTECH-DIV, JNE Consulting, and Lockheed Martin Systems Integration. Special thanks to those fellow graduate students who deserve proper mention: Aaron Teng, Yao Wang, Hao Su, Quishi Fu, Pat Miller, Zhongyuan Lu, Kurt Bessel, Sean Burley, Thomas Leach, and many others who know who they are but probably have heard enough about my research to not have to read it.

Finally, it is a pleasure to acknowledge my parents, friends, and my brother.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, there have been developments in methods of signal representations. Signals can now be represented as a collection (a dictionary) of parameterized waveforms (h_i) with coefficients (x_i) such that a signal (s) can be decomposed as $s = \sum_{i=1}^{n} x_i h_i + \mathbf{r}$, where $\mathbf{r}$ is the residual (the difference between the measurement and the approximation). A better approximation will therefore minimize the absolute sum of the residual. Signals can vary in dimensionality - having different number of inputs. For example, modeling the altitude of an aircraft with respect only to time, would be a one dimensional approximation. If that same altitude, of that same aircraft, was modeled with respect to thrust, angle of attack, gross weight, and pressure altitude, a four dimensional model would be used to form the approximation. An important assumption in this research is that even though many systems have multiple inputs AND multiple outputs, each approximation formed will be in terms of one output only. This is due to the assumption that if two outputs have no effect on each other, they can be approximated separately. If one output does have an effect on another, it can be used as an input when forming an approximation model for the other output.

The second important assumption of this research is that the underlying dynamics of the signal is unknown. The focus of this study is to simply reproduce the signal using any number of basis functions available. The advantage of this type of analysis is its extensibility in solving a wide range of problems. Of course, there is a disadvantage to this approach as well. A 'complicated' signal is more difficult to approximate, and when a model is formed successfully, it will consist of many parameters. This takes computational time and in some cases captures the noise of the signal rather than the signal itself. The challenge then becomes developing a model that captures the main components of the signal, thus limiting the complexity of the model while achieving a reasonably minimized residual squared sum.

1.1 Problem

To create a model used in a signal approximation, an algorithm [1] must select the proper components that will minimize the residual. Since it is assumed that the main components of the signal are unknown, the algorithm must pick from a wide variety of candidate components. These candidate components are known as basis functions (which will form the basis of the approximation model) and the collection of the basis functions is known as a dictionary function. Dictionaries can contain any number or variety of function types, and multiple dictionaries can be combined to form a larger dictionary of candidate basis functions. Due to the high availability of dictionaries [2], such as orthogonal polynomials, sinusoids, radial basis functions, step functions, wavelets, chirplets, and many others, approximation algorithms can have more basis functions than the actual signal has measurement points. This, and the assumption that a dictionary can have more basis functions than needed to compose the original signal, makes an 'over-complete' dictionary.

The problem then involves selecting the best basis functions, from that dictionary, to generate the model that can reconstruct the signal. This means eliminating much of the dictionary. By forcing the algorithm to utilize only the most dominant basis functions, a principle known as 'sparsity' is enforced.

1.2 Objective

This research will examine the principle known as sparsity, which through various methods automatically selects the fewest number of basis functions that will form an approximation, given certain tolerances in minimization of the residual. The tradeoff between having fewer basis functions or having a smaller residual will be examined for systems with varying number of inputs. When an algorithm forms an approximation model, a certain weight is given to every function in the dictionary. In some cases, the weight is zero, showing no contribution of this function, but more often than not, without any constraints, every basis function is at least assigned some weight. Therefore, a convenient way to minimize both the residual and the number of coefficients is through an optimization routine, which can change the emphasis placed on minimizing the residual or the sum of the coefficients. Including minimization of the sum of the coefficients will force the algorithm to assign more zero weights to the basis functions when forming the approximation model.

By limiting the number of basis functions, the dominant functions used to reproduce the signal are selected. When too few basis functions are used, the residual sum grows rapidly. It is at this point that a tradeoff is found. Increasing the number of basis functions does little for minimizing the residual or improving the model approximation, and decreasing the number of basis functions shows a growth in the residual and neglects to capture certain features of the signal. The systems to be analyzed include, but are not limited to, benchmark functions, terrain maps, and stock market models.

To further reduce the number basis functions needed, methods of global-local mapping, using weighted regional approximations, will be implemented. This adds a new complexity to the observable tradeoffs. Of course, the goal is still to minimize the global residual (through 2-norm minimization), but now the number of basis functions used is decreased simply by increasing the number of regional approximations. Since the minimization routine is still used, the emphasis placed on minimizing the sum of the coefficients does not need to be as significant. In this research, our dictionaries, unless otherwise mentioned, are $3 \cdot (20! \cdot n_{samples})^{n_{dimensionality}}$ basis functions in size.

Through the study of these two major methods, the accuracy of the model, the construction of the model, and the limitations of the model will be discussed for each example ranging in dimensionality and complexity.

1.3 Outline

This research forms the foundation of residual minimization through optimization routines [1], develops variations in minimization problems that leads to sparsity, elaborates on further model approximation improvements through global-local approximation mapping, and reviews the achievements and application of this research.

In chapter 2, the foundation of error minimization will be discussed, and the least squares [3] process will be developed. Using various examples, global models will be made and the ability of each to approximate the signal and form predictions of the signal behavior will be analyzed. To achieve a basic understanding of the limitations of the 2-norm minimization posed by the least squares, a standard dictionary will be used.

In chapter 3, an over-complete dictionary will be utilized, and sparse solutions will be formed. The tradeoff between lower number of coefficients (cardinality) and low 2-norm error will be observed. Using convex optimization routines [4], emphasis constraint values for the minimization problem will be investigated.

In chapter 4, regional approximations of signals will be investigated. These regional approximations will be globally mapped and combined using smoothing functions [5]. The tradeoff between lower global 2-norm error and lower computational time will be observed - and compared to the former global models.

Chapter 5 will review the methods of the previous three chapters, and compare the advantages for each in the proposed examples. Further examples will be provided to illustrate the effectiveness of the methods developed - as well as opening the door for further research and concluding remarks.

Chapter 2

Least Squares Method

2.1 Introduction

The concept of Least Squares [6] [7] is introduced in this chapter. Least Squares has many applications, including but not limited to curve fitting, model realization, and parameter identification. It is a procedure commonly used for finding the best-fitting curve to a given set of points. If a signal has a normally distributed error, the least squares solution represents the most likely solution.

The Least Squares method minimizes the sum of the squares of the residuals (the perpendicular distance from the points to the curve). This makes the solution continuously differentiable, such that a minimization of the least squares can be established (seen in section A.3).

Least Squares grew in popularity when used in accurately describing the movement of celestial bodies [8], such that ships could better navigate open seas. It was shown that a combination of celestial observations was more reliable than many individual observations - and a simple Least Squares aided in developing this “best” observation.

Only eighteen years of age in 1795, Carl Friedrich Gauss has been credited for pioneering the fundamentals of least-squares analysis [9]. Later, the first well-known scientific demonstration of these principals was when Giuseppe Piazzi accurately tracked the position of the comet Ceres, even while it was made unobservable by the sun's glare. At the time, Gauss was then 24 years old, and his method of Least Squares outshone the nonlinear equations of planetary motion developed by Kepler. Though Kepler's model better understood the underlying dynamics of the system [10], Gauss's model more accurately determined the position of Ceres when it was observable again, 40 days later.

Similar to Gauss's success, the Least Squares method will be implemented to approximate multi-input single-output systems, without actually understanding the underlying system dynamics. Measurements and models are never perfect, and will always contain some errors (υ). The measured value of the truth x is denoted $\tilde{x}$. Estimated values ($\hat{x}$) are found using the measurements, as well as the model formed from the measurements. This model too has a residual error (r). Therefore, if x is measured, the following equations are typically used:

$$\begin{array}{ccccc} \text{measured} & & \text{true value} & & \text{measurement error} \\ \widetilde{\mathrm{x}} & = & x & + & \upsilon \end{array}$$

$$\begin{array}{ccccc} \text{measured} & & \text{estimated value} & & \text{residual error} \\ \widetilde{\mathrm{x}} & = & \widehat{\mathrm{x}} & + & \mathrm{r} \end{array}$$

Both the true value (x) and the measurement error (υ) are unknown, unlike the estimate ($\widehat{x}$) and the residual error (r), which are known explicitly. Therefore, the measurement and the residual are used to form the estimation.

In the Linear Least Squares model, approximations are formed using linear coefficients as follows:

$$\begin{array}{cccccc}
\text{measured output value} & & \text{truth model} & & \text{measurement error} \\
\tilde{y}_j & \equiv & \sum_{\mathrm{i=1}}^{n} h_i \cdot x_i & + & \upsilon_i \\
\text{estimated output value} & & \text{estimated model} & & \\
\hat{y}_j & \equiv & \sum_{\mathrm{i=1}}^{n} h_i \cdot \hat{x}_i & &
\end{array}$$

It is assumed that there is an error between the measured output and the estimated output. This is known as the residual and is defined as follows:

$$\begin{array}{ccccc}
\text{residual error} & & \text{measured output} & & \text{estimated output} \\
r_j & \equiv & \tilde{y}_j & - & \hat{y}_j
\end{array}$$

Now the following identity can be considered:

$$\begin{array}{ccccc}
\text{measured output value} & & \text{estimated model} & & \text{residual} \\
\tilde{y}_j & \equiv & \sum_{\mathrm{i=1}}^{n} h_i \cdot \hat{x}_i & + & r_j
\end{array}$$

This can be simplified as a matrix operation in the form

$$\tilde{\mathbf{y}} = H\hat{\mathbf{x}} + \mathbf{r} \tag{2.1}$$

where

$$\begin{aligned}
\tilde{\mathbf{y}} &= \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 & \cdots & \tilde{y}_{n-1} & \tilde{y}_n \end{bmatrix}^T &&= \text{measured output} \\
\mathbf{r} &= \begin{bmatrix} r_1 & r_2 & \cdots & r_{n-1} & r_n \end{bmatrix}^T &&= \text{residual errors} \\
\hat{\mathbf{x}} &= \begin{bmatrix} \hat{\mathrm{x}}_1 & \hat{\mathrm{x}}_2 & \cdots & \hat{\mathrm{x}}_{m-1} & \hat{\mathrm{x}}_m \end{bmatrix}^T &&= \text{estimated x - values} \\
H &= \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ \vdots & \vdots & & \vdots & \vdots \\ h_{n-1,1} & h_{n-1,2} & \cdots & h_{n-1,m-1} & h_{n-1,m} \\ h_{n,1} & h_{n,2} & \cdots & h_{n,m-1} & h_{n,m} \end{bmatrix} &&= \text{basis functions of model}
\end{aligned}$$

In principle, the Least Squares selects an optimum choice for the unknown parameters of $\hat{x}$ that minimize the sum of the square of the residual given by

$$J = \frac{1}{2}\mathbf{r}^T\mathbf{r}$$

This can be expanded, by submitting $\mathbf{r} = \tilde{\mathbf{y}} - \hat{\mathbf{y}} = \tilde{\mathbf{y}} - H\hat{\mathbf{x}}$ to get

$$J = \frac{1}{2}(\tilde{\mathbf{y}}^T\tilde{\mathbf{y}} - 2\tilde{\mathbf{y}}^T H\hat{\mathbf{x}} + \hat{\mathbf{x}}^T H^T H\hat{\mathbf{x}})$$

Optimality would then have a necessary condition that the first derivative is zero

$$\frac{\partial J}{\partial \hat{\mathbf{x}}} = \frac{1}{2}(-2\underbrace{\tilde{\mathbf{y}}^T H}_{H^T\tilde{\mathbf{y}}} + 2H^T H\hat{\mathbf{x}})$$

$$\frac{\partial J}{\partial \hat{\mathbf{x}}} = H^T H\hat{\mathbf{x}} - H^T\tilde{\mathbf{y}} \tag{2.2}$$

This is known as the Jacobian - and can be set equal to zero as follows:

$$H^T H\hat{\mathbf{x}} - H^T\tilde{\mathbf{y}} = 0$$

$$H^T H\hat{\mathbf{x}} = H^T\tilde{\mathbf{y}}$$

$$(H^T H)^{-1} H^T H\hat{\mathbf{x}} = (H^T H)^{-1} H^T\tilde{\mathbf{y}}$$

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T\tilde{\mathbf{y}} \tag{2.3}$$

The sufficient condition states that the second derivative is positive definite

$$\frac{\partial^2 J}{\partial \hat{\mathbf{x}}\partial \hat{\mathbf{x}}^T} = H^T H \tag{2.4}$$

This is known as the Hessian. The following examples will utilize this analytical development to develop approximations and predictions for signals of varying dimension.

2.2 1-D Examples

To explore the principal of Least Squares, simple 1-D examples are posed that will later motivate further theoretical developments. As developed in equation 2.3, the Least Squares approximation is formed by $\hat{\mathbf{x}} = (H^T H)^{-1} H^T \tilde{\mathbf{y}}$. Orthogonal polynomials [11] are generated, as seen in Appendix B, and are used as the basis functions. Also in Appendix B, the first 11 polynomials are shown in table 2.2.

Table 2.1: Orthogonal Polynomials

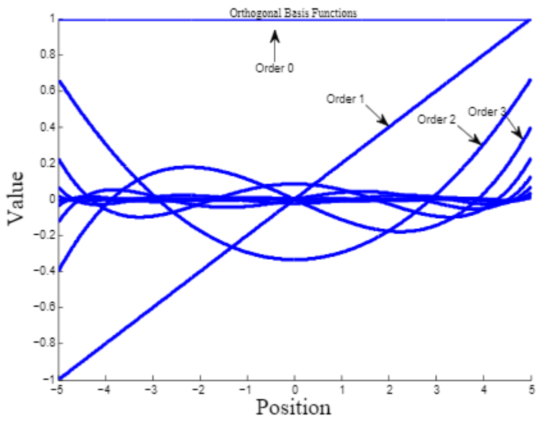
Order	Polynomial
0	1
1	x
2	$x^2 - \frac{1}{3}$
3	$x^3 - \frac{3}{5}x$
4	$x^4 - \frac{6}{7}x^2 + \frac{3}{35}$
5	$x^5 - \frac{10}{9}x^3 + \frac{5}{21}x$
6	$x^6 - \frac{15}{11}x^4 + \frac{5}{11}x^2 - \frac{5}{231}$
7	$x^7 - \frac{21}{13}x^5 + \frac{105}{143}x^3 - \frac{35}{429}x$
8	$x^8 - \frac{28}{15}x^6 + \frac{14}{13}x^4 - \frac{28}{143}x^2 + \frac{7}{1287}$
9	$x^9 - \frac{36}{17}x^7 + \frac{126}{85}x^5 - \frac{84}{221}x^3 + \frac{63}{2431}x$
10	$x^{10} - \frac{45}{19}x^8 + \frac{630}{323}x^6 - \frac{210}{323}x^4 + \frac{315}{4199}x^2 - \frac{63}{46189}$

The orthogonal polynomials are plotted in figure 2.1.

The advantage of orthogonal polynomials is that they guarantee linear independence of the columns of the H matrix when used to form inputs. As the columns of the H matrix become orthogonal, $H^T H$ becomes diagonal. Generally speaking, an input is expanded using the polynomials, across the columns of the H matrix, and the coefficients weight this expansion to form an approximation. The following examples are a demonstration of that principle.

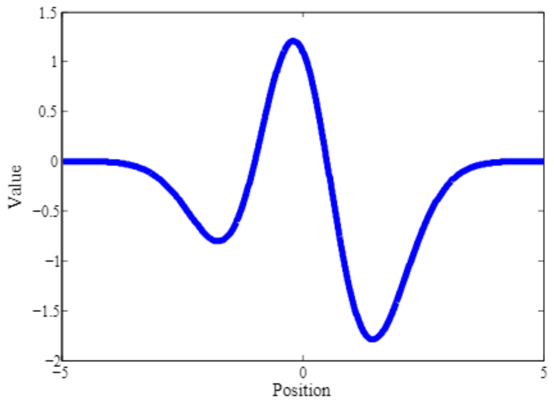
2.2.1 Example 1

This example hopes to illustrate the capabilities of the Least Squares method when approximating a signal using the orthogonal polynomials described above. The signal for this example is illustrated in figure 2.2.



(a) Orthogonal Basis Functions

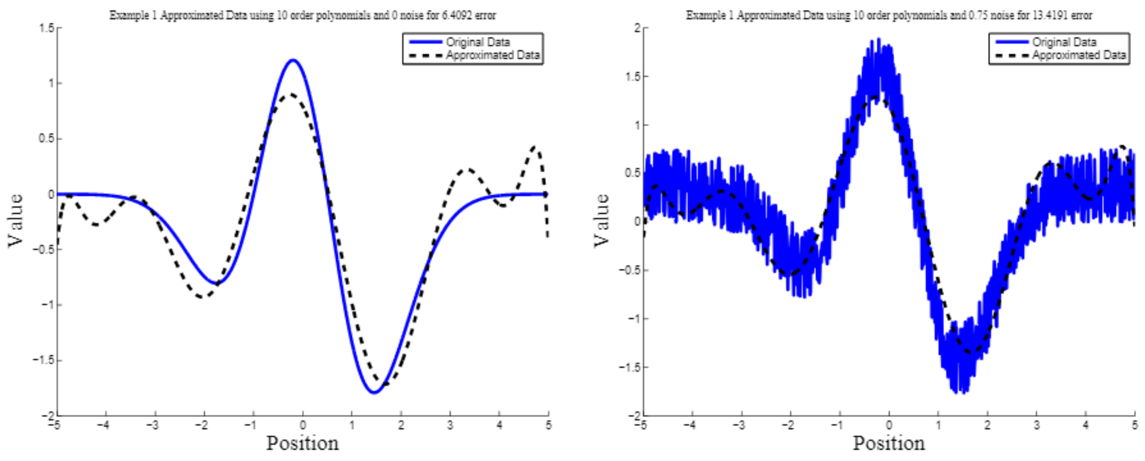
Figure 2.1: The orthogonal polynomials form a basis for the Least Squares approximation examples in this chapter.



(a) Example 1: Original Data

Figure 2.2: In this example, the original function of $1.1 \cdot (1 - x - 2 \cdot x^2) \cdot e^{-\frac{x^2}{2}}$ is used to generate the data.

In this example, both noise amplitude and number of polynomials were varied. In figure 2.3, a typical approximation is illustrated.



(a) Least Squares Approximation using 10th order polynomials and to approximate the signal with 0 noise

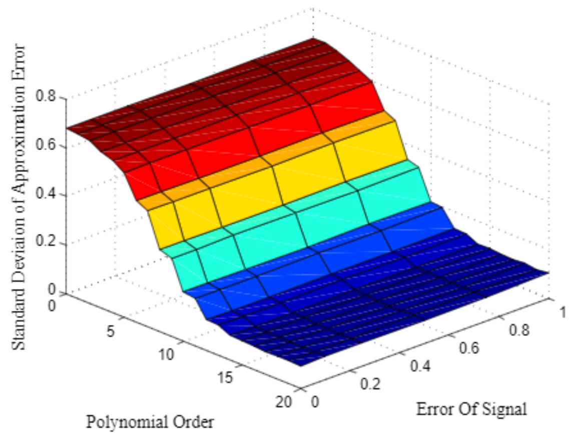
(b) Least Squares Approximation using 10th order polynomials and to approximate the signal with .75 noise

Figure 2.3: These two plots show the Least Squares approximation using 10th order polynomials, both without noise and with noise.

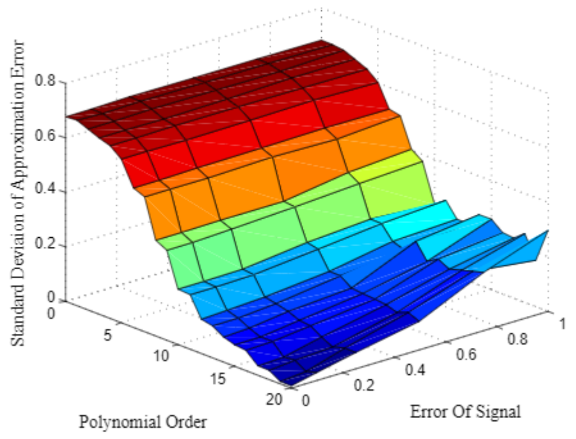
As seen in figure 2.3, the approximation is not perfect. In fact, there is almost always some degree of error. Figure 2.4 evaluates the 2-norm error of the approximation with respect to the true signal, varying noise input and polynomial order.

Though it is seen that decreasing noise and increasing polynomial order allows for a better approximation, it is seen that using more than 11 polynomials no longer has a significant improvement on error minimization. To further explore error minimization techniques, the least squares error will be compared with 2-norm error minimization using convex optimization language, CVX [12]. This method is compared to the analytical least squares, and section A.5 shows it produces the same results.

In Example 1, the Least Squares method was evaluated while approximating a 1-D region. Lim-



(a) Standard Deviation of the Approximation Error (Approximated vs Noise Free)



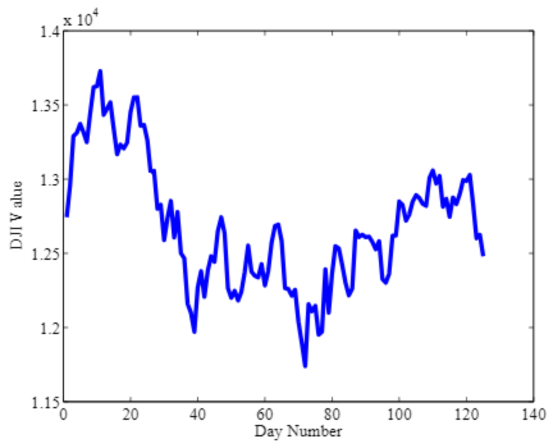
(b) Standard Deviation of the Approximation Error (few samples)

Figure 2.4: The error represented in this plot is the standard deviation of the error of the approximation to the original signal without noise. It is seen that decreasing signal noise and increasing polynomial order forms a better approximation with few samples (≈ 25), but with more samples (≈ 1000) increasing polynomial order alone forms a better approximation.

itations were shown by decreasing the number of data points, and increasing the measurement noise. For this example (as with most examples), it was seen that the approximation exhibited little improvement beyond a certain number of basis functions.

2.2.2 Example 2a: 1-D Stock Market Approximation

Example 2 focuses on using a partial data set for training, in an attempt to reproduce the full data set. An easy way (and profitable way) to demonstrate this principle would be through a stock market model. As seen in 2.5, the Dow Jones Industrial Average closing price is the signal to be approximated. The same orthogonal polynomials from Example 1 are utilized, but now only part of the data set will be used to generate a model. In this example, 25 days are predicted, while varying the number of days in the training set and the number of orthogonal polynomials.



(a) Dow Jones Industrial Average Closing Price

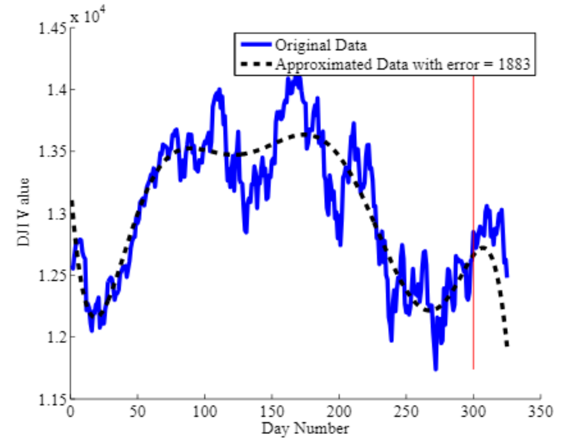
Figure 2.5: The original data, which consists of the most recent Dow Jones Industrial Average closing price

All but the last 25 points (note the vertical line toward the right) were used in creating a model. The model was then used to evaluate the entire data set as seen in figure 2.6.

As seen in figure 2.6, there is an error in the approximation. This is due to high frequency noise, and limitations of the polynomial approximation. The error appears greater in the prediction region (to the right) of the model. This is verified by figure 2.7, which shows the Standard Deviation of the Error with respect to number of polynomials and size of training data.

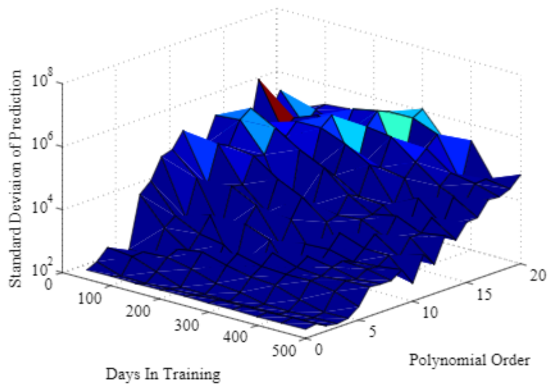
As seen in figure 2.7, the standard deviation of the error of the prediction (the reconstruction of the data not used in training) is greater than that of the approximation. In fact, the prediction error accounts for much of the error of the global approximation. In Example 2b, the goal is to lower the error of the prediction by weighting the value of the training data, or the error between the model and the original data, such that the model has less error near the region of the prediction.

Though this Example has, in some cases, reasonably approximated and predicted the closing

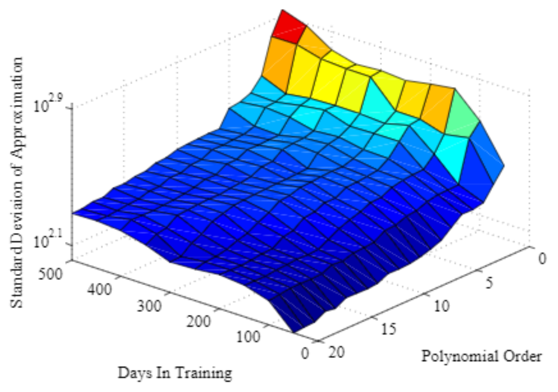


(a) Approximation using 7 polynomials and 300 days to train the model

Figure 2.6: A sample approximation showing how training data can be used to form a prediction signal.



(a) Standard Deviation of Prediction Error



(b) Standard Deviation of Approximation Error

Figure 2.7: For the global approximation, the Standard Deviation of the Error decreases as polynomial order decreases and number of training days increases. This is widely influenced by the model's ability to form a prediction.

price of the Dow Jones Industrial Average, further improvements can be made on the prediction.

2.2.3 Example 2b: 1-D Stock Market Weighted Approximation

In section 2.1 it was established that an analytical solution to the Least Squares exists in the form $\hat{\mathbf{x}} = (H^T H)^{-1} H^T \tilde{\mathbf{y}}$ which minimizes $J = \frac{1}{2}\mathbf{r}^T\mathbf{r}$. The problem with this formulation is that it gives equal weight to the approximation error. If a prediction of future data points is to be made, more emphasis needs to be placed on minimizing the error closest to the predicted points.

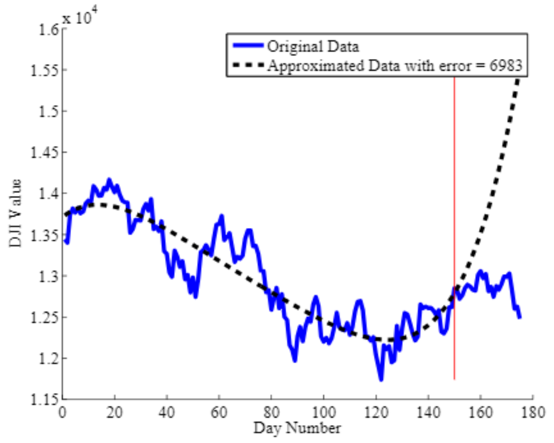
As explained by Crassidis and Junkins [6], emphasis can be giving using a weight matrix, which is a diagonal matrix in the form

$$W = \begin{bmatrix} w_{closest} & 0 & 0 & 0 & 0 \\ 0 & w_{closer} & 0 & 0 & 0 \\ 0 & 0 & w_{close} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

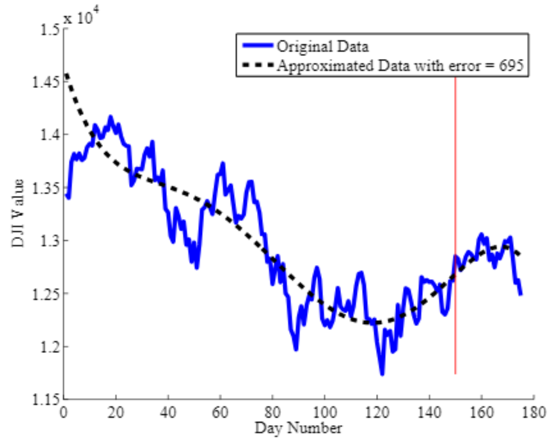
In order to prevent the analytical solution from developing a slightly singular $H^T W H$, the diagonal elements of W must not be close to zero. Also, H must have a greater number of rows than columns. This weighting matrix forms a minimization of $J = \frac{1}{2}\mathbf{r}^T W \mathbf{r}$, which like Appendix A and equation 2.3, will give the analytical solution of

$$\hat{\mathbf{x}} = (H^T W H)^{-1} H^T W \tilde{\mathbf{y}} \tag{2.5}$$

The problem is setup in the same manner as Example 2a, and as seen in figure 2.8, an example of improvement in the prediction is seen with low-order polynomials.



(a) Approximation using 5 polynomials and 150 days to train the model with NO weight



(b) Approximation using 5 polynomials and 150 days to train the model with a weighted analytical solution

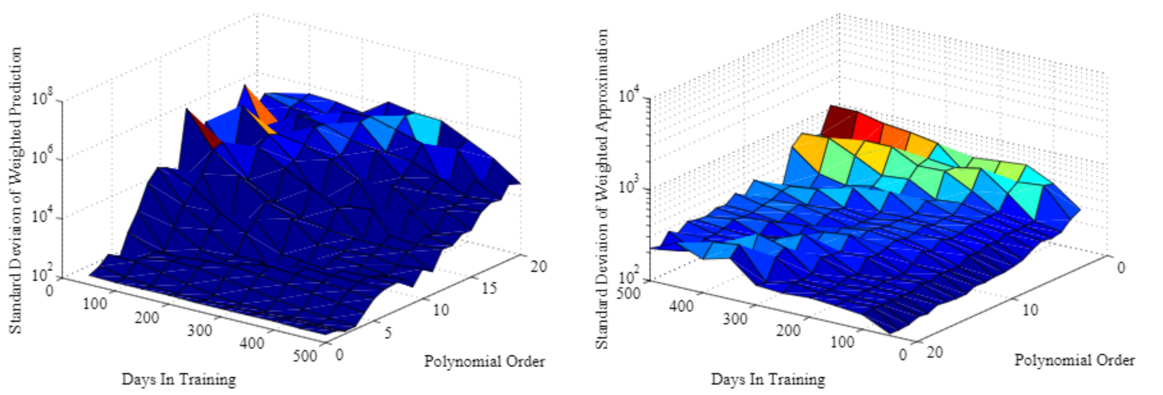
Figure 2.8: When forming a prediction, there are advantages to weighting the error minimization closer to the predicted points.

When compared in figure 2.9, it is seen that the standard deviation of the weighted prediction error is only slightly less than the unweighed, while the standard deviation of the approximation error is slightly greater.

Though weighting improves the prediction error slightly, the fact that the prediction error is still very high indicates an adequate model has not been developed. To better understand the system, a different set of basis functions should be used or more inputs may be utilized in model development. The following examples utilize multiple inputs to develop a model.

2.3 2-D Examples

For 2-D examples, the basis functions (which are the same as in the 1-D examples) are combined, as each dimension (input) can act independently or collaboratively. The H matrix must reflect



(a) Standard Deviation of Prediction Error (b) Standard Deviation of Approximation Error

Figure 2.9: The difference between (average prediction error - average approximation error) is smaller when the least squares is weighted near the prediction.

that as well. Table 2.2 shows a typical combination of two inputs, with respect to desired order. The each row of the H matrix will contain all orders up to the desired order.

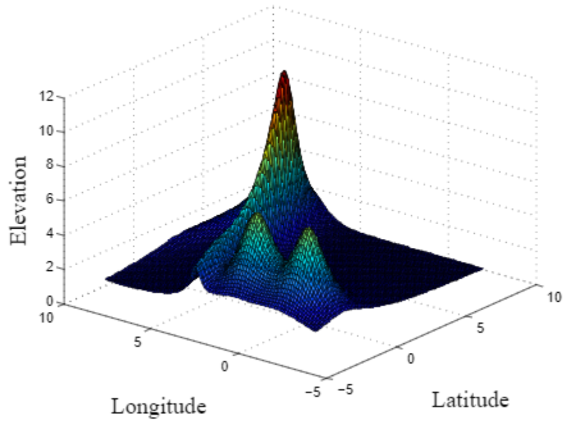
Table 2.2: Relationships of Increasing Order for 2 Dimensions

Order	Relationship Formed			
0		1		
1	x		y	
2	x^2	$x \cdot y$		y^2
3	x^3	$x^2 \cdot y$	$x \cdot y^2$	y^3
higher		$\vdots$		

An algorithm to develop multi-dimensional relationships as such is explained in appendix C. For the next two examples, orthogonal polynomials will still be utilized, as seen from the 1-D examples. They will be combined using the multi-dimensional relationships discussed.

2.3.1 Example 3: "Monster Function"

The first 2-D example is a simple benchmark function known as the "Monster Function". The original function is displayed in figure 2.10.



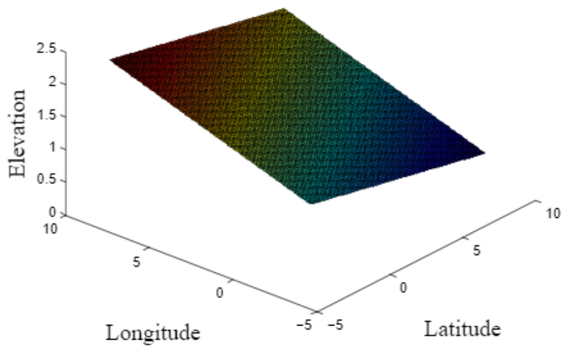
(a) Example 3: Original 2-D function

Figure 2.10: The original function is $f = \frac{10}{(y-x^2)^2+(1-x)^2+1} + \frac{5}{(y-8)^2+(x-5)^2+1} + \frac{5}{(y-8)^2+(x-8)^2+1}$

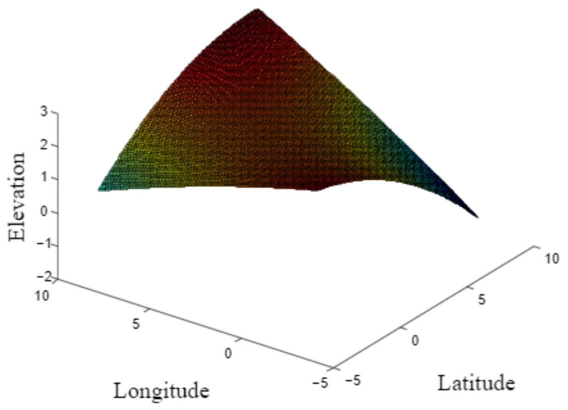
This example was approximated, varying the number of polynomials along with number of samples. An example approximation, while keeping number of samples consistent) is shown in figure 2.11.

Figure 2.11 demonstrates that a better reconstruction can be formed with an increase in polynomial order. This is confirmed in figure 2.12, and also demonstrates the advantage of having a greater number of training points. The standard deviation of the error is plotted in figure 2.12.

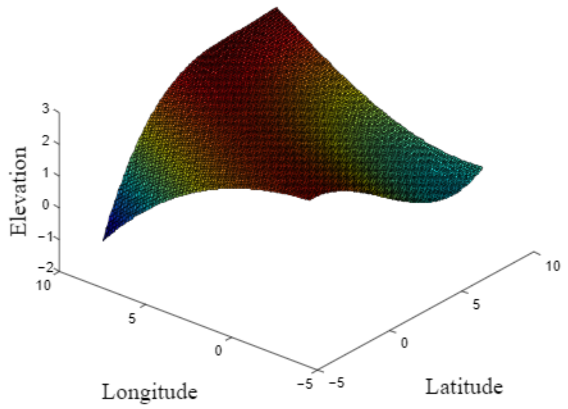
It is seen that in general, increasing polynomial order and increased number of samples decrease the standard deviation of the error. The number of samples has a greater influence when the number of basis functions is high - almost as high as the number of samples. When the number of measurements are less than the number of basis functions, the problem is said



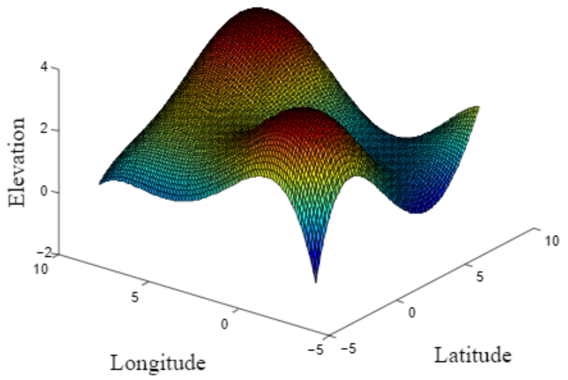
(a) Approximation formed with 1 polynomials



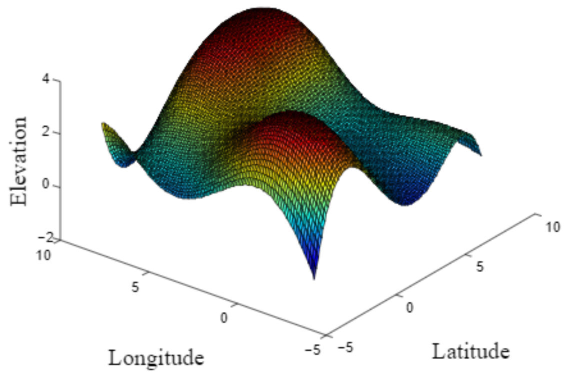
(b) Approximation formed with 2 polynomials



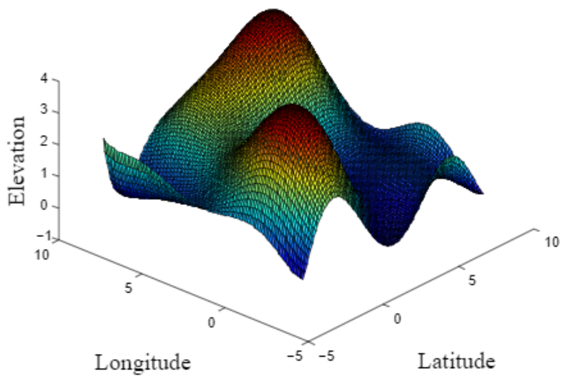
(c) Approximation formed with 3 polynomials



(d) Approximation formed with 4 polynomials

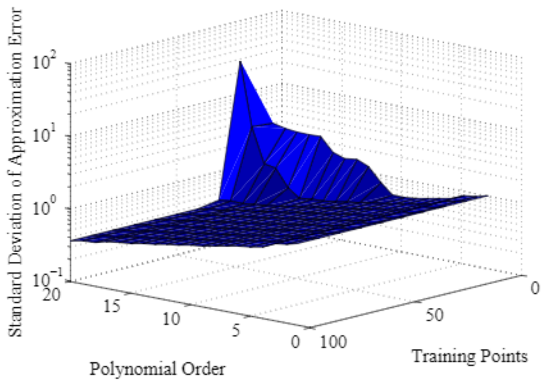


(e) Approximation formed with 5 polynomials

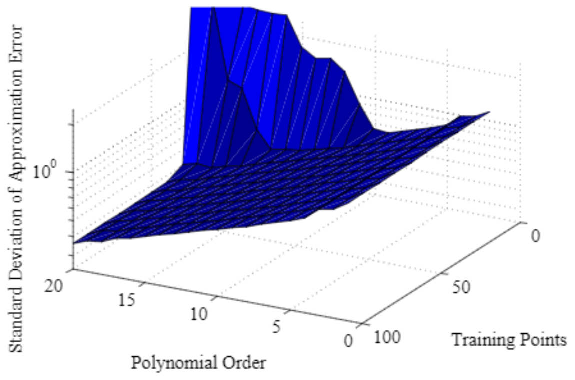


(f) Approximation formed with 6 polynomials

Figure 2.11: When increasing the order of polynomials, the approximation better reconstructs the original data as seen in figure 2.10.



(a) Standard Deviation of Approximation Error



(b) Standard Deviation of Approximation Error (zoom)

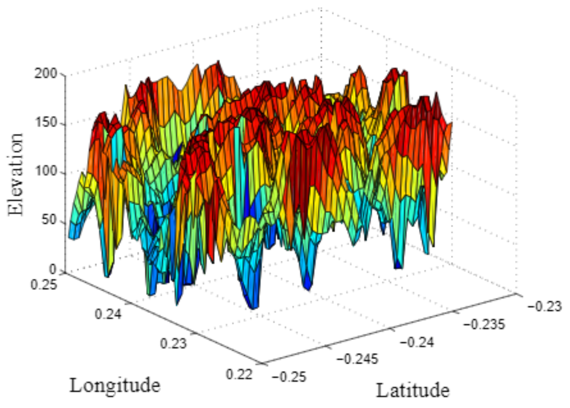
Figure 2.12: The standard deviation of the approximation error decreases with increased polynomial order, and decreases slightly with increased training points.

to be "under-defined" - and it is difficult for the least squares to form a solution. It is seen that at high polynomial order and low number of samples, that the error increases for this reason.

The "Monster Function" was a very basic benchmark problem, with only three main features to approximate. As the number of polynomials increased, the standard deviation of the error decreased with no notable tradeoff between number of polynomials and error (it is hypothesized that there would be a tradeoff with a higher polynomial order - or a different set of basis functions). Still, for this simple model, the least squares approach does a reasonable job developing an approximation model, but this next example will consider a more complicated actual model.

2.3.2 Example 4: Moon Data

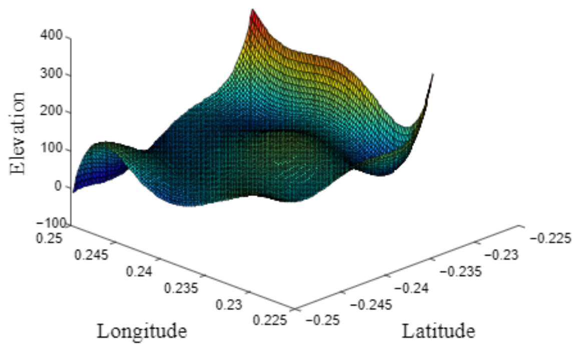
The second 2-D example is a much more complicated signal. This signal represents the terrain of the moon in a certain region, as shown in figure 2.13.



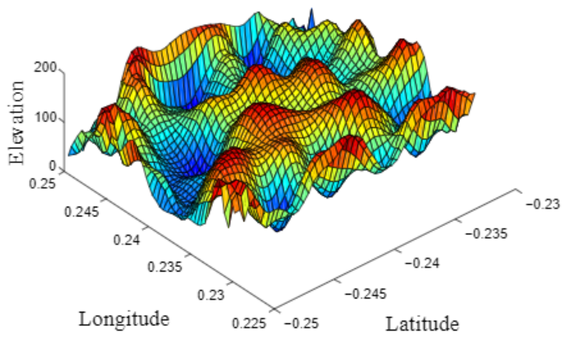
(a) Example 4: Original 2-D function

Figure 2.13: The Original Moon Terrain Data

To highlight the capabilities of least square model development, polynomial order and number of training samples were varied. Samples approximations are given in figure 2.14. Even with a computationally-expensive 20th order polynomial basis function set, the approximation still lacks many of the signal's features.



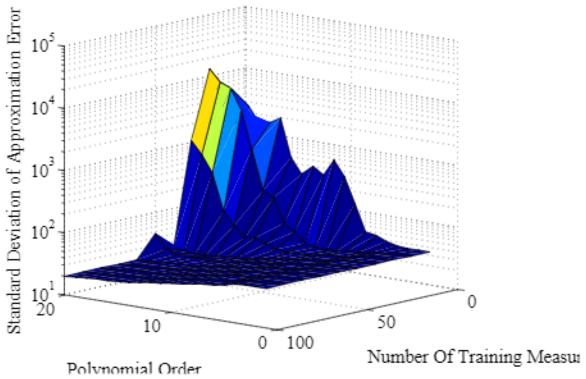
(a) Approximation using 5 polynomials and 10x10 measurements



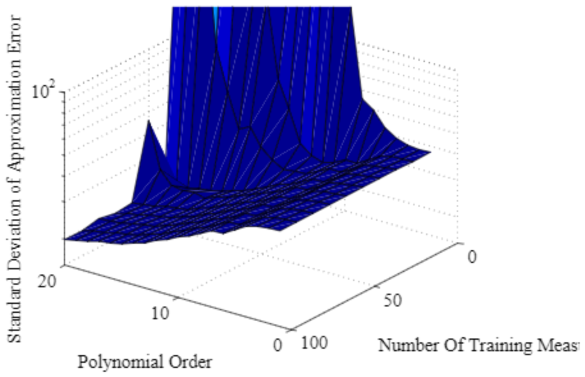
(b) Approximation using 20 polynomials and 50x50 measurements

Figure 2.14: As seen, increasing the number of polynomials vastly improves the approximation model - as does increasing the number of training points

As seen from figure 2.14, improvements in the model are made by increasing the polynomial order, and less significantly, the number of training points. The standard deviation of the error is plotted in figure 2.15.



(a) Standard Deviation of Approximation Error



(b) Standard Deviation of Approximation Error (zoomed in)

Figure 2.15: The standard deviation of the approximation error decreases with increased polynomial order, and decreases slightly with increased training points.

It is seen that in general, increasing polynomial order and increased number of samples decrease the standard deviation of the error. It is seen that at high polynomial order and low number of samples, that the error increases. This is, again, because the number of measurements approaches the size of the model, and it becomes more difficult for the least squares to form a solution. This is one of the key advantages of sparsity, and will be discussed further in Chapter 3.

2.4 n-Dimensional Example

There are certain models, that when varying in dimensionality, or number of inputs, better approximate the system in question. For this example, the stock market model is revisited. This time, the basis functions are not used to expand the input, as the computational burden is too great and will lead to more columns than rows of the H matrix, making the problem un-accomplishable through least squares. Instead, previous state data from the signal will be used as the input.

2.4.1 Example 5a: multi-stream n-D Stock Market Approximation

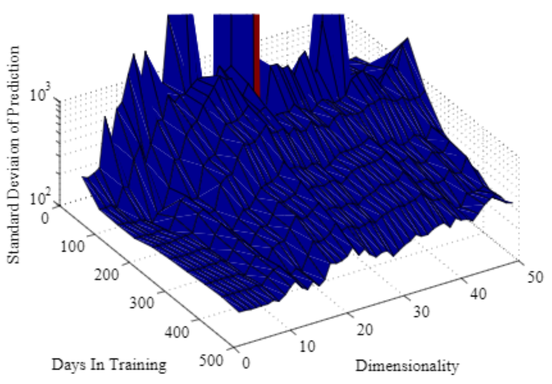
In this example, the Dow Jones Industrial average is treated as a finite difference model, indicating that the previous states have an influence on the current state. The original data is seen in 2.5 - again, the closing price.

In this first example, the model is posed as follows:

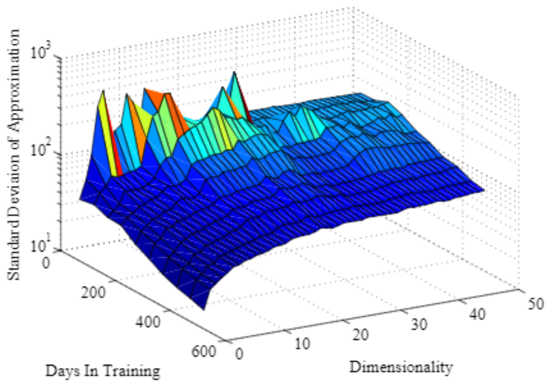
$$\begin{aligned} close_{today} = \quad & x_1 \cdot open_{yesterday} + x_2 \cdot high_{yesterday} + x_3 \cdot low_{yesterday} + x_4 \cdot close_{yesterday} \\ & + x_5 \cdot open_{day-before-yesterday} + x_6 \cdot high_{day-before-yesterday} + x_7 \cdot low_{day-before-yesterday} \\ & + x_8 \cdot close_{day-before-yesterday} \cdots \end{aligned}$$

where close is the closing price, open is the opening price, and so on. Each day (n)'s prices represents a dimension, an input, while each price in the day represents a stream. Therefore, this model has 4 streams and n*4-dimensions. Examples of this type of model generation are given in figure 2.18. The standard deviation of the error is plotted in figure 2.16.

In this example, it is shown that low dimensionality is of an advantage in building a model - indicating perhaps that future data is based mostly on more current data. This is further emphasized as in instances of low dimensionality, the standard deviation of the prediction error is lowest at 100 days of training. Though this example includes all possible data to build the model, it will be investigated how using only one stream - the closing price only - will build a better prediction.



(a) Standard Deviation of Prediction Error



(b) Standard Deviation of Approximation Error

Figure 2.16: From this analysis, fewer dimensions and more predictions form a better prediction approximation model.

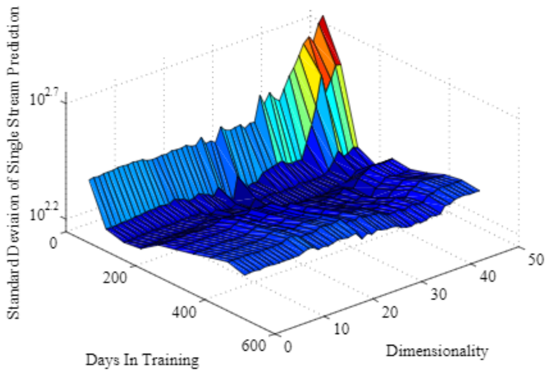
2.4.2 Example 5b: single-stream n-D Stock Market Approximation

This example only considers the closing prices. Previous states are the input and the current state is the output, thus making a finite difference equation. In this example, the model is posed as follows:

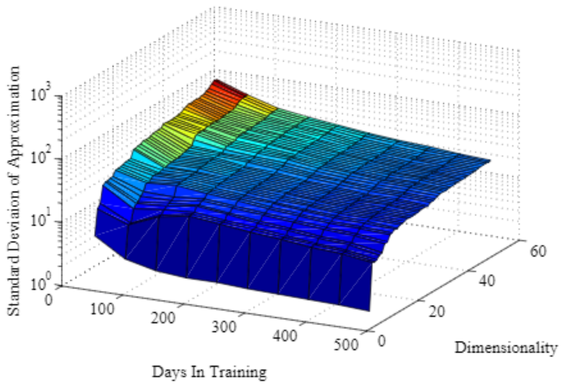
$$close_{today} = x_1 \cdot close_{yesterday} + x_2 \cdot close_{day-before-yesterday} \cdots$$

Each day (n) used to form the model represents a dimension. The standard deviation of the error is plotted in figure 2.17.

It is seen that the standard deviation of the error is lower when only a single-stream model is used. The following plots in figure 2.18 compare the two approaches through examples. It is noted that the standard deviation of the prediction would most likely be more comparable to that of the approximation if the prediction into the future was fewer states.



(a) Standard Deviation of Prediction Error

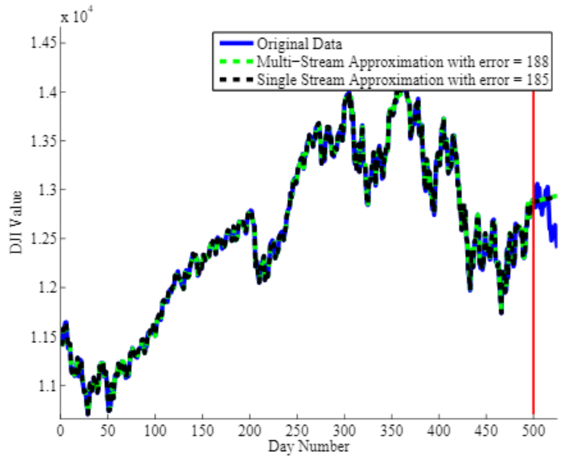


(b) Standard Deviation of Approximation Error

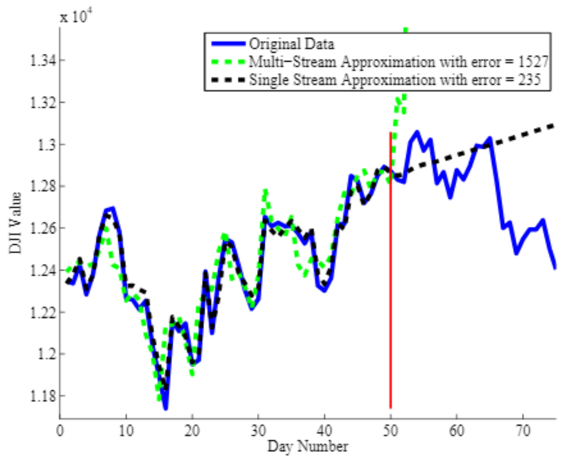
Figure 2.17: From this analysis, 40 days and 200 measurements best form the prediction model, while a high number of measurements and low number of days forms the best approximation model.

2.5 Conclusion

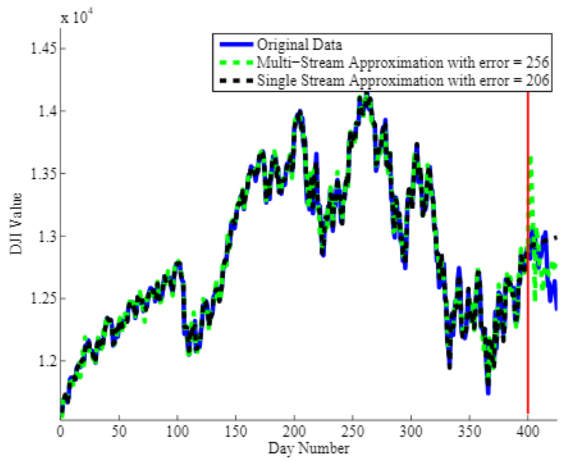
The least squares method for model development has many advantages, as demonstrated from the examples above. In some examples, a problem of analytical rank deficiency is faced when the number of parameters in the model approach the number of measurements. Also, all these examples (excluding example 5) used orthogonal polynomials, when other basis functions could be used. Adding more basis functions would increase the number of parameters in the model, thus the least squares method may no longer be an adequate method for model development for the rank-deficient reason mentioned earlier. In chapter 3, a larger set of basis functions are used, but some are eliminated to form a sparse solution.



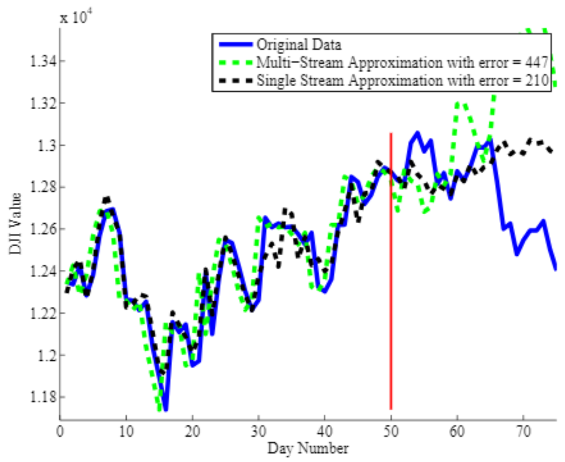
(a) n=1 with 500 measurements



(b) n=6 with 50 measurements



(c) n=15 with 400 measurements



(d) n=24 with 50 measurements

Figure 2.18: It is shown that a better prediction is formed when the model uses days closer to the time of prediction.

Chapter 3

Sparse Approximation By Convex Optimization

3.1 Introduction

In chapter 2, the concept of least squares was introduced. Limitations of using a least squares solution were seen for an over-defined problem, in which the number of samples was less than the number of basis functions. For an algorithm to pick the best basis functions to reconstruct a signal, very large dictionary should be used - thus increasing the opportunity for the signal's dominant function(s) to be present. Using such a dictionary makes a problem over-defined, so a different approach must be used to determine the function weighting that produces the minimal residual. Posing a least squares problem as an optimization routine instead of using an analytical solution will have some computational advantages, but will still exhibit many of the same difficulties. To select the best basis from a large dictionary, sparsity is enforced.

The intent of sparse approximation [13] is to approximate an n-Dimensional signal over a redundant set of functions, which are known to be the 'dictionary'. The cardinality, or number of elements in the set, would greater than the number of measurements. This makes the dictionary over-complete, and the goal is to find the minimum number of non-zero coefficients which weight the functions in the dictionary, to form the sparsest [2] solution. To accomplish this, the problem is posed as an optimization routine, and linear programming [14] methods are utilized to find only one solution. Linear programming is the process of minimizing an linear equality subject to a finite number of constraints and has application in solving a diverse range of combinational problems. Using the Matlab modeling language CVX [12], objectives and constraints are specified using standard Matlab syntax. In this way, penalties can be placed on the objective function if the number of non-zero coefficients are too great, or the error, relative to the the least squares solution, is too large.

This chapter details the results of various optimization approaches to achieve a sparse solution.

3.1.1 Basic Definitions

The following basic descriptions will be used to vary the optimization routine to evaluate a sparse solution:

If a vector exists in the form $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

Cardinality The cardinality of a set is a measure of the "number of elements of the set".

Norm-0 (written $\|\cdot\|_0$) The 0-norm is given as $\|x\|_0 = \sum_{r=1}^{n} \begin{cases} 0 & \text{if } x_r^2 = 0 \\ 1 & \text{if } x_r^2 > 0 \end{cases}$

Norm-1 (written $\|\cdot\|_1$) The 1-norm is given as $\|x\|_1 = \sum_{r=1}^{n} \sqrt{x_r^2}$

Norm-2 (written $\|\cdot\|_2$) The 2-norm is given as $\|x\|_2 = \sqrt{\sum_{r=1}^{n} x_r^2}$

Norm-infinity (written $\|\cdot\|$) The infinity-norm is given as $\max(x_r)$

3.2 Convention

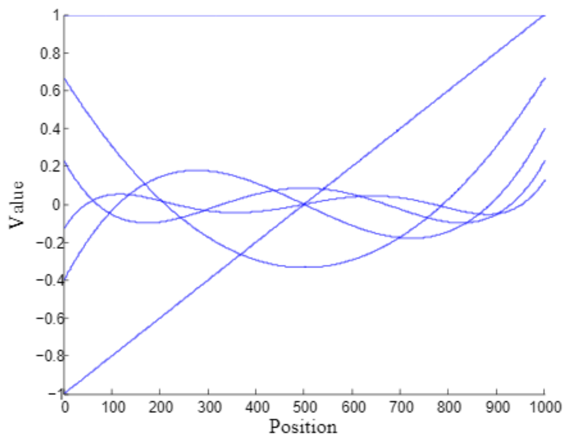
Generally, the set of basis functions is defined as A (this is the same as H from Chapter 2), and the coefficients x (same as $\hat{x}$). Thus, the measured output b (same as $\hat{y}$ from Chapter 2) can form the relationship of

$$A \cdot x \approx b \tag{3.1}$$

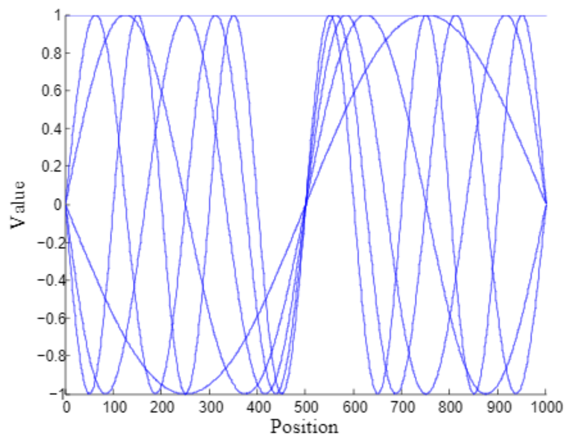
This would be the ideal solution, but with most any reconstruction of a signal, there is an error, which can be expressed as $A \cdot x - b$. Because there are many solutions to set this error equal to zero, the least square solution , $\|A \cdot x - b\|^2$ or $[A \cdot x - b]^T [A \cdot x - b]$, is often seen as "the best solution". The goal then, is still to minimize $\|x\|_1$, while trying to achieve an error close to that of the solution (x) that minimizes $\|A \cdot x - b\|_2$. Since b and A are given, the variables involved in this minimization are the variables of the vector x.

3.3 Experimental Setup

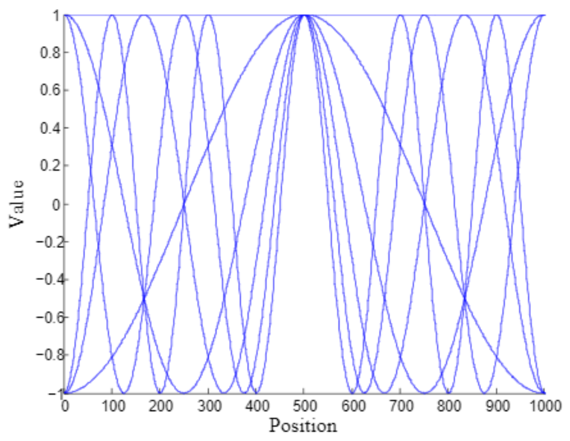
An over-complete dictionary is formed using Polynomials, Cosine-based Functions, and Exponent-based Functions. These types of functions are described in figure 3.1. Not only are the example functions used, but also each function shifted for each measurement point. This means that there are MANY more columns than there are rows, such that a least squares approach would be impossible. Also, with multi-dimensional inputs, the basis functions are combined as described in appendix C.



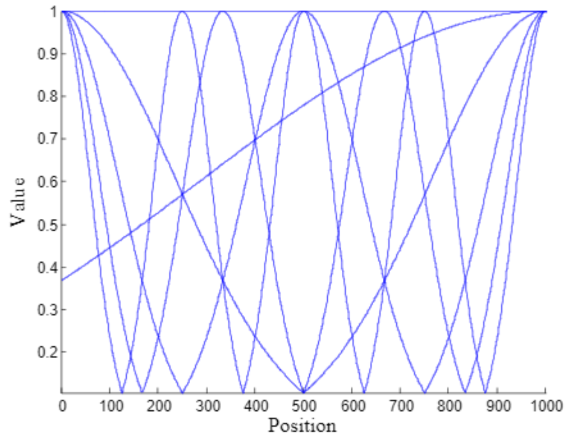
(a) Polynomials



(b) Sine-based Functions



(c) Cosine-based Functions



(d) Exponent-based Functions

Figure 3.1: In this example, an over-complete set of basis functions is used, involving Polynomial functions, Sine-based functions, Cosine-based functions, and Exponent-based functions. When combining these sets, and over-complete dictionary is formed.

3.4 Problem 1

Given the definitions above in section 3.2, it is seen that the absolute error of the least squares can be be defined as norm$(A \cdot x - b, 2)$. This is the error that when minimized, produces the "best" solution. Since the dictionary is highly over-complete, a simple pseudo-inverse will not yield meaningful coefficients due to the computational limitations of Matlab. Therefore, CVX was used to solve for the 2-norm coefficients, and these coefficients were assumed to yield the optimal solution.

In this example, the 1-norm of the coefficients is minimized, subject to the constraint that relates the allowed error of this minimization to the optimal solution. The problem is posed as follows:

```
cvx_begin
    variable coeffs2(y)
    minimize(norm(A*coeffs2-b,2));
cvx_end


cvx_begin
    variable coeffs(y)
    minimize( norm(coeffs,1));
    subject to
        norm(A*coeffs-b,2) <= norm(A*coeffs2-b,2)*gamma;
cvx_end
```

3.4.1 Example 1.1: 1-D benchmark function

To test the development of sparsity, and to see its effects, a 1-Dimensional example is used. The original data is found in figure 3.2. This is the same function as seen in section 2.2.1 of $1.1 \cdot (1 - x - 2 \cdot x^2) \cdot e^{-\frac{x^2}{2}}$. Varying γ, it is seen the effects on the approximation.

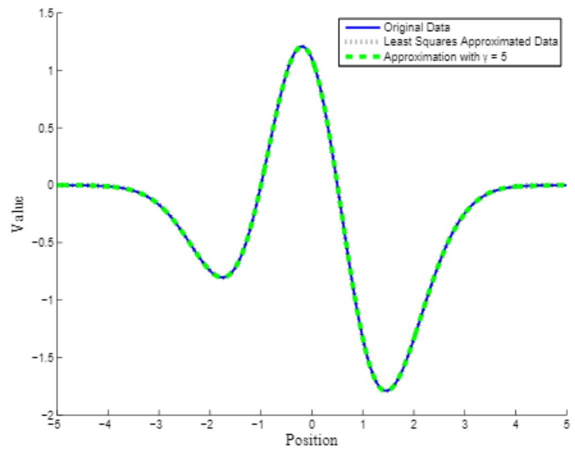
In this example, the benchmark function is very simple, and therefore easy to approximate. When relating the norm-1 of the coefficients to the norm-2 of the error, a larger value of γ is required to produce any noticeable difference. The advantage of this optimization routine, however, is that it dramatically reduces the number of model coefficients through only a modest increase in γ. This is further illustrated in figure 3.3.

When increasing γ, the number of coefficients drops to, and remains level at 15, while the standard deviation of the error increases. If thought of in terms of cardinality $\cdot$ standard deviation as a heuristic parameter to minimize to form the best sparse approximation, one would imagine an algorithm would adaptively select this development. It is therefore seen that $\gamma = 4$ produces an adequately sparse solution, with 15 coefficients weighted in such a way that the error of the approximation is relatively small.

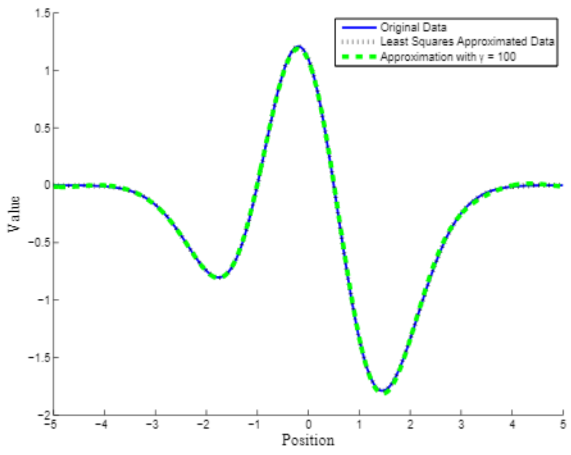
3.4.2 Example 2.1: 1-D Stock Market Model

The second model, is again the stock market model, which approximates the Dow Jones Industrial average. In this example, a set of data is used to develop a model, which forms a prediction signal. As seen in figure 3.14, there are immediate advantages to a sparse approximation.

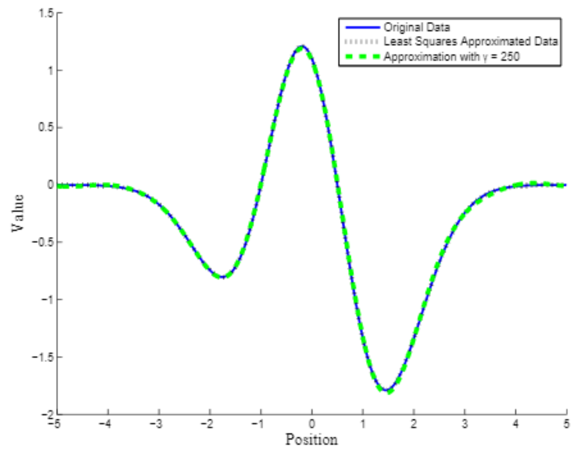
As illustrated in figure 3.15, the least squares poorly developed a prediction model, while the sparse approximation reasonably developed a prediction model. In fact, even with 7 coefficients, a reasonable approximation and prediction was formed.



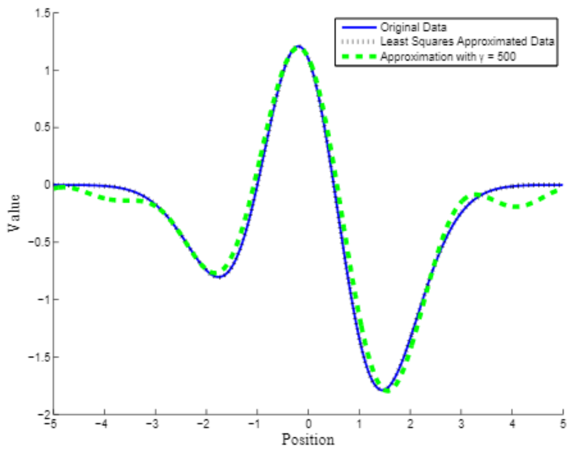
(a) $\gamma = 5$



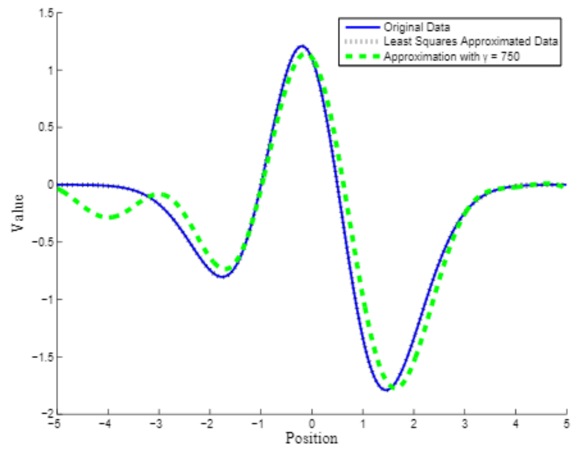
(b) $\gamma = 100$



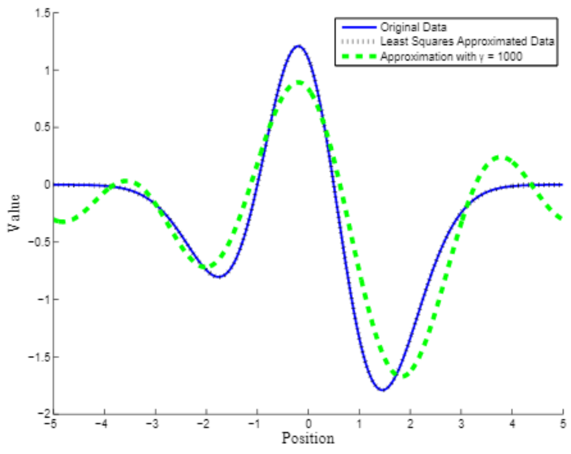
(c) $\gamma = 250$



(d) $\gamma = 500$

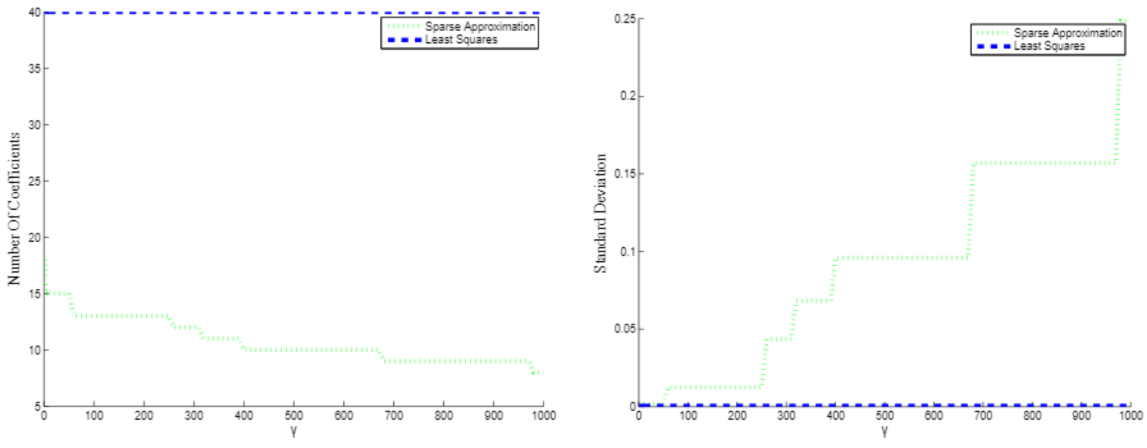


(e) $\gamma = 750$



(f) $\gamma = 1000$

Figure 3.2: By increasing γ, fewer basis functions are used to approximate the signal.



(a) Number of Coefficients vs γ (b) Standard Deviation of Error vs γ

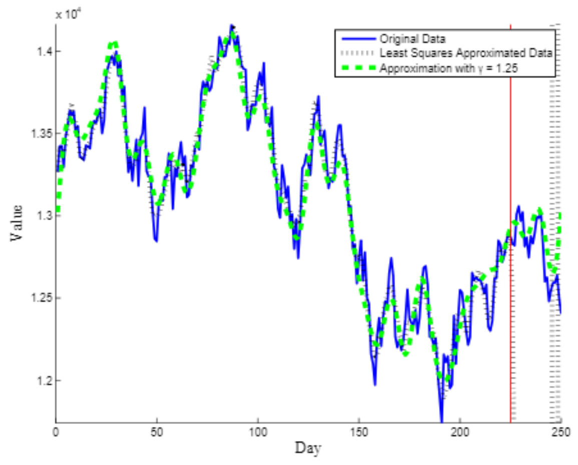
Figure 3.3: It is seen that an adequate model is made with 15 coefficients, found with $\gamma = 4$.

Sparse approximations have shown to reasonably develop approximation models for one dimensional signals. For less complicated models (as seen in example 1), higher values of γ are used to decrease the number of coefficients, but with a more complicated model (such as example 2), lower values of γ can adequately decrease the number of coefficients used to build an approximation model.

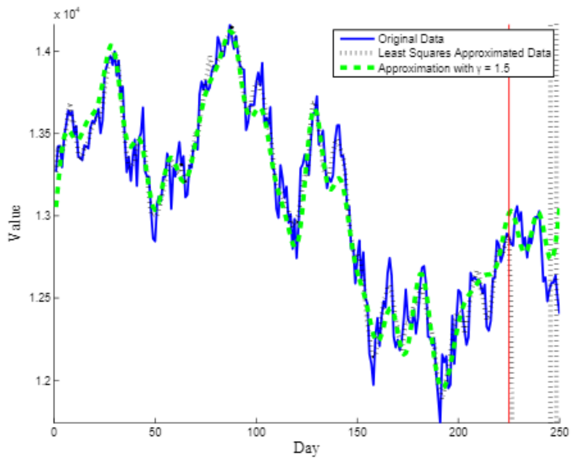
3.4.3 Example 3.1: 2-D "Monster Function"

Now multi-dimensional signals will be investigated. The first example is the benchmark function, with few complications. This example uses a benchmark function, $\frac{10}{(y-x^2)^2+(1-x)^2+1} + \frac{5}{(y-8)^2+(x-5)^2+1} + \frac{5}{(y-8)^2+(x-8)^2+1}$. The results from low-$\gamma$ approximations are shown in figure 3.6, in which lower values of γ produce a reasonable approximation.

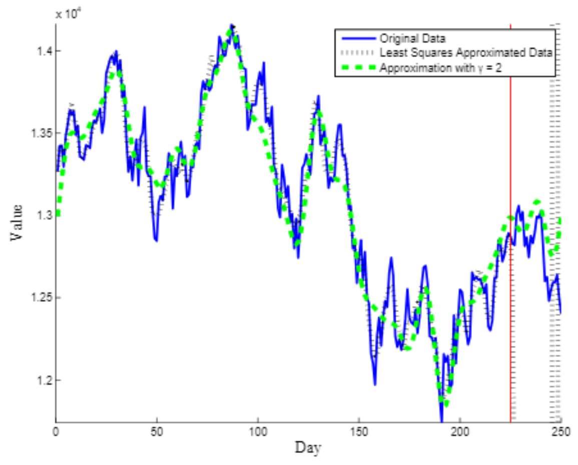
As seen in figures 3.6 and 3.7, lower values of γ form a very good approximation, while higher values of gamma, in the range of 4 and higher, utilize too few basis functions to form a reasonable approximation.



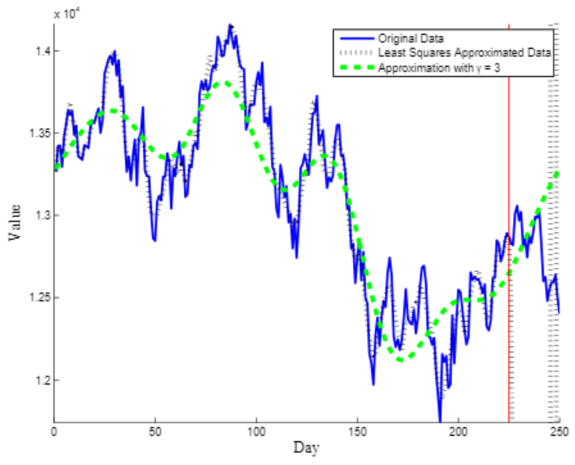
(a) $\gamma = 1.25$



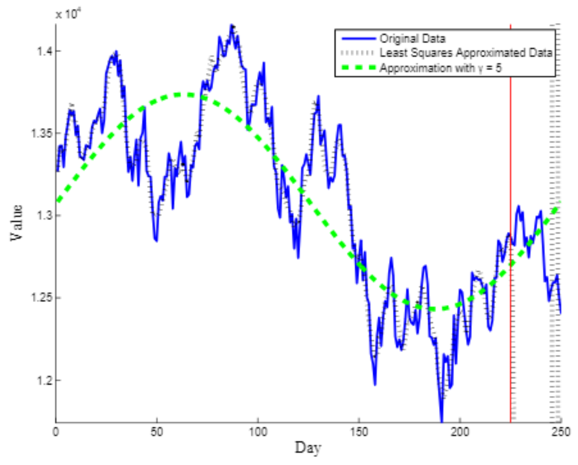
(b) $\gamma = 1.5$



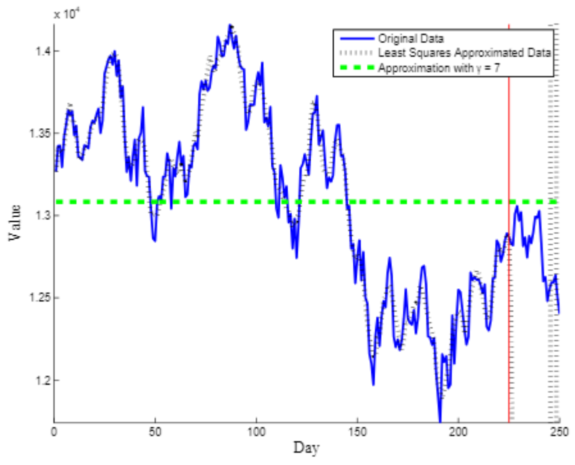
(c) $\gamma = 2$



(d) $\gamma = 3$

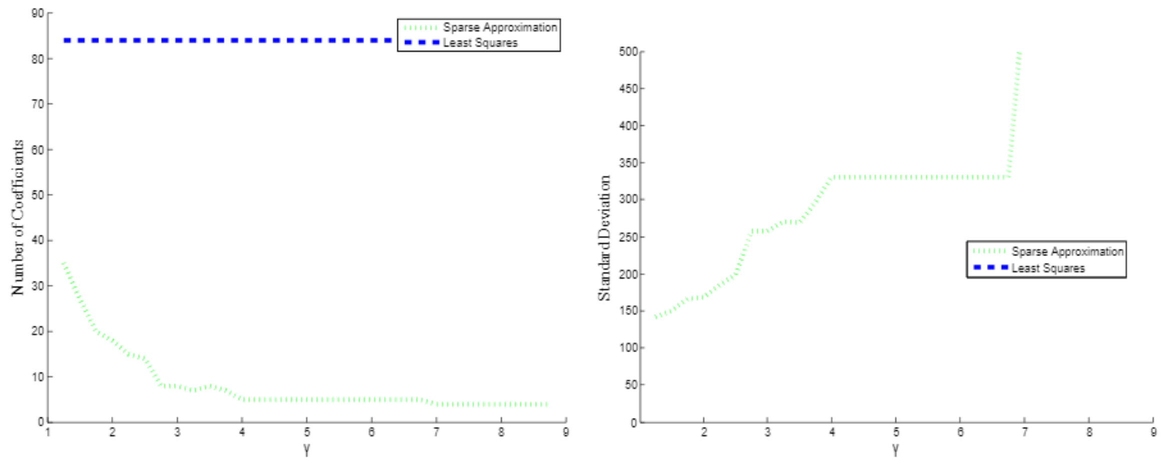


(e) $\gamma = 5$



(f) $\gamma = 6$

Figure 3.4: By increasing γ, fewer basis functions are used to approximate the signal.



(a) Number of Coefficients vs γ (b) Standard Deviation of Error vs γ

Figure 3.5: It is seen that an adequate model is made with 7 coefficients, found with $\gamma = 2.75$.

As seen in figure 3.17, the number of coefficients does not increase dramatically through increasing γ beyond 3.75, while the standard deviation of the error continues to increase.

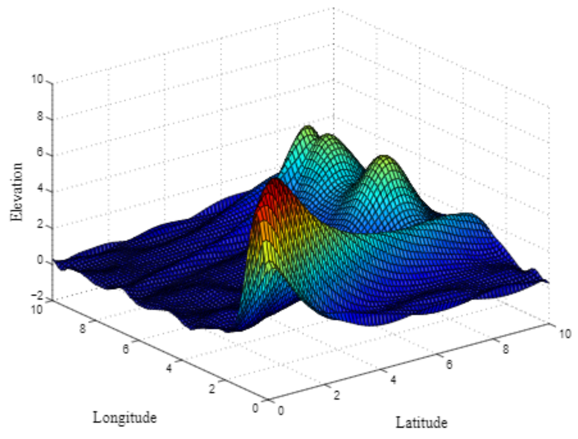
3.4.4 Example 4.1: 2-D Moon Terrain

A more complicated signal is the moon terrain data. This is the same data as seen in figure 2.13. In figure 3.9, it is seen that for low values of γ, a reasonable approximation is formed.

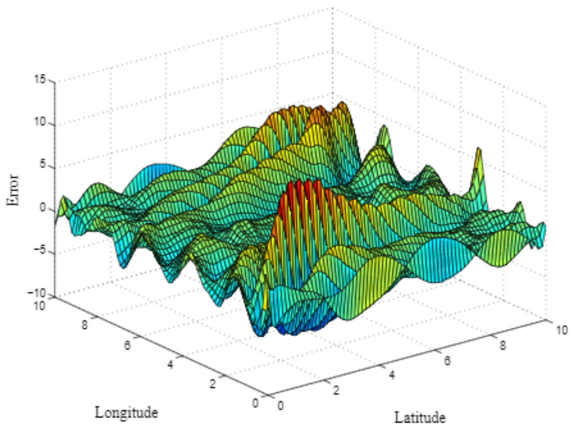
Though the approximations in figure 3.9 shows a degradation in the approximation through increasing allowed error (by increasing γ), it is interesting to notice what happens after a trade-off point as seen in figure 3.10.

Figure 3.11 plots the number of coefficients and the standard deviation of the approximation formed with respect to γ.

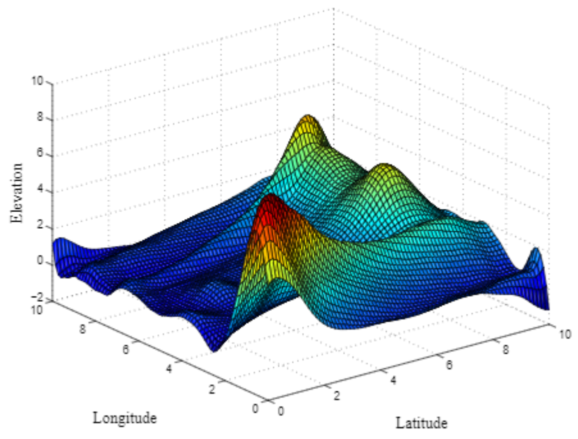
Though a reasonable model has been formed by finding the minimum number of coefficients in the model approximation, it is seen that the optimization statement will encourage the error to increase linearly while varying γ until eventually no coefficients are weighted. For a better



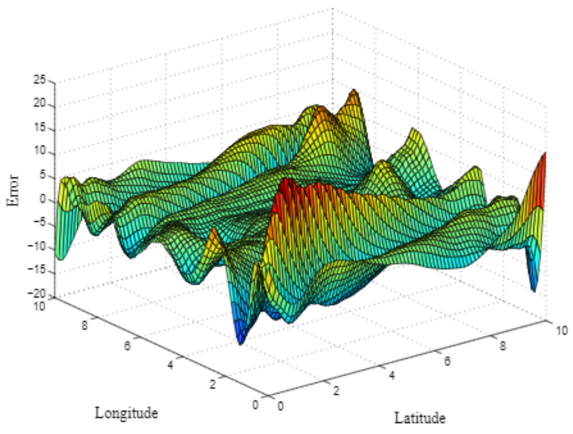
(a) Approximation $\gamma = 1.25$



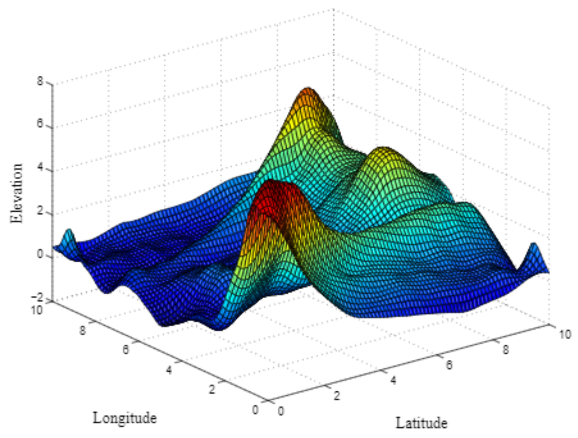
(b) Error Surface for $\gamma = 1.25$



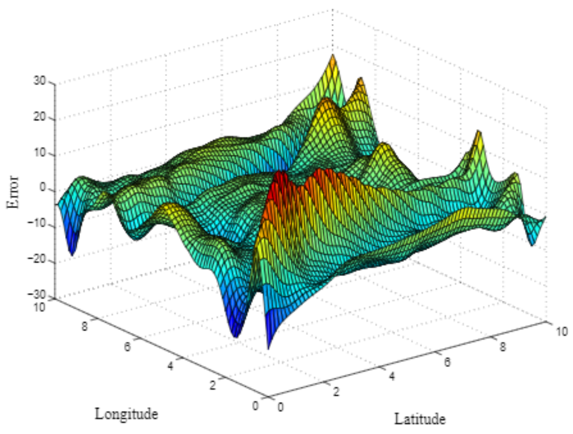
(c) Approximation $\gamma = 2.25$



(d) Error Surface for $\gamma = 2.25$

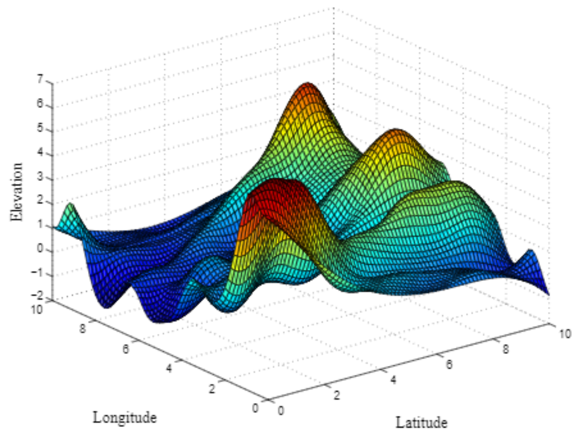


(e) Approximation $\gamma = 3.25$

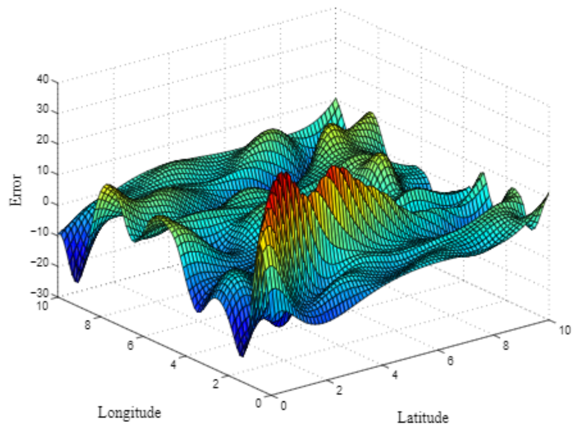


(f) Error Surface for $\gamma = 3.25$

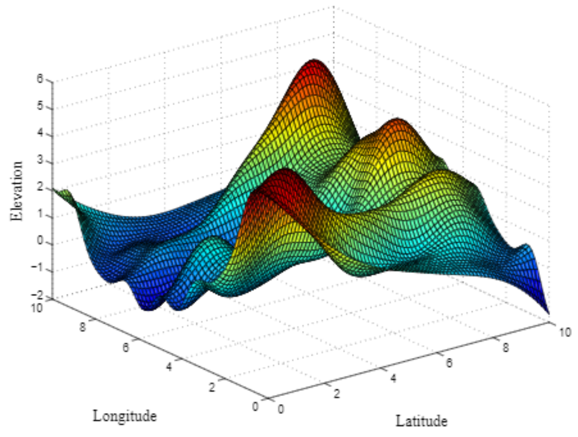
Figure 3.6: With low values of γ, the approximate adequately re-constructs the signal, while drastically decreasing the number of basis functions used in the model.



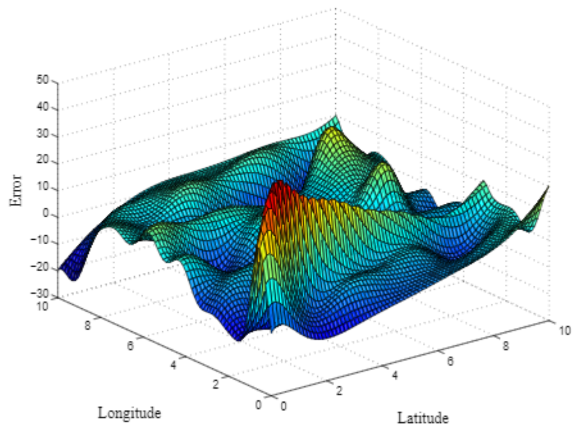
(a) Approximation $\gamma = 4.25$



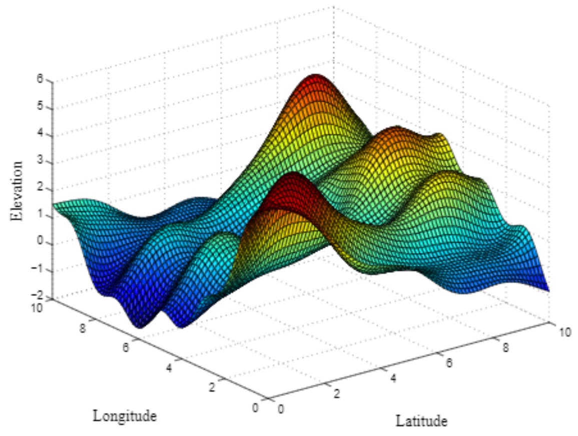
(b) Error Surface for $\gamma = 4.25$



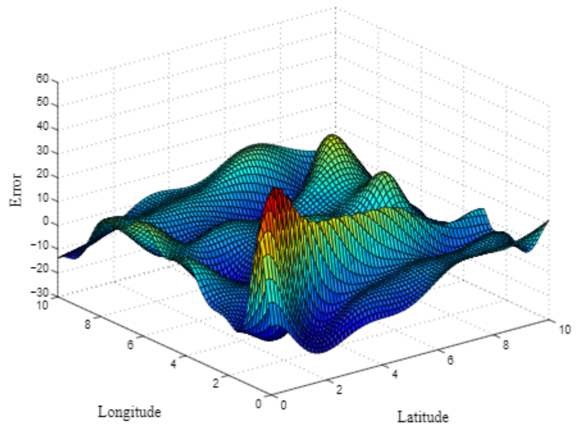
(c) Approximation $\gamma = 4.75$



(d) Error Surface for $\gamma = 4.75$

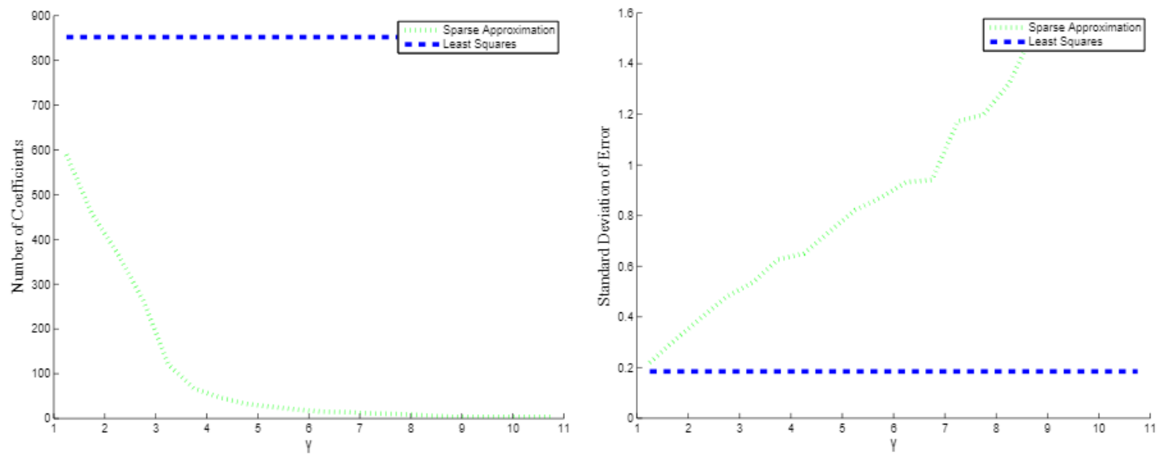


(e) Approximation $\gamma = 5.25$



(f) Error Surface for $\gamma = 5.25$

Figure 3.7: Around the limit point, an increase in γ produces dramatic changes to the approximation model.



(a) Cardinality vs γ (b) Standard Deviation of Error vs γ

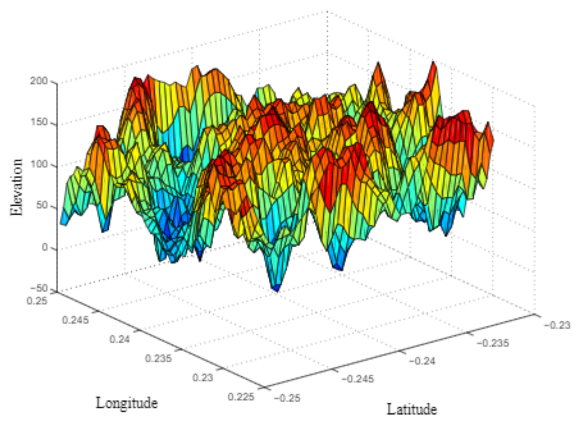
Figure 3.8: The results indicate that after $\gamma = 3.75$, there is relatively no change in the model.

approximation development, alternate statements are investigated.

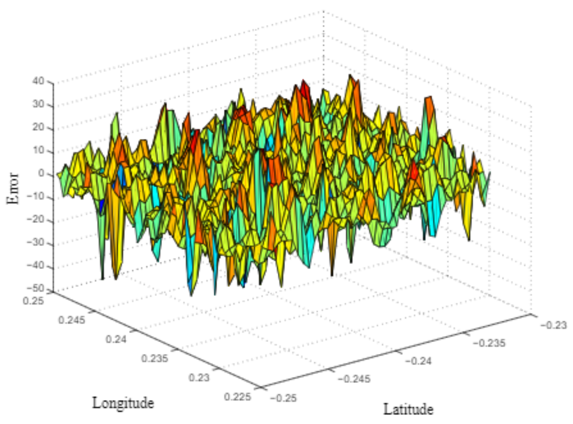
3.5 Problem 2

In this problem, the 2-norm error and the number of coefficients are minimized simultaneously. In this problem γ enforces the emphasis that is placed on minimizing the 1-norm of the coefficients with respect to the 2-norm error of the approximation. The optimization statement is posed as follows:

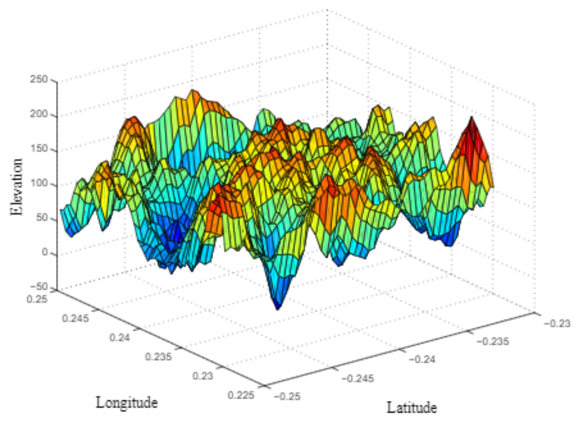
```
cvx_begin
    variable coeffs(y)
    minimize( norm(coeffs,1)*gamma+norm(A*coeffs-b,2));
cvx_end
```



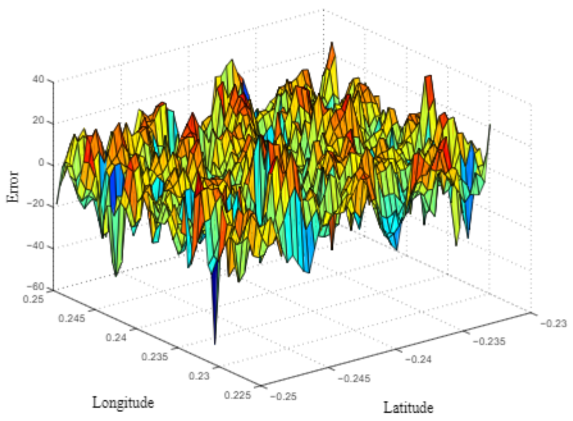
(a) $\gamma = 1.25$



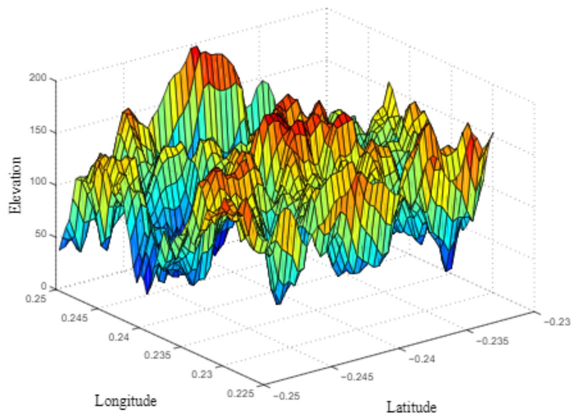
(b) Error Surface for $\gamma = 1.25$



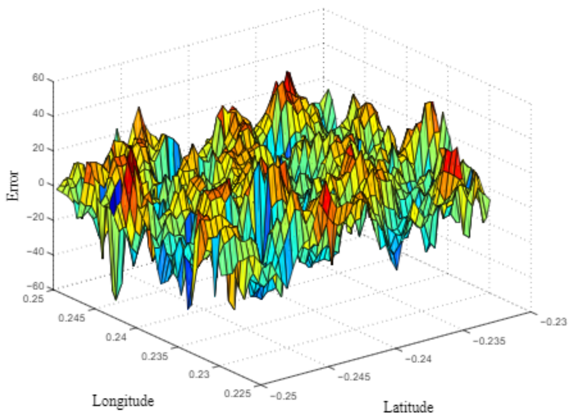
(c) $\gamma = 1.5$



(d) Error Surface for $\gamma = 1.5$

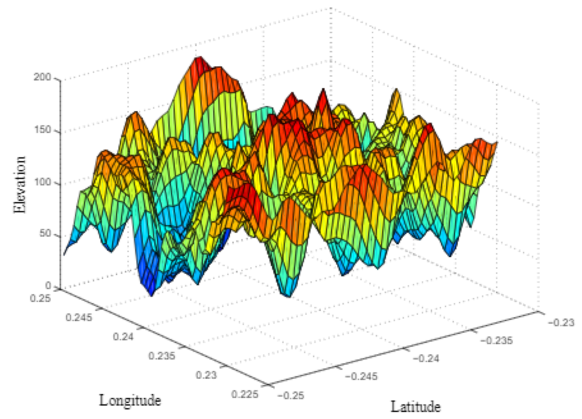


(e) $\gamma = 2$

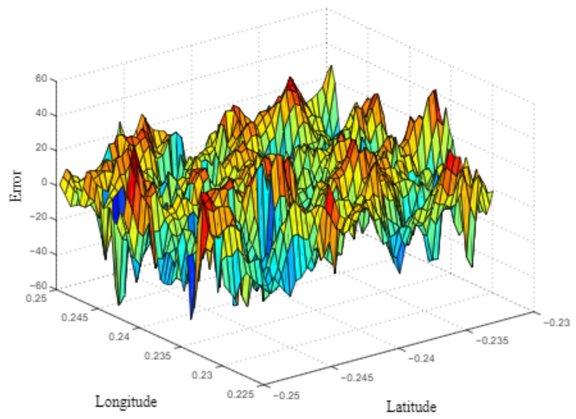


(f) Error Surface for $\gamma = 2$

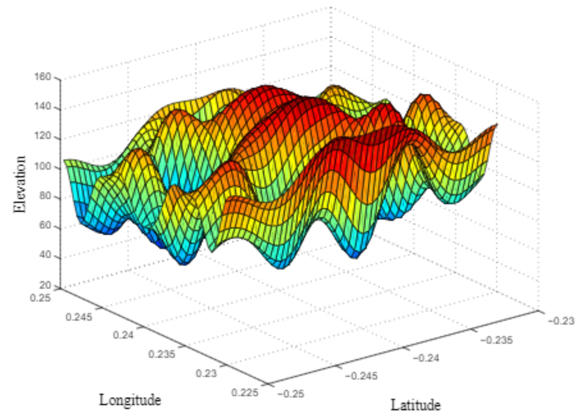
Figure 3.9: For low values of γ few basis functions are eliminated and a low-error approximation is formed.



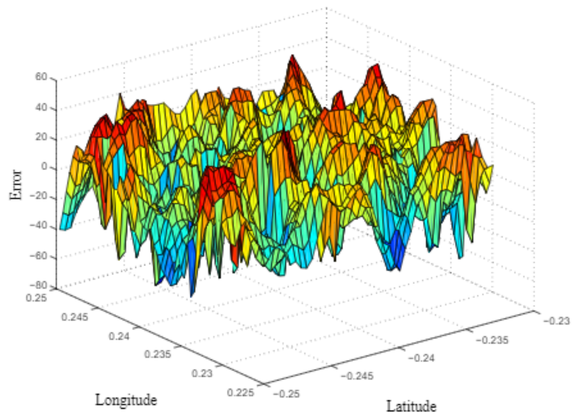
(a) $\gamma = 2.25$



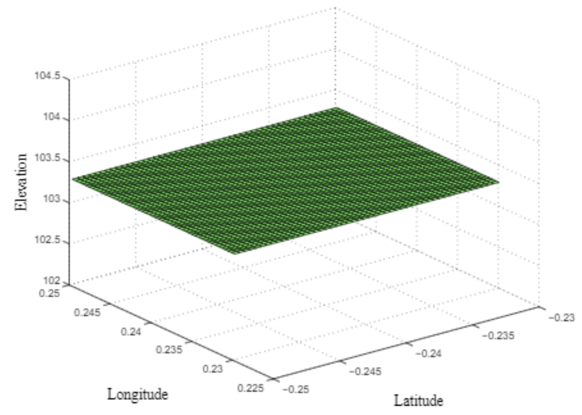
(b) Error Surface for $\gamma = 2.25$



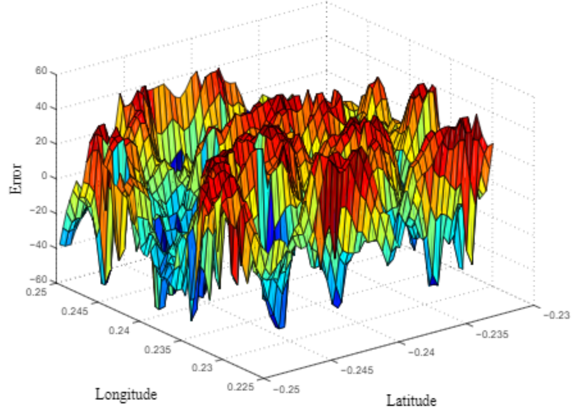
(c) $\gamma = 2.75$



(d) Error Surface for $\gamma = 2.75$

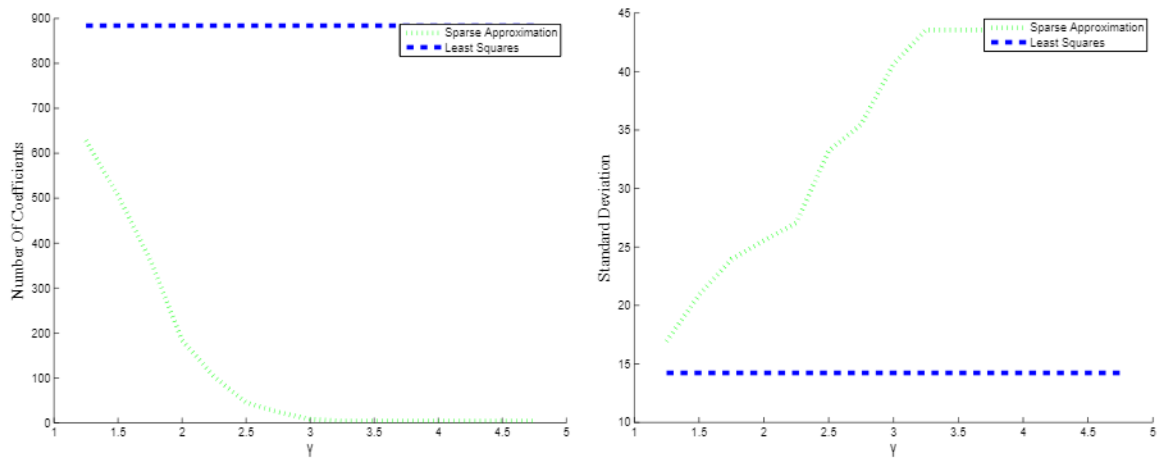


(e) $\gamma = 3.25$



(f) Error Surface for $\gamma = 3.25$

Figure 3.10: Around the trade-off point, an increase in γ produces dramatic changes to the approximation model.



(a) Cardinality vs γ (b) Standard Deviation of Error vs γ

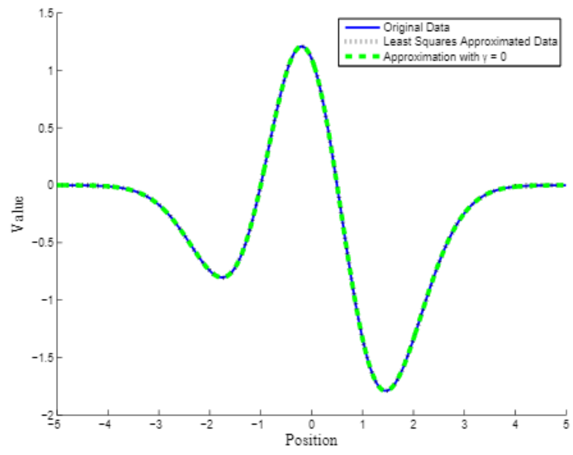
Figure 3.11: The results indicate that after $\gamma = 2$, there is relatively no improvement in the model.

3.5.1 Example 1.2: 1-D benchmark function

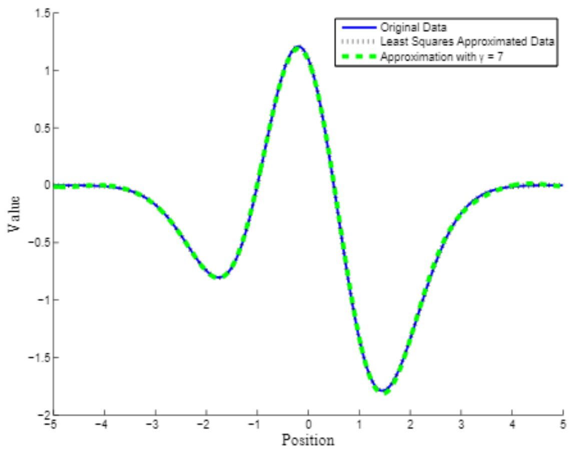
To test the development of sparsity, and to see its effects, a 1-Dimensional example is revisited. Varying γ, it is seen the effects on the approximation.

From figure 3.12, it is clear to see that after $\gamma = 7$, the approximation becomes noticeably less accurate. In figure 3.13, it is seen that the number of coefficients has been cut in half, yet the standard deviation of the error is still very low.

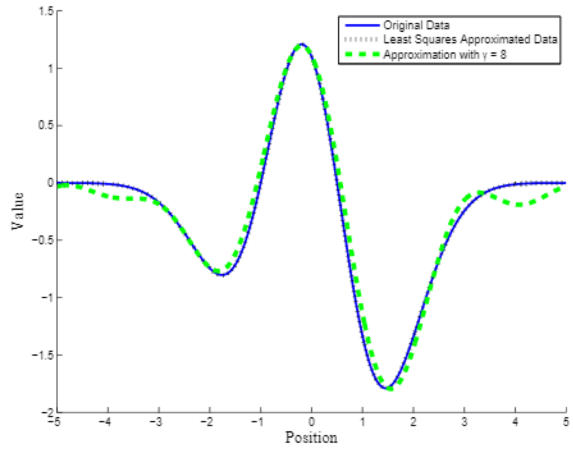
Comparatively, it is easier to judge the which basis functions can be immediately rejected, by increasing γ only slightly to achieve a sparse approximation. Though this became clear for a simple model, more complicated models will be tested to see if this same advantage is true if the complexity of the signal is increased.



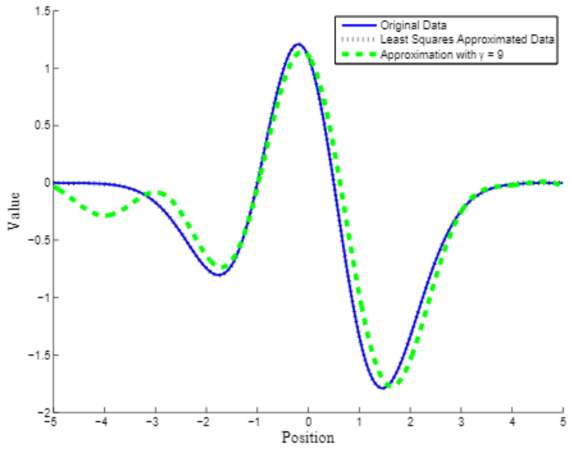
(a) $\gamma = 0$



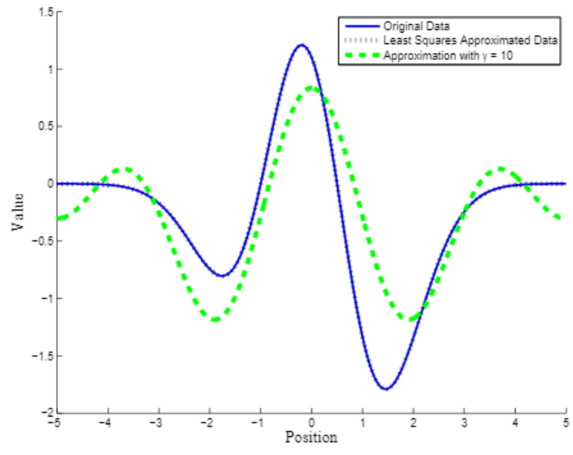
(b) $\gamma = 7$



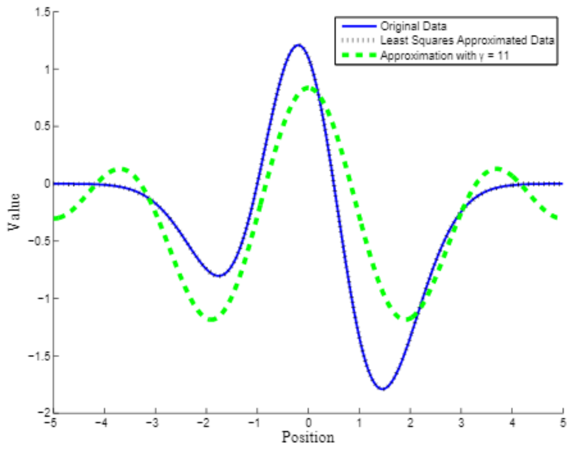
(c) $\gamma = 8$



(d) $\gamma = 9$

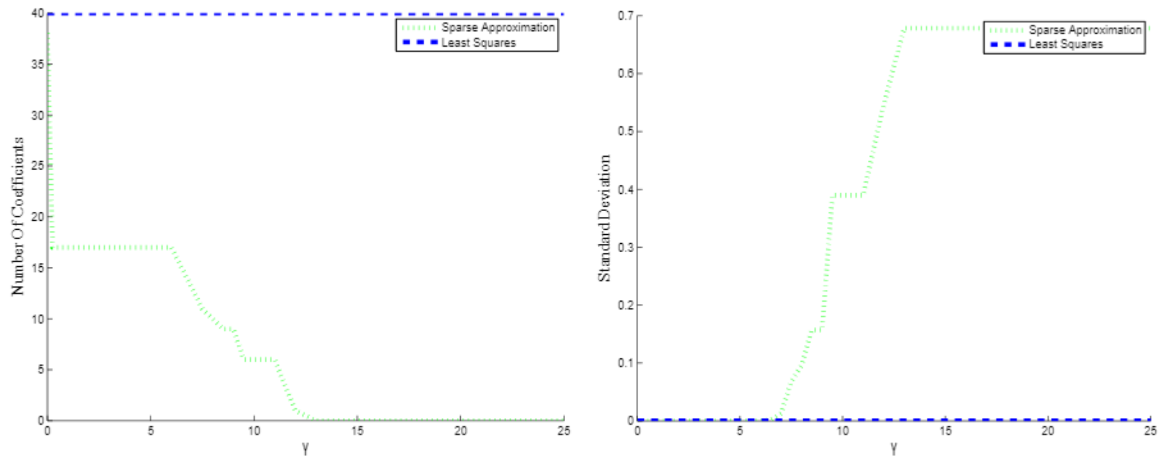


(e) $\gamma = 10$



(f) $\gamma = 11$

Figure 3.12: By increasing γ, fewer basis functions are used to approximate the signal.



(a) Number of Coefficients vs γ (b) Standard Deviation of Error vs γ

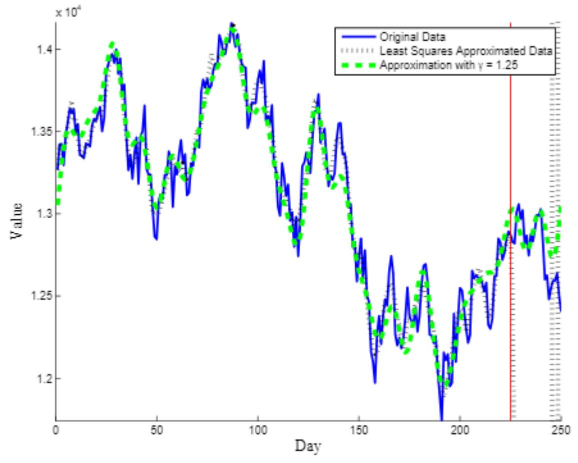
Figure 3.13: It is seen that an adequate model is made with 15 coefficients, found with $\gamma = 4$.

3.5.2 Example 2.2: 1-D Stock Market Model

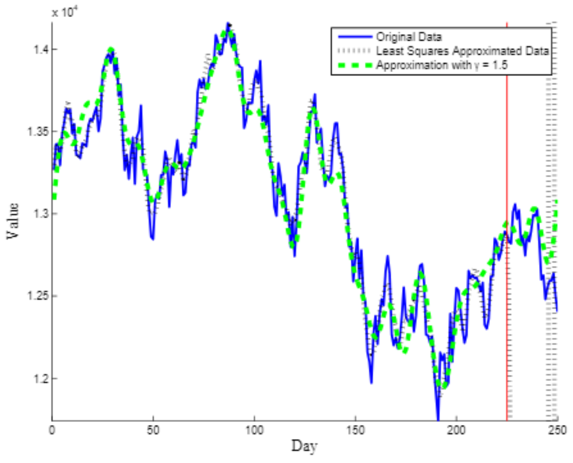
The second model, is again the stock market model, which approximates the Dow Jones Industrial average. In this example, a set of data is used to develop a model, which forms a prediction signal. As seen in figure 3.14, there are immediate advantages to a sparse approximation.

As illustrated in figure 3.15, the least squares poorly developed a prediction model, while the sparse approximation reasonably developed a prediction model. In fact, even with 7 coefficients, a reasonable approximation and prediction was formed.

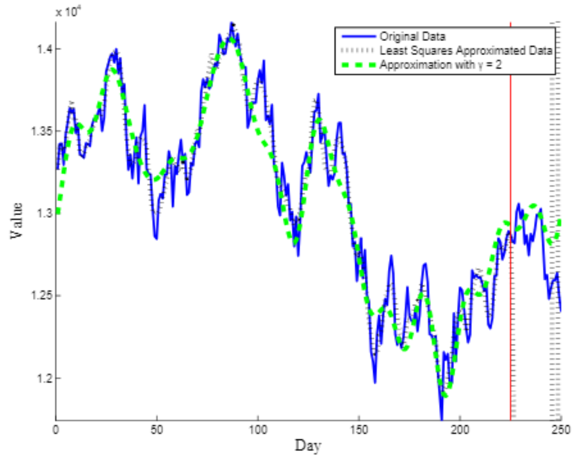
Sparse approximations have shown to reasonably develop approximation models for one dimensional signals. For less complicated models (as seen in example 1), higher values of γ are used to decrease the number of coefficients, but with a more complicated model (such as example 2), lower values of γ can adequately decrease the number of coefficients used to build an approximation model.



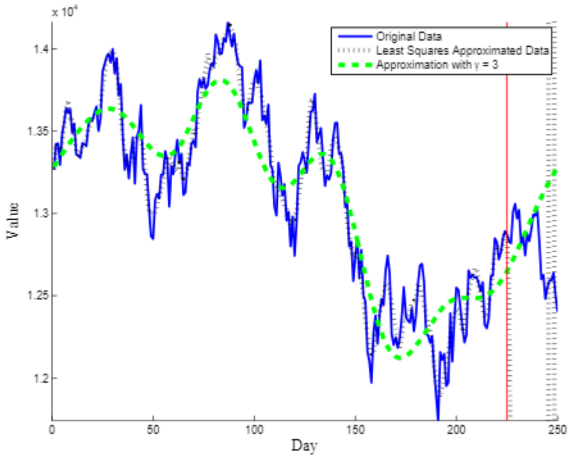
(a) $\gamma = 1.25$



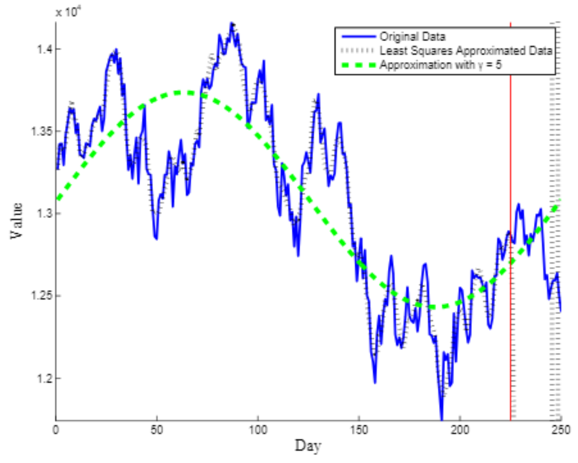
(b) $\gamma = 1.5$



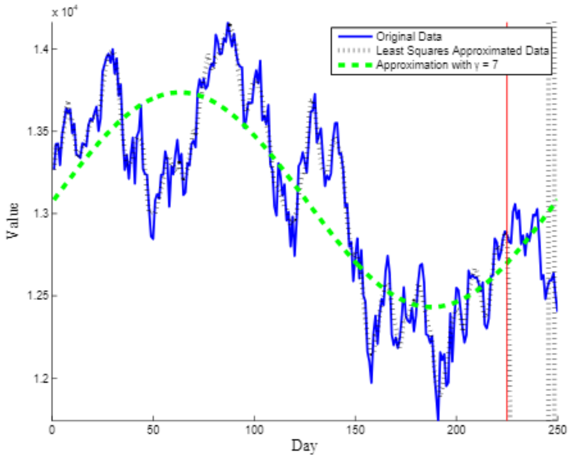
(c) $\gamma = 2$



(d) $\gamma = 3$

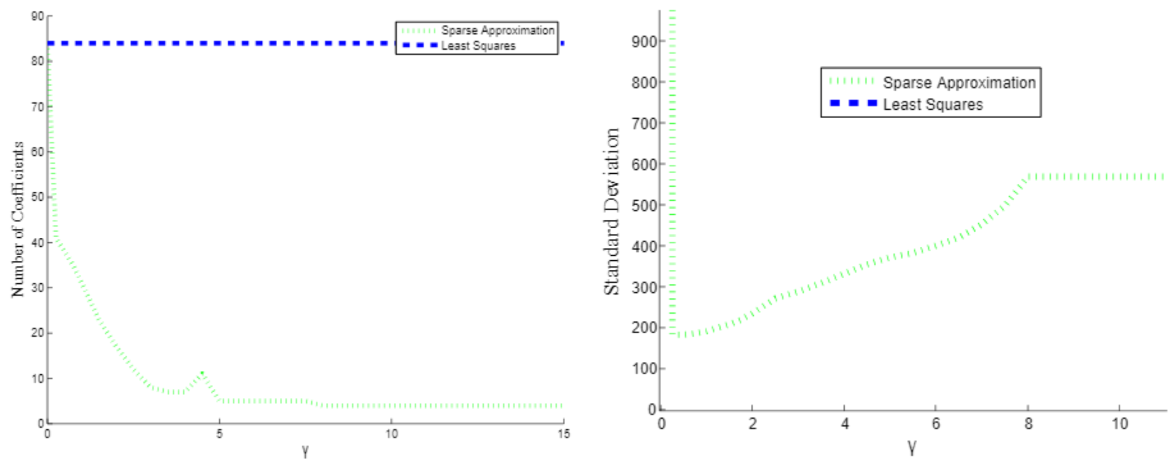


(e) $\gamma = 5$



(f) $\gamma = 6$

Figure 3.14: By increasing γ, fewer basis functions are used to approximate the signal.



(a) Number of Coefficients vs γ (b) Standard Deviation of Error vs γ

Figure 3.15: It is seen that with the immediate addition of sparsity constraints, the number of coefficients is cut in half and the sparse solution with the lowest error is formed.

3.5.3 Example 3.2: 2-D "Monster Function"

Again multi-dimensional signals will be investigated. The results from low-γ approximations are shown in figure ??, in which lower values of γ produce a reasonable approximation.

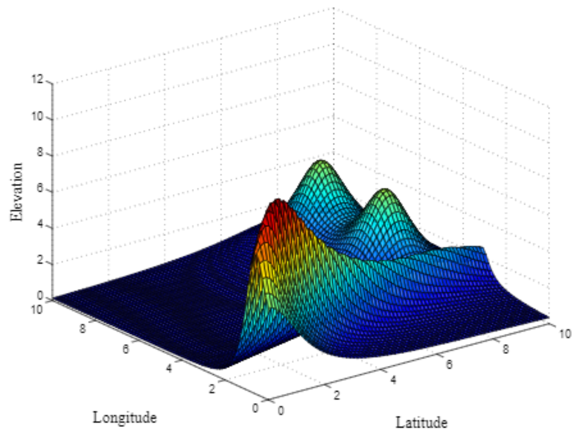
As seen in figures 3.16, lower values of γ form a very good approximation, while higher values of gamma, greater than .25, utilize too few basis functions to form a reasonable approximation.

As seen in figure 3.17, the number of coefficients does not increase dramatically through increasing γ beyond 1, while the standard deviation of the error continues to increase.

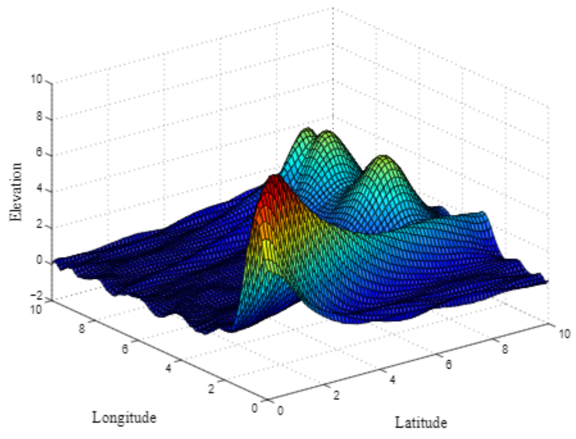
3.5.4 Example 4.2: 2-D Moon Terrain

Again, this is the same data as seen in figure 2.13. In figure 3.18, it is seen that for low values of γ, a reasonable approximation is formed.

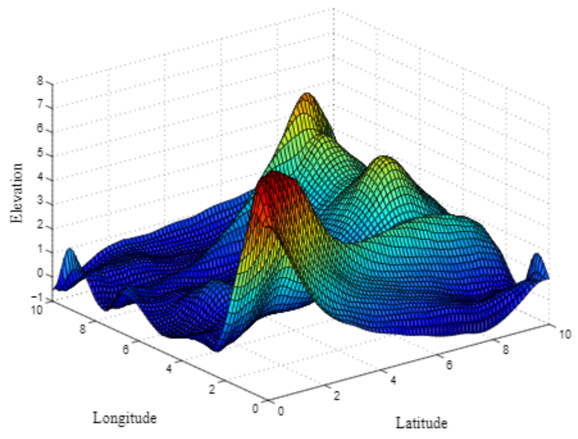
Figure 3.18 that by increasing γ, only the dominant features of the moon's geometry are captured. Figure 3.19 plots the number of coefficients and the standard deviation of the ap-



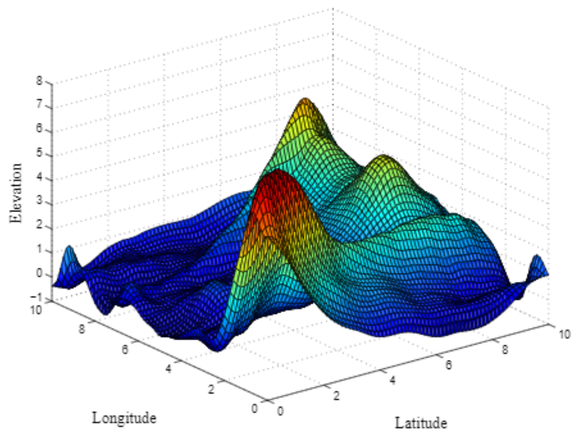
(a) Original



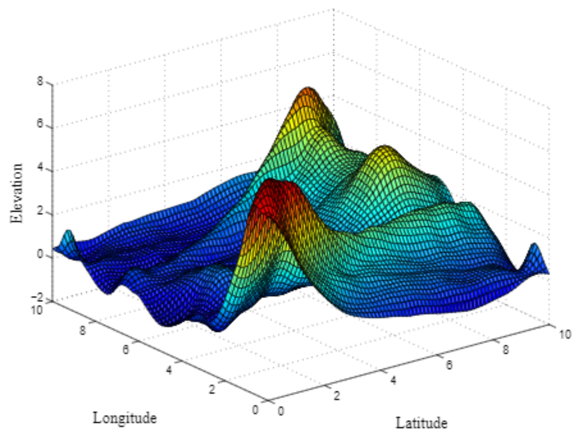
(b) Approximation for $\gamma = 0$



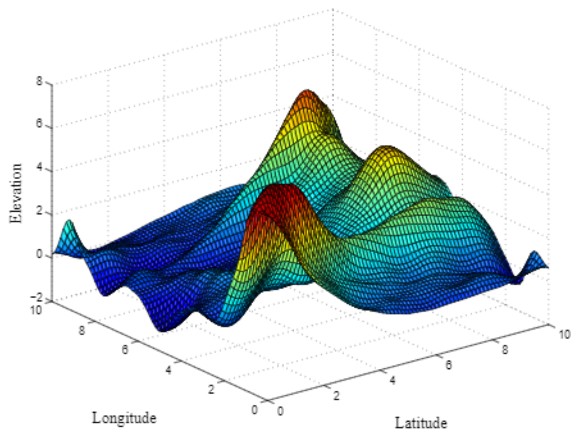
(c) Approximation for $\gamma = .25$



(d) Approximation for $\gamma = .5$

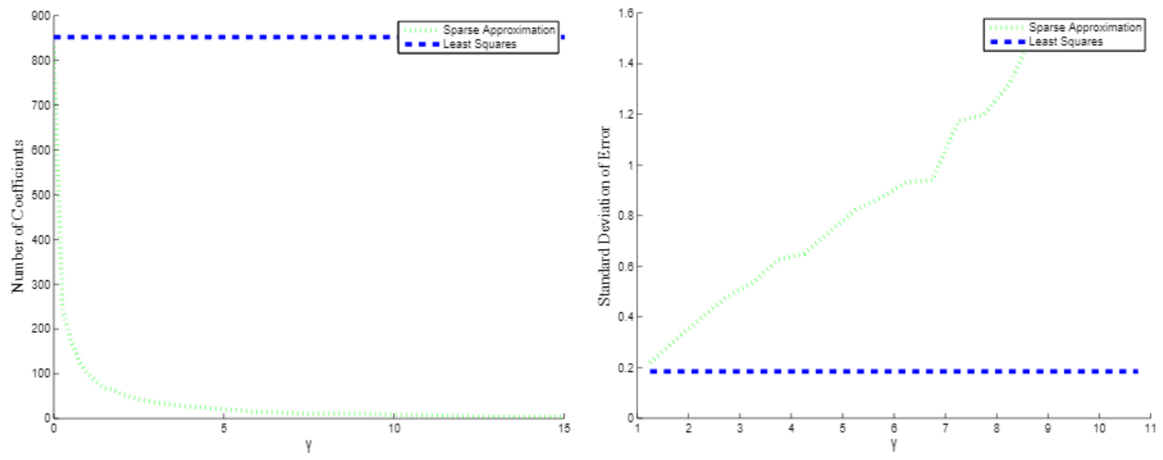


(e) Approximation for $\gamma = .75$



(f) Approximation for $\gamma = 1$

Figure 3.16: With low values of γ, the approximate adequately re-constructs the signal, but with a noticeable error as γ increases.



(a) Cardinality vs γ (b) Standard Deviation of Error vs γ

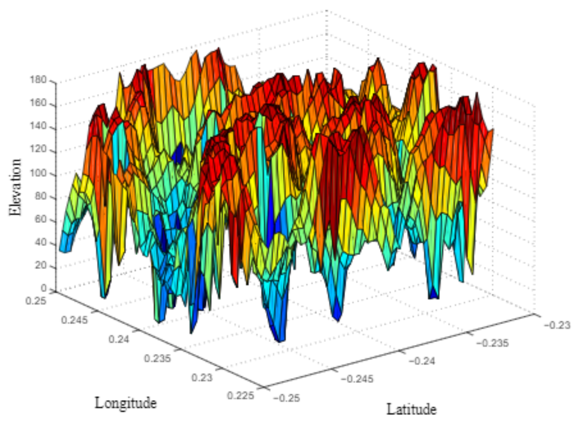
Figure 3.17: The results indicate that after $\gamma = 1$, there is relatively no change in the model.

proximation formed with respect to γ.

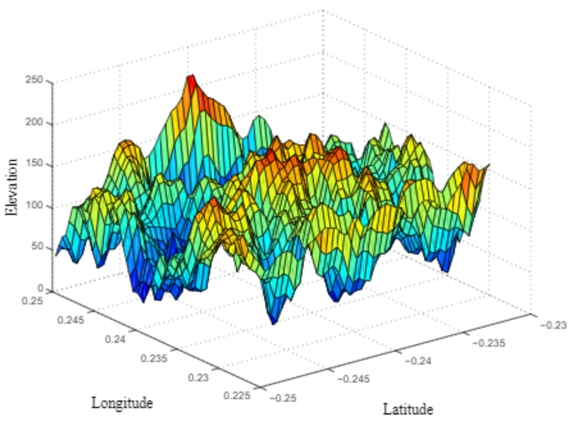
The problem with a more complicated model is that a slight increase in γ will cause a large increase in the standard deviation of the error. This is because there are so many functions that compose the original data, that eliminating even the less significant ones contributes much error.

3.6 Conclusion

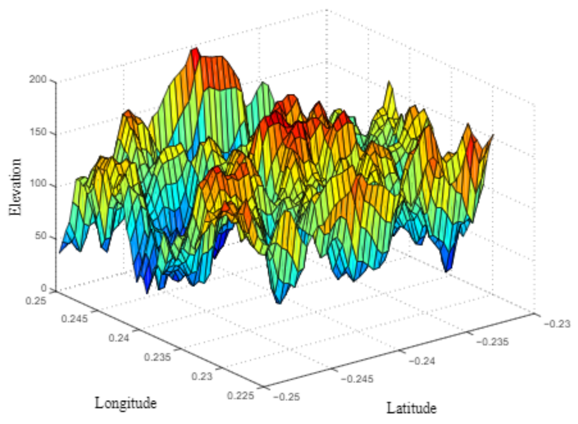
It has been demonstrated that just by adding $\|x\|_1$ to the minimization routine, a certain amount of sparsity is enforced. A similar advantage is found in Chapter 4, where sparsity is promoted simply through regionally approximating the signal. Though both optimization methods discussed can further promote sparsity, the second has the advantage of retaining $\|A \cdot x - b\|_2$ in the minimization routine, rather than as a constraint. This can allow for a trade-off to be pre-selected, rather than found heuristically.



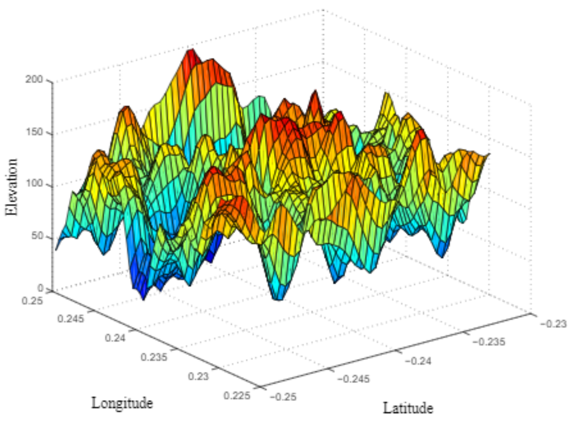
(a) Original



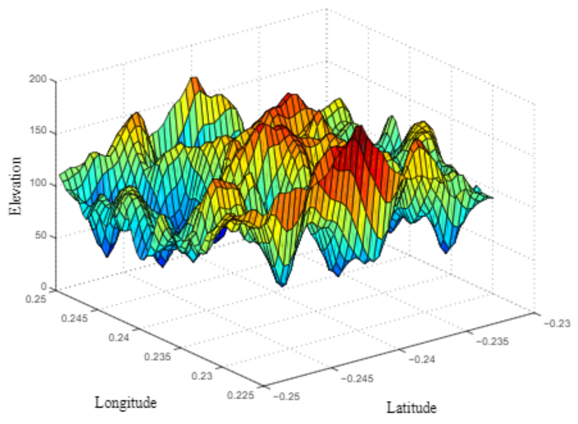
(b) $\gamma = 0.25$



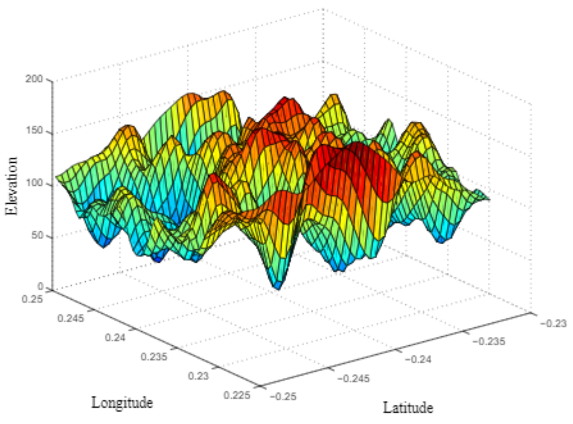
(c) $\gamma = 0.5$



(d) $\gamma = 0.75$

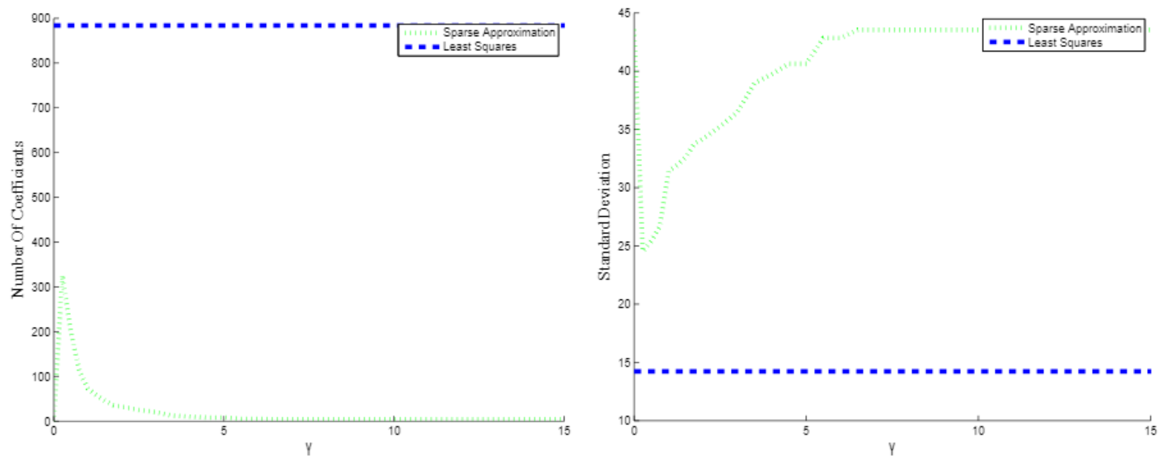


(e) $\gamma = 1$



(f) $\gamma = 1.25$

Figure 3.18: By increasing γ slightly, noticeably less basis functions are used to create the model.



(a) Cardinality vs γ (b) Standard Deviation of Error vs γ

Figure 3.19: The results indicate that after $\gamma = 2$, there is relatively no improvement in the model.

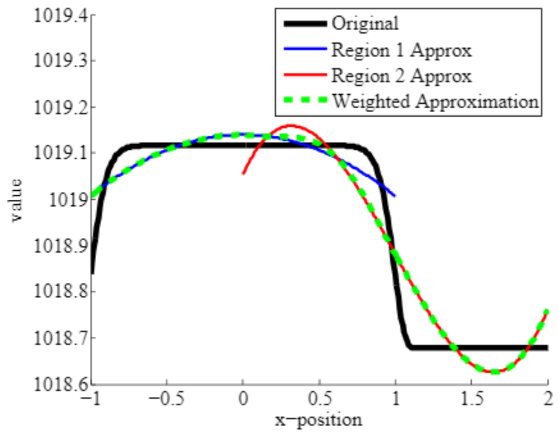
Chapter 4

Global-Local Approximation Mapping

4.1 Introduction to Global-Local Mapping

I chapter 3, a sparse solution gave way to an approximation that utilized only the most dominant basis functions. This was enforced in tradeoff with error. Fewer basis functions led to a greater residual. In this section, the goal is to maintain fewer basis functions, but to decrease the residual through regional approximation. If regional approximation is successful in accomplishing this task, even fewer basis functions may be utilized, and sparsity may be enforced through global-local approximation.

Signal approximation and reconstruction has a variety of different approaches. Though an entire signal *could* be reconstructed by a single approximation, this would be a computationally taxing process for larger data sets and may neglect regional characteristics [15] if an insufficient quantity of basis functions is used. An alternative method has been proposed in which the signal is regionally approximated [16], and then sequentially combined to reconstruct [17] an approximated signal. The advantage of a regional approximation is that fewer basis functions are needed to achieve the same level of global error. This is seen in figure 4.1, where low-order polynomials are used to regionally approximate a signal.



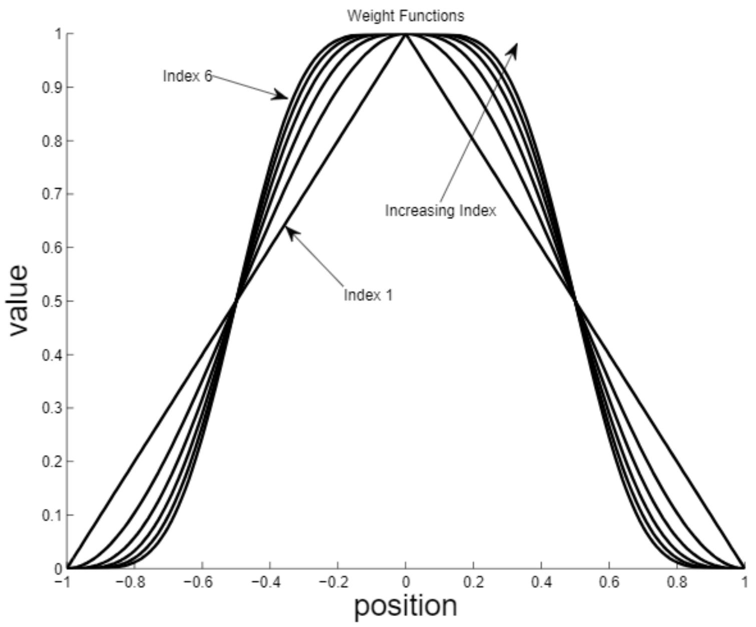
(a) Sample Regional Approximation

Figure 4.1: This complicated signal is approximated using two low-order regional approximations.

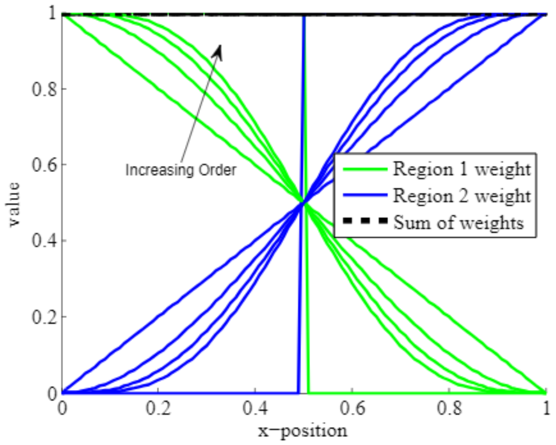
As seen in figure 4.1, the two regional approximations are blended together to form a global approximation [18]. This is accomplished using weighting/smoothing functions [19], whose derivation is described in Appendix D, as first proposed by Singla and Junkins [5]. Figure 4.2 shows how two weighting functions blend adjacent regions for a 1-D signal.

Figure 4.2 shows some important features [20] of the weighting functions. First, for higher order weighting functions the higher order derivatives are zero at the center of the region and at the limits of the region. Second, the weighting evaluates to 1/2, half way away from the center of the region. Third, and most important, the sum of all adjacent regions is always 1. As seen in figure 4.3,

Higher dimensions will not be reviewed in this chapter, but the same smoothing algorithm still applies. Figure 4.4 illustrates how the number of approximations increase exponentially (as an exponent of 2 in this case) based on the dimensionality of the signal.

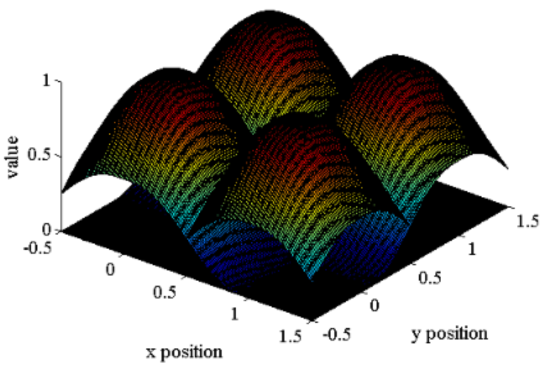


(a) Basic 1-D Smoothing Functions

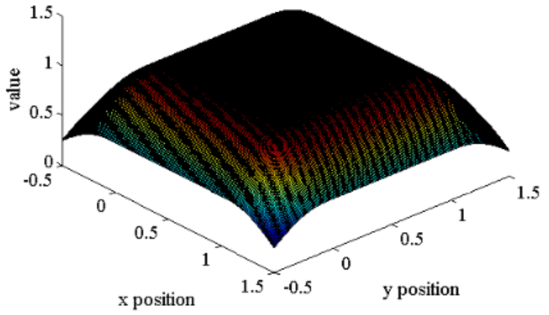


(b) 1-D Weighting Functions combined

Figure 4.2: As a weighting function moves away from the center of the region, its value approaches zero. Meanwhile, the sum of adjacent weighting functions is always equal to 1.



(a) 2-D Weighting Functions



(b) Sum of four adjacent weighting functions

Figure 4.3: In two dimensions, the key properties of the weighting functions are still valid.

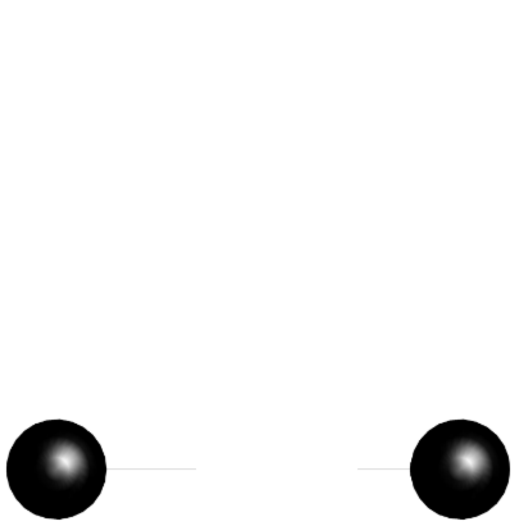
Since it is difficult to display signal reconstructions for dimensionality greater than 2, this chapter only re-visits Examples 1-4.

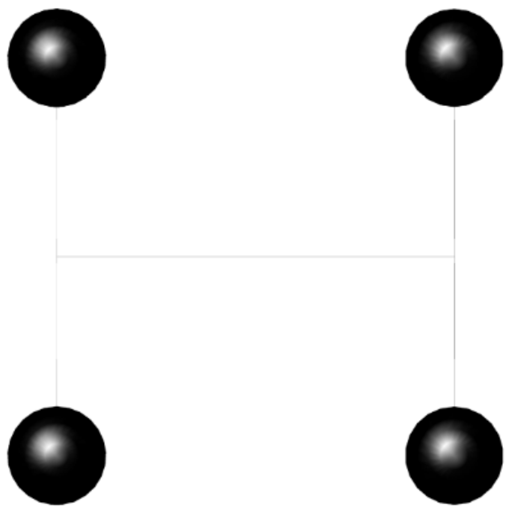
4.2 Examples

The following examples use the second problem statement from chapter 3. This problem statement appears in the form

```
cvx_begin
    variable coeffs(y)
    minimize( norm(coeffs,1)*gamma+norm(A*coeffs-b,2));
cvx_end
```

This optimization problem minimizes the number of coefficients as well as the 2-norm of the error. The value of γ is set very low ($\gamma = .01$) since it is only required that sparsity exist, as the sum of the coefficients will have more weight against the 2-norm will change as the regional


(a) Two adjacent regional centers


(b) Four adjacent regional centers

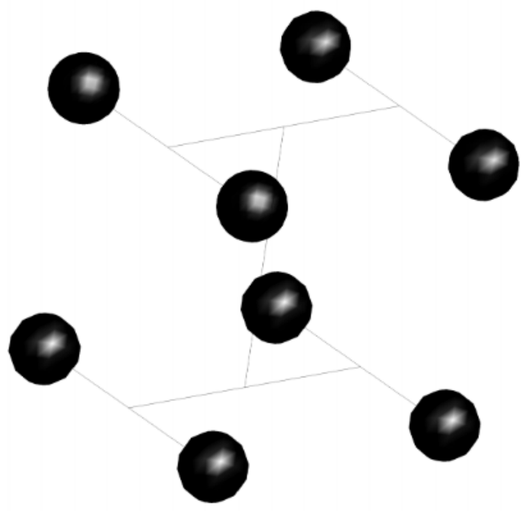

(c) Eight adjacent regional centers

Figure 4.4: In two dimensions, four centers are used to approximate a region. In three dimensions, eight centers are used to approximate a region. In n-D space, 2^n centers are used to approximate a region.

size decreases in number of data points and the number of regions increase.

The results are given in terms of cardinality vs number of regions, and standard deviation of error vs number of regions. The cardinality is the global cardinality - how many of the original over-complete dictionary were used. If a function is re-used, it contributes once to the cardinality count. This helps recognize dominant regional functions across the global signal. Standard deviation of error is also a global measure. The error is formed by taking the difference between the global approximation and the original data. This analysis ensures that results of varying dimensionality are comparable.

4.2.1 Example 1: 1-D benchmark function

To test the development of sparsity, and to see its effects, a 1-Dimensional example is used. This is the same function as seen in section 2.2.1 of $1.1 \cdot (1 - x - 2 \cdot x^2) \cdot e^{-\frac{x^2}{2}}$. Varying the number of regions, it is seen the effects on the approximation in figure 4.5.

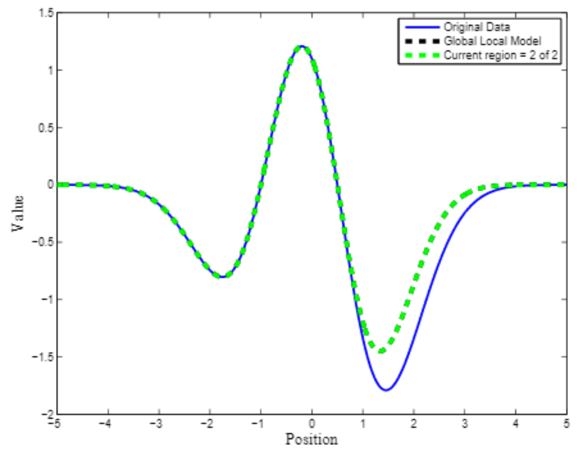
The continued trend in decreasing the number of basis functions is seen in figure 4.6. Increasing the number of regions effectively enforces sparsity.

Figure 4.6 indicates that at 6 regions and 20 basis functions, the model begins to outperform the least squares approximation with the over-complete dictionary.

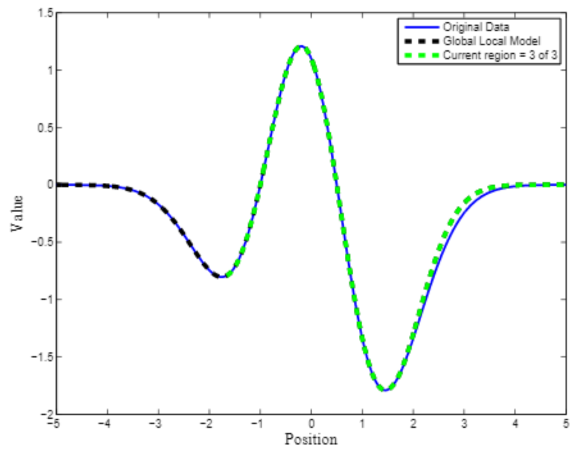
4.2.2 Example 2: 1-D Stock Market Model

Using a regional model of one dimension, it is no longer realistic to form a prediction of the Dow Jones Industrial Average. Instead, this analysis regionally approximates the signal (see figure 4.7) in order to find dominant or repeated basis functions as shown in figure 4.8.

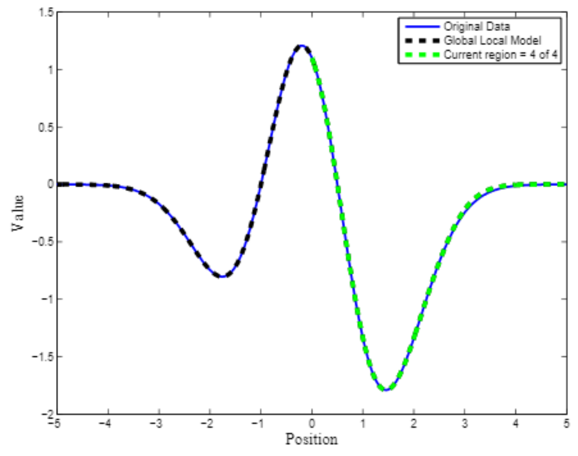
The signal was approximated in figure 4.7, but as seen in figure 4.8, increasing the number



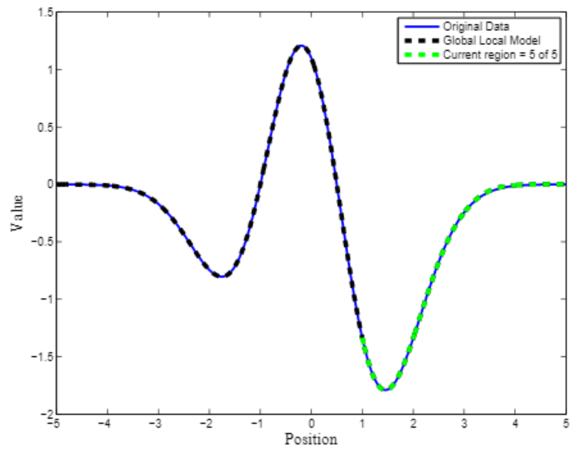
(a) 34 basis functions and 2 regions



(b) 35 basis functions and 3 regions

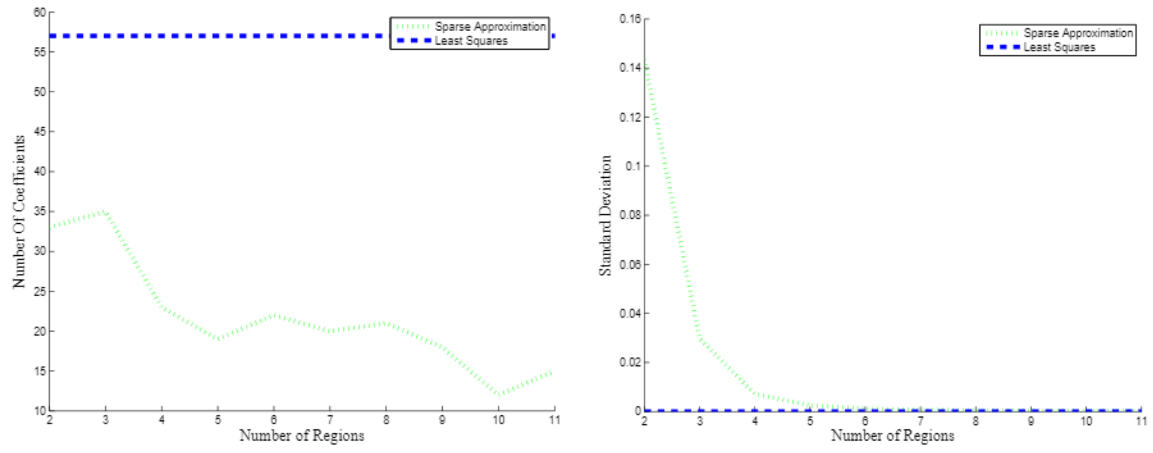


(c) 23 basis functions and 4 regions



(d) 19 basis functions and 5 regions

Figure 4.5: Increasing the number of regions allows for a better approximation and a smaller global dictionary.



(a) Number of Coefficients vs Number of Regions (b) Standard Deviation of Error vs Number of Regions

Figure 4.6: Through increasing the number of regional approximations, a sparse approximation can be formed with less error than that of the least squares approximation. The sparsity of the approximation is further improved through increasing the number of regions.

of regions only decreased the error and not the number of basis functions used.

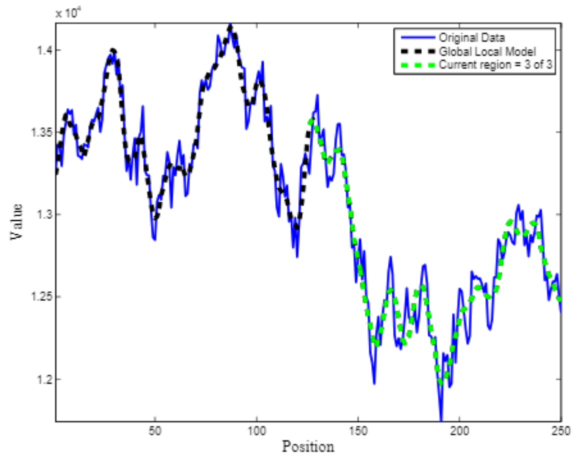
This lends to the conclusion that for complicated or noisy signals, sparsity cannot be enforced easily by increasing the number of regional approximations.

4.2.3 Example 3: "Monster Function"

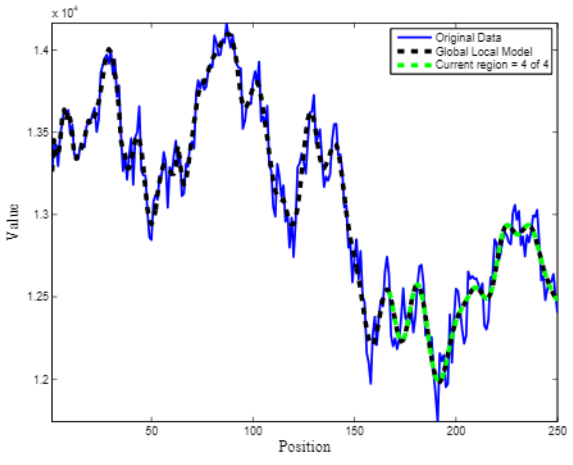
The monster function is a simple 2-D benchmark problem that will show the relationship between sparsity and number of regional approximations. Sample approximations are shown in figure 4.9

From figure 4.9 would appear that the approximation improves quickly by increasing the number of regions, but only to a point. This is because so much of the signal is the same. This is confirmed by figure 4.10.

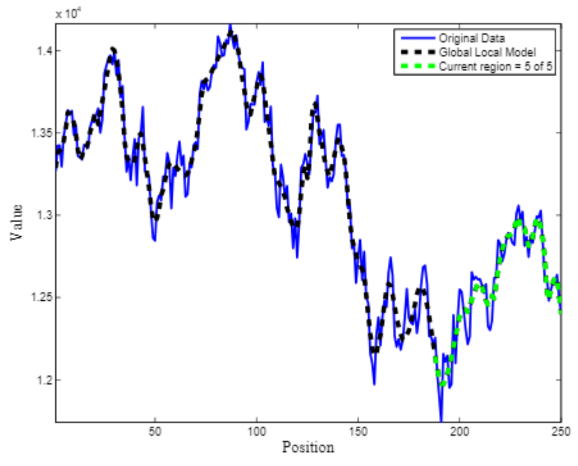
For simple 2-D models, a tradeoff is quickly found as increasing the number of regions does



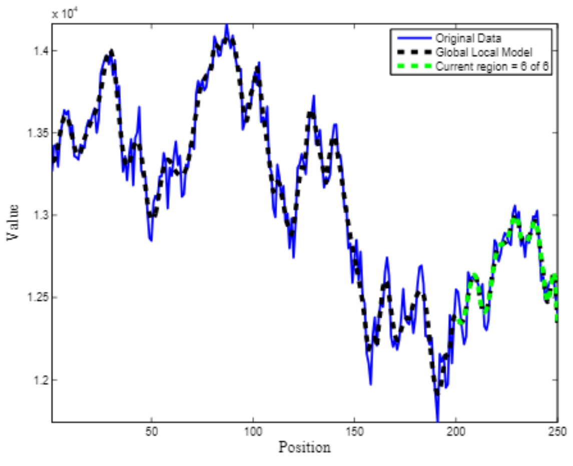
(a) 28 basis functions and 3 regions



(b) 20 basis functions and 4 regions

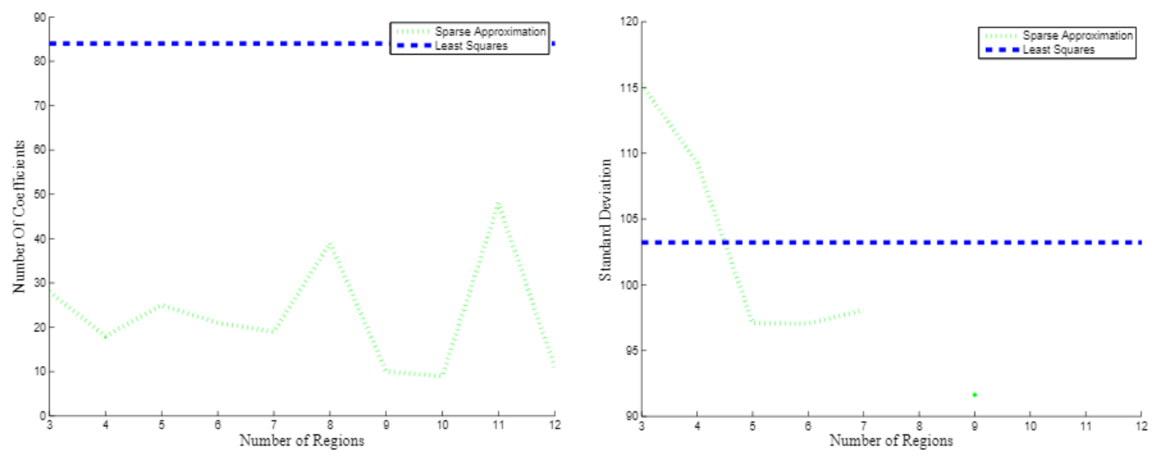


(c) 25 basis functions and 5 regions



(d) 22 basis functions and 6 regions

Figure 4.7: Due to the complexity of this signal, sparsity seems only enforced by the optimization statement alone, since many of the same basis functions are cyclic and are dominant throughout.



(a) Number of Coefficients vs Number of Regions (b) Standard Deviation of Error vs Number of Regions

Figure 4.8: Through combining regional approximations it is seen that a better approximation can be formed than the global least squares. The number of coefficients, on the other hand, shows little improvement through increasing the number of regions. This is due to the complexity of the signal in question.

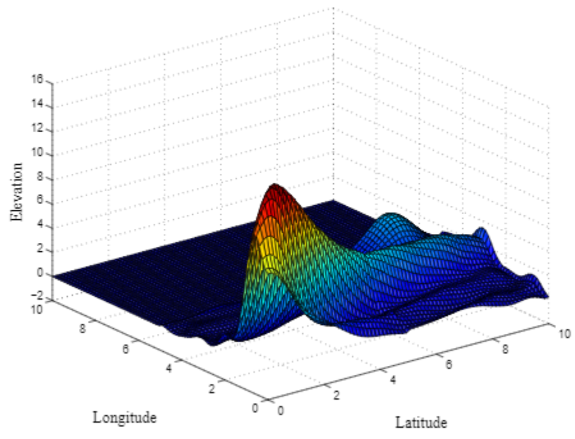
little to enforce sparsity or improve error.

4.2.4 Example 4: Moon Data

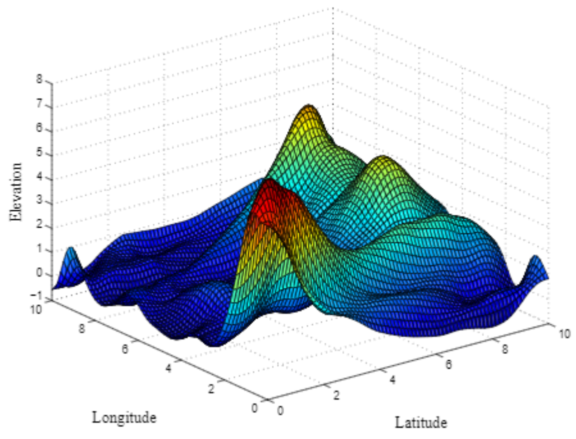
The moon terrain is a complicated 2-D problem that will show the advantage of using regional approximations.

Figures 4.11 and 4.12 show how even with using 5 basis functions, the approximation improves through increasing the number of regions. This is further illustrated in figure 4.13.

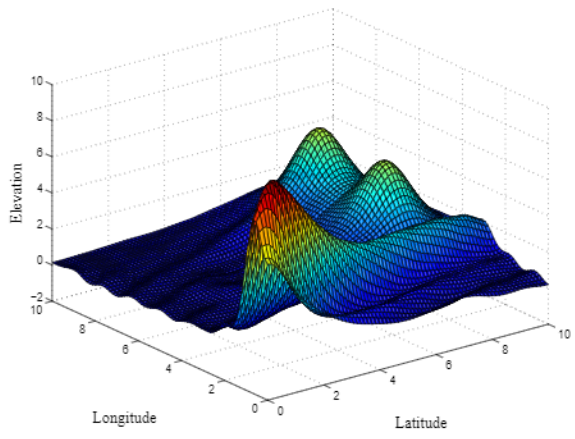
Earlier it was stated that for complicated or noisy signals, the enforcement of sparsity was not enforced through regional approximation. Now that a low-noise complicated signal has been approximated, it is seen that sparsity is enforced through regional approximation.



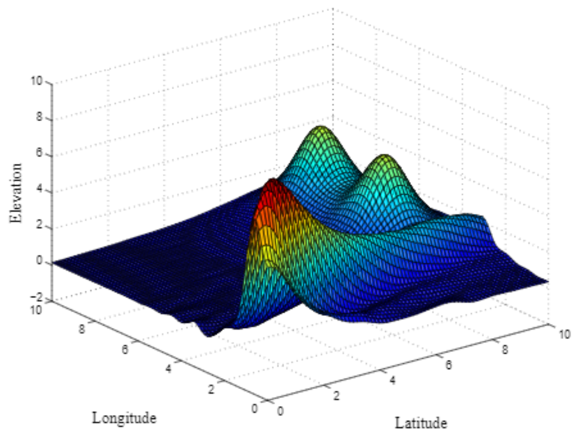
(a) 4 regions



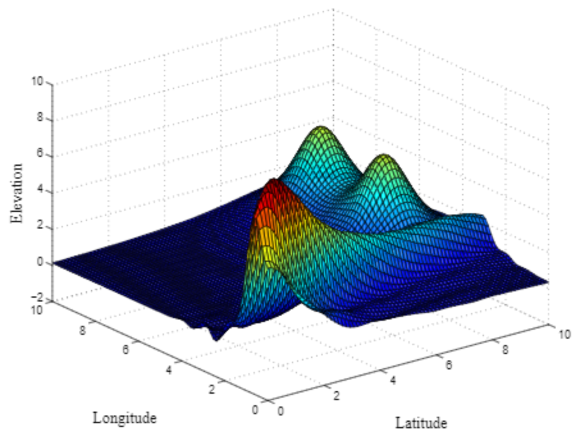
(b) 9 regions



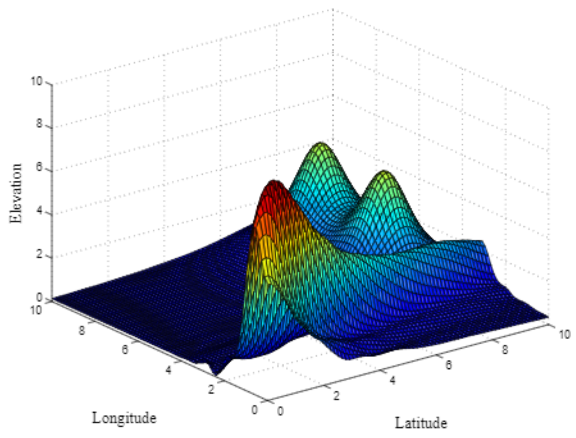
(c) 16 regions



(d) 25 regions

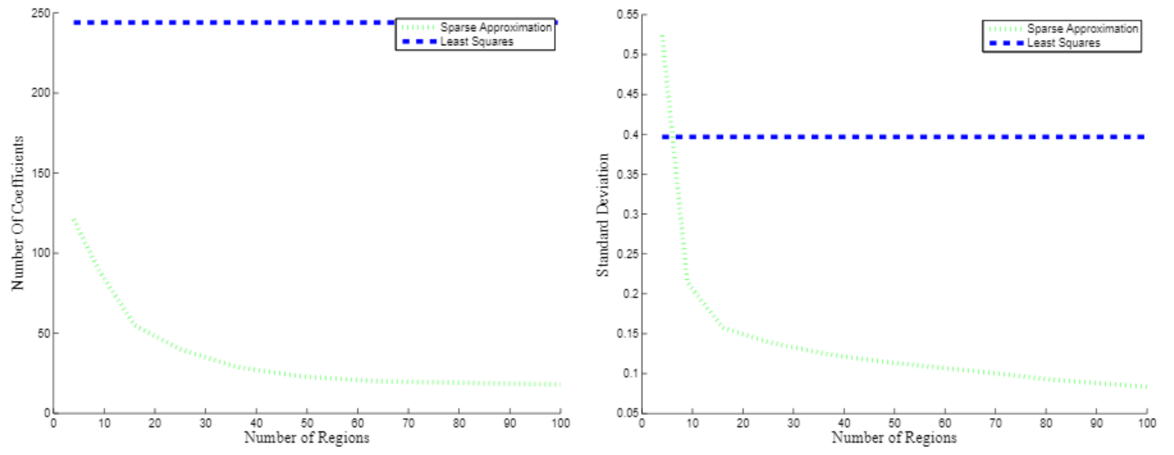


(e) 36 regions



(f) 49 regions

Figure 4.9: With 25 regions and 20 basis functions, a seemingly reasonable approximation is formed.

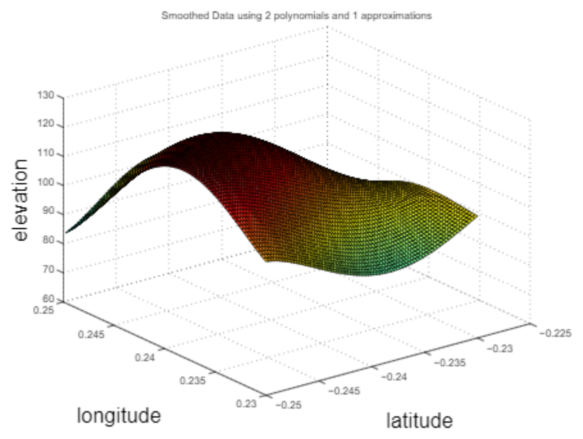


(a) Number of Coefficients vs Number of Regions (b) Standard Deviation of Error vs Number of Regions

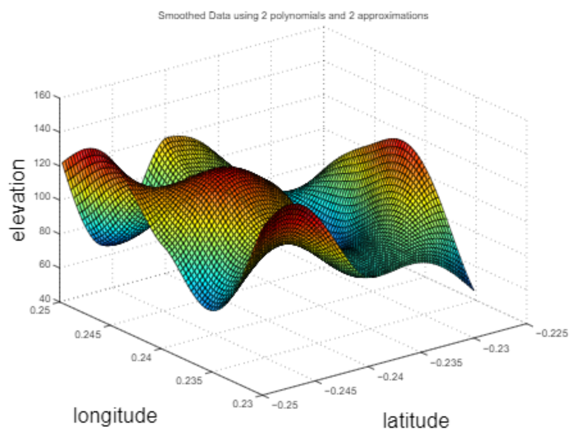
Figure 4.10: Sparsity is enforced successfully through regional approximation, as 20 basis functions accurately form the model.

4.3 Conclusions

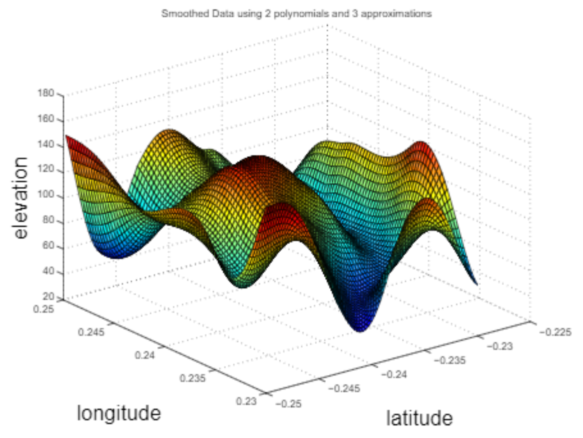
Increasing the number of regional approximations does enforce sparsity, but this is accomplished more easily for noise-free signals, and secondly, for less complicated signals. This principle has been demonstrated for 1-D and 2-D samples, but the algorithm is extensible in n-Dimensions. Due to computational limitation and understanding of the output, higher dimensional examples were not reviewed in this research.



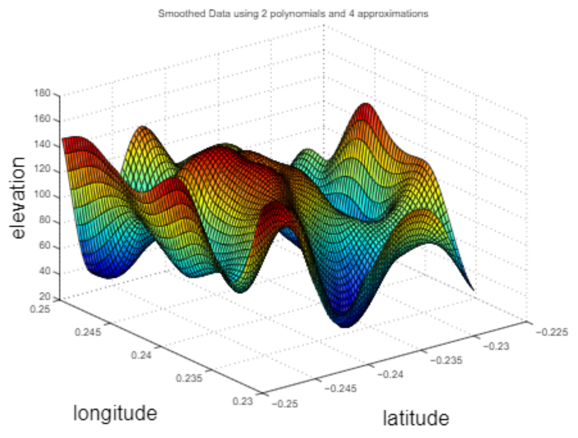
(a) 2nd order Polynomials and 1 region



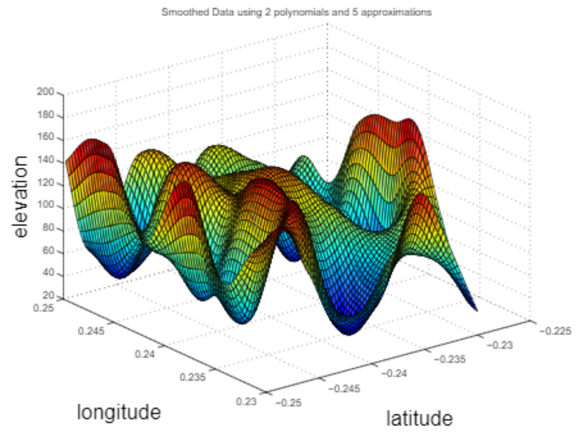
(b) 2nd order Polynomials and 4 regions



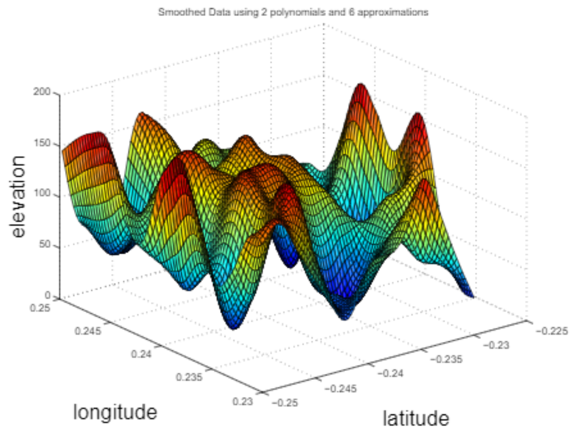
(c) 2nd order Polynomials and 9 regions



(d) 2nd order Polynomials and 16 regions

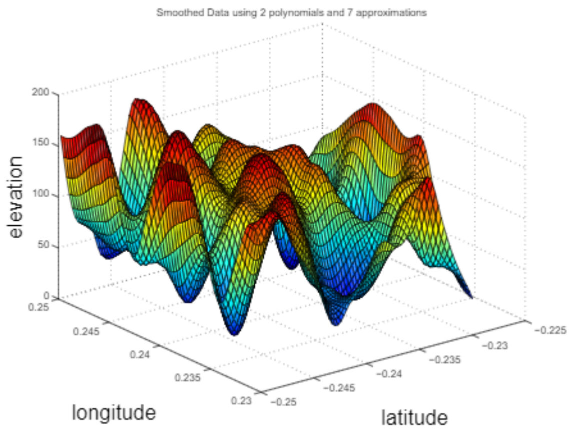


(e) 2nd order Polynomials and 25 regions

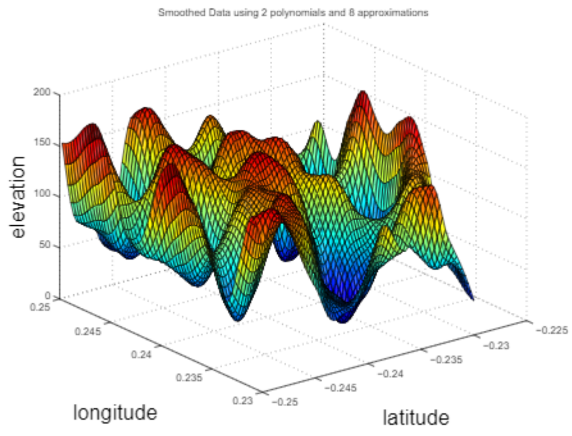


(f) 2nd order Polynomials and 36 regions

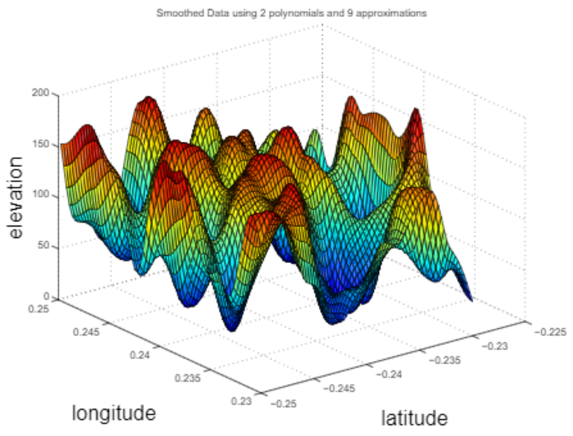
Figure 4.11: When only using 3 basis functions, it is seen that increasing the number of regions better approximates the moon's surface.



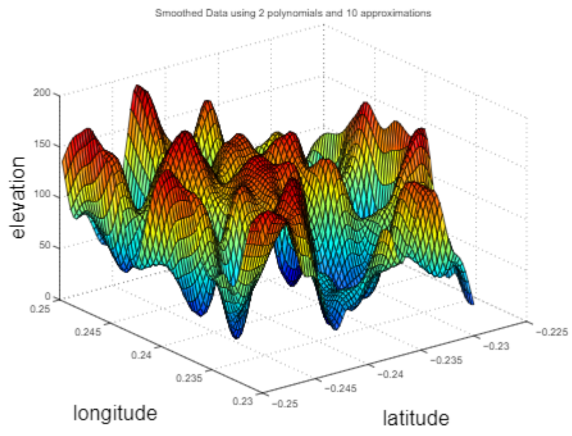
(a) 2nd order Polynomials and 49 regions



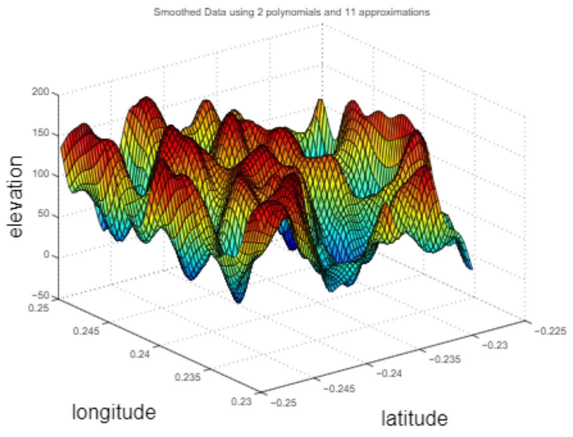
(b) 2nd order Polynomials and 64 regions



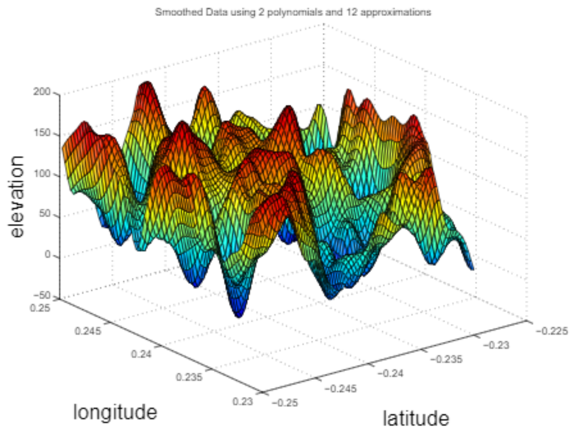
(c) 2nd order Polynomials and 81 regions



(d) 2nd order Polynomials and 100 regions

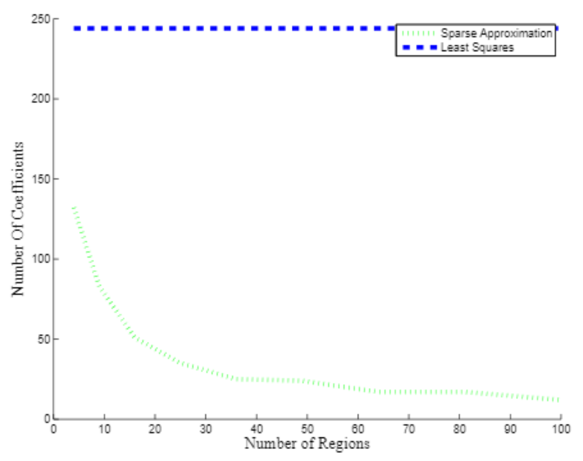


(e) 2nd order Polynomials and 121 regions

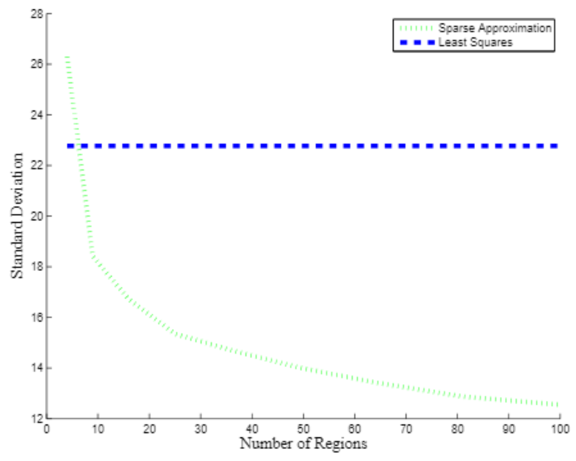


(f) 2nd order Polynomials and 144 regions

Figure 4.12: With the use of 12 regions and 5 basis functions $(1,\ x,\ y,\ x \cdot y,\ x^2 - \frac{1}{3},\ y^2 - \frac{1}{3})$, a very reasonable approximation of the moon's surface is formed.



(a) Number of Coefficients vs Number of Regions



(b) Standard Deviation of Error vs Number of Regions

Figure 4.13: The regional model out-performs the global least-squares easily and utilized drastically fewer coefficients.

Chapter 5

Applications, Review, and Conclusions

5.1 Introduction

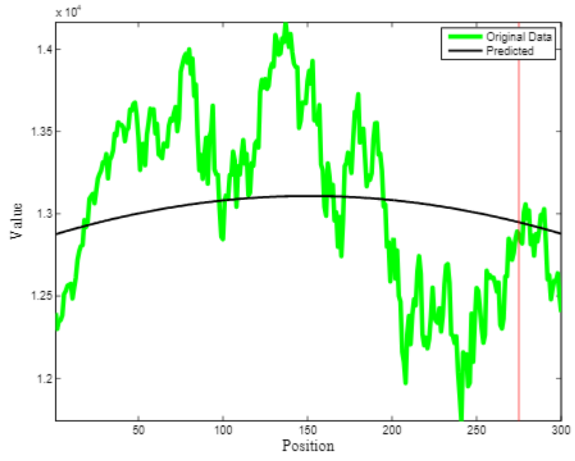
In previous chapters improvements in signal modeling were introduced. Beginning with least squares, enforcing sparsity through optimization routines, and enforcing sparsity by regional approximation, it was shown that an algorithm could select the best basis functions (or at least adequate ones) from an over-complete dictionary. This chapter is a set of final thoughts that recaps on this research, discussing its applications, as well as the direction for future research. As an introduction, least squares approximation showed how a signal could be globally approximated using a minimization routine. The limitations of this approach were the neglect of regional features as well as the dependence on a large set of basis functions to produce a better approximation. Then it was seen that by enforcing sparsity, a small amount of error was allowed in exchange for a drastic decrease in the number of basis functions used. Having this knowledge of the dominant functions, sparsity was further enforced by regional approximations - in which it was seen that in many cases, fewer functions were needed when more regions were globally mapped. In this chapter, a critical look is taken at the accomplishments of the current research, as well as the continued exploration of this exciting branch of study.

5.2 How Much Money have we made?

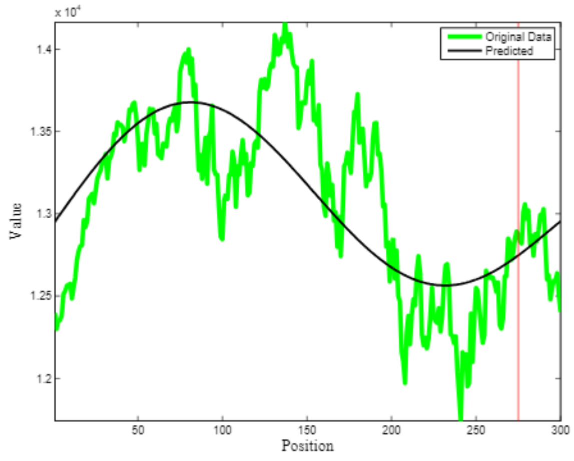
With all the financial modeling, the ability to accurately produce an approximation has added incentive. . . profit. Previous chapters have shown the difficulties of modeling a complicated signal, but this section shows the fruits of that labor. To begin, figures 5.1 and 5.2 show how the model for the 1-year Dow Jones Industrial Average changes by allowing for more basis functions.

As seen in figures 5.1 and 5.2, as more basis functions are allowed to form the model, more features of the signal are captured by the approximation. Using the dominant basis functions that the algorithm selected, it would appear that an adequate prediction is formed. Since money can be made off the transition of the price of a stock or security, predicting how a signal behaves is a huge advantage.

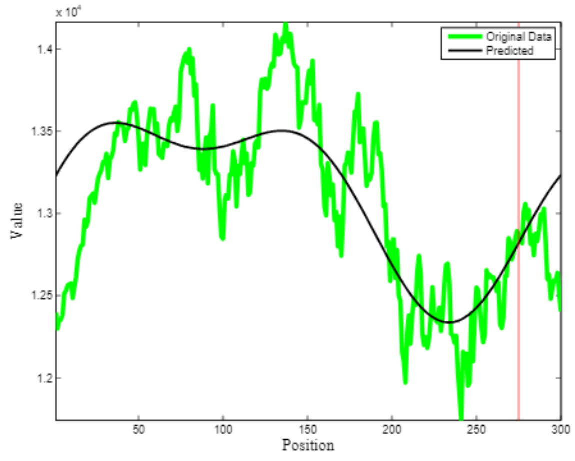
To understand how profit would be made, a technique known as backtracking will determine if the buy and sell indicators can indeed obtain a profit. Using the predictions formed by the model, a buy signal would occur when the derivative in price is positive, and a close signal would occur when the derivative of the price is negative. If the derivative is zero, the next day's derivative is considered. Figure 5.3 shows the results of trading in this trading pattern. As hypothesized, there are advantages to trade using a sparse model.



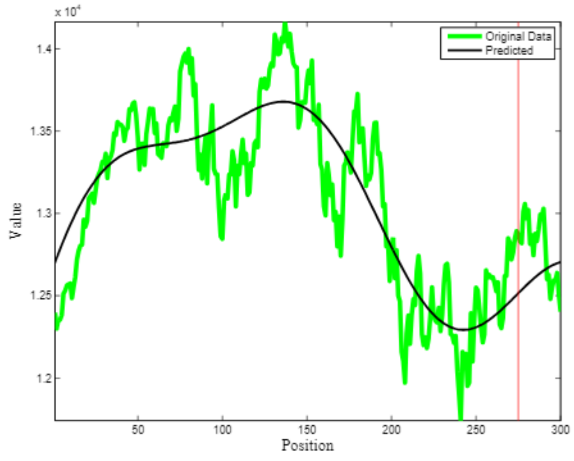
(a) 1 year of training and 1 basis function



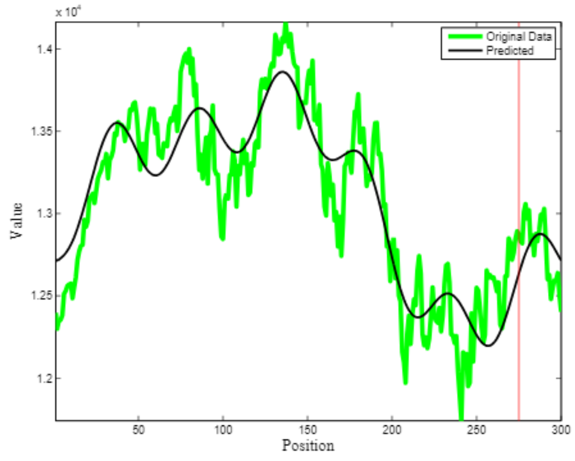
(b) 1 year of training and 2 basis functions



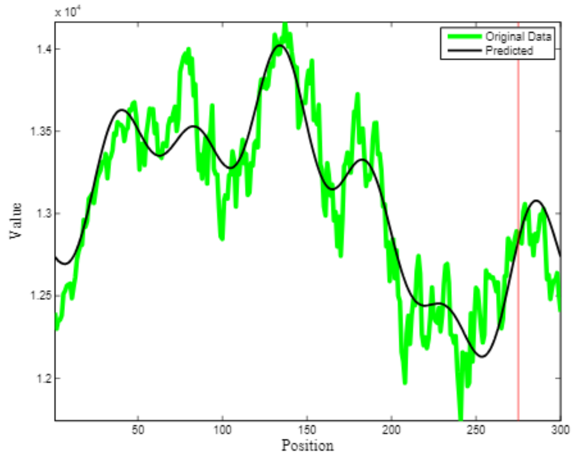
(c) 1 year of training and 3 basis functions



(d) 1 year of training and 4 basis functions

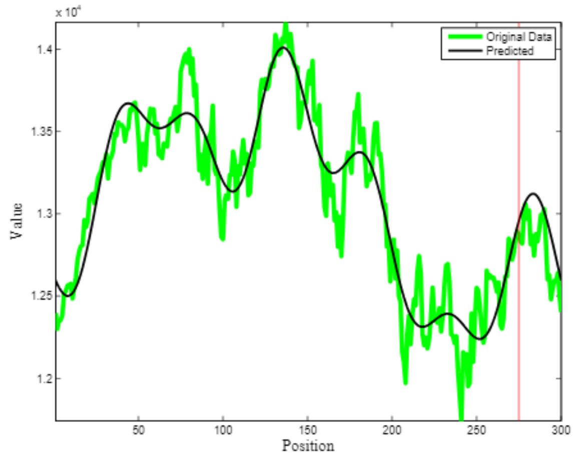


(e) 1 year of training and 5 basis functions

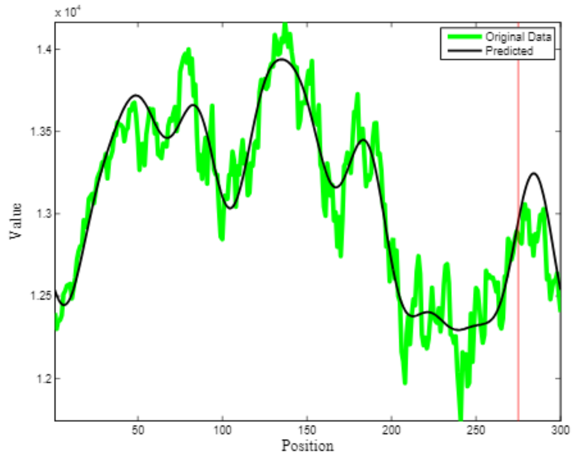


(f) 1 year of training and 6 basis functions

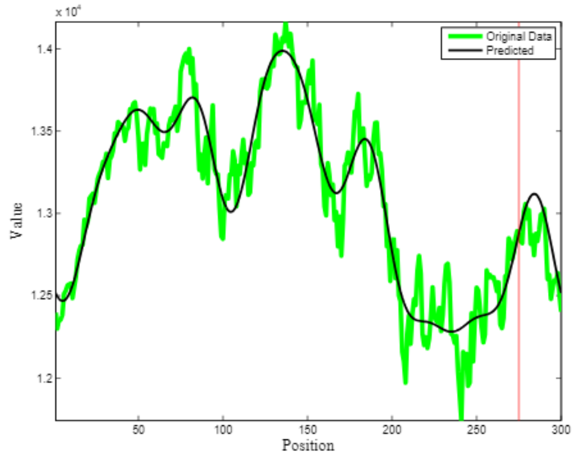
Figure 5.1: The DJI approximation captures more features of the original by increasing the number of basis functions used.



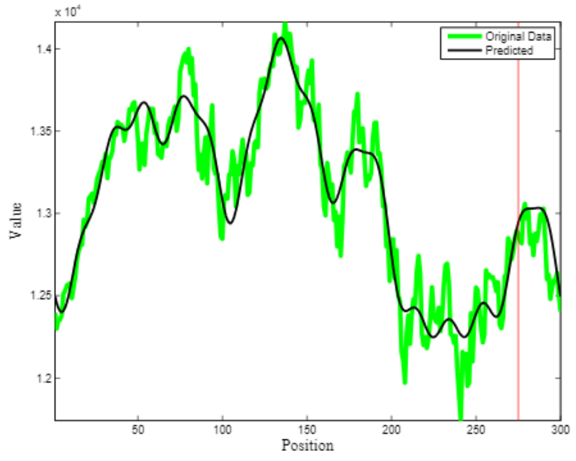
(a) 1 year of training and 7 basis function



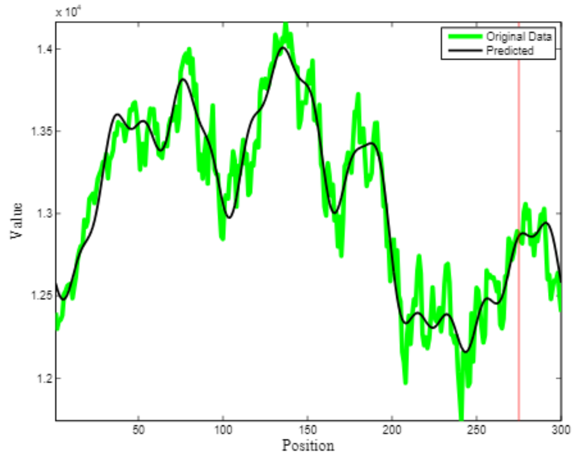
(b) 1 year of training and 8 basis functions



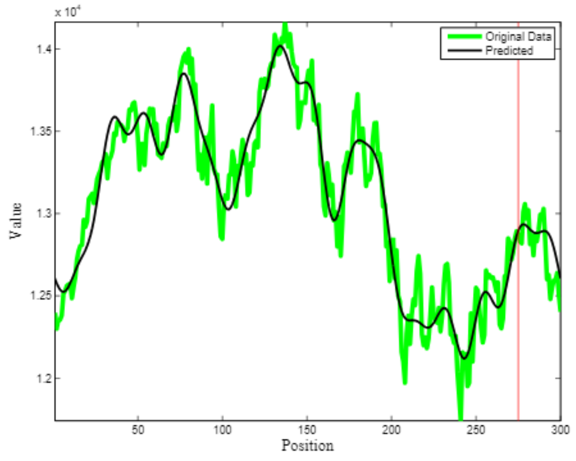
(c) 1 year of training and 9 basis functions



(d) 1 year of training and 10 basis functions

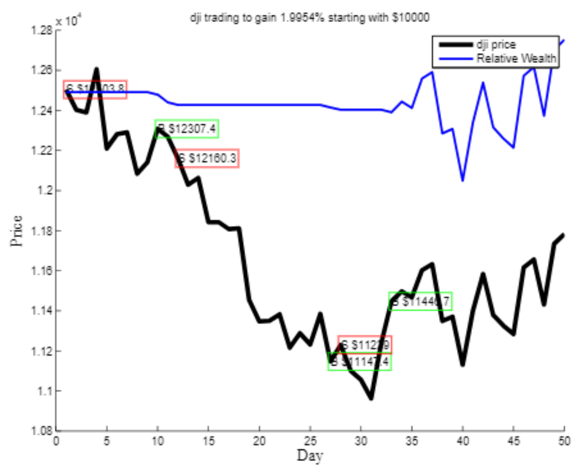


(e) 1 year of training and 11 basis functions

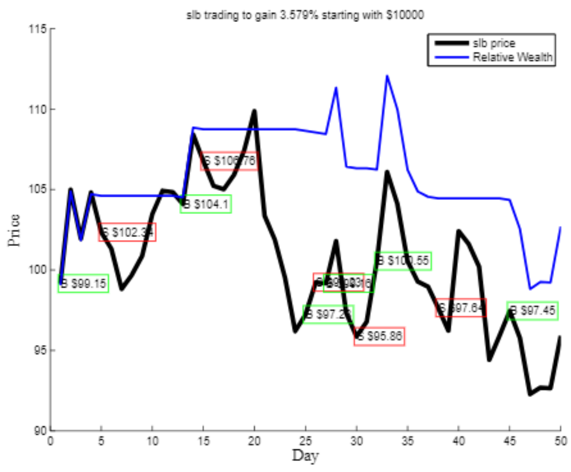


(f) 1 year of training and 12 basis functions

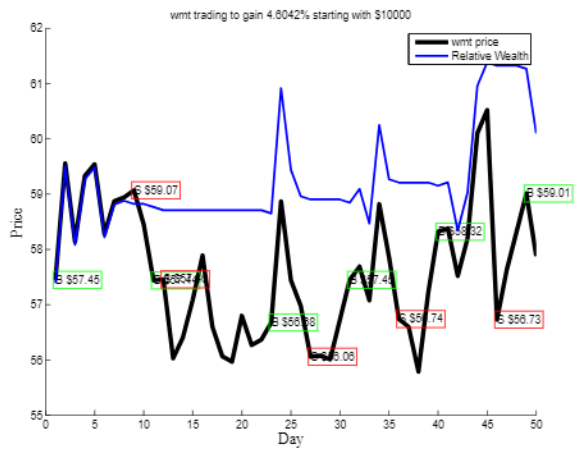
Figure 5.2: After a certain point, the approximation no longer improves through the addition of basis functions.



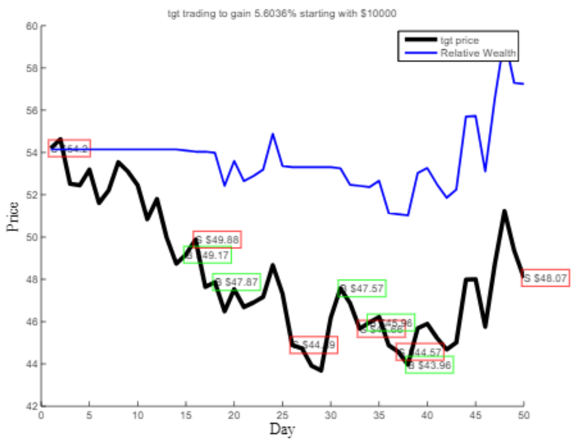
(a) DJI 50-day back tracking



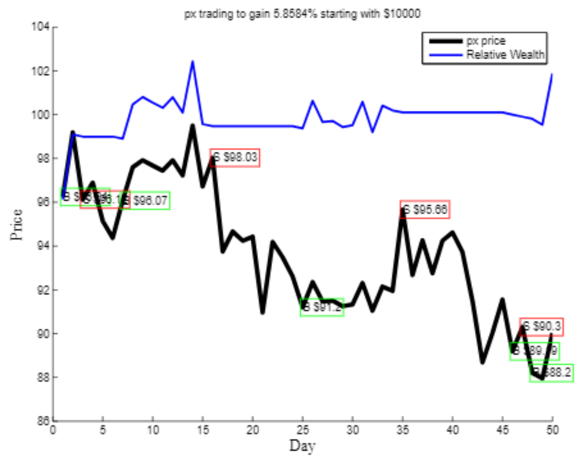
(b) SLB 50-day back tracking



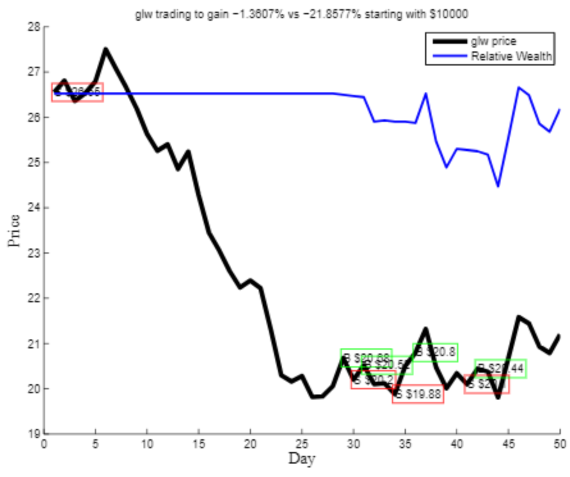
(c) WMT 50-day back tracking



(d) TGT 50-day back tracking

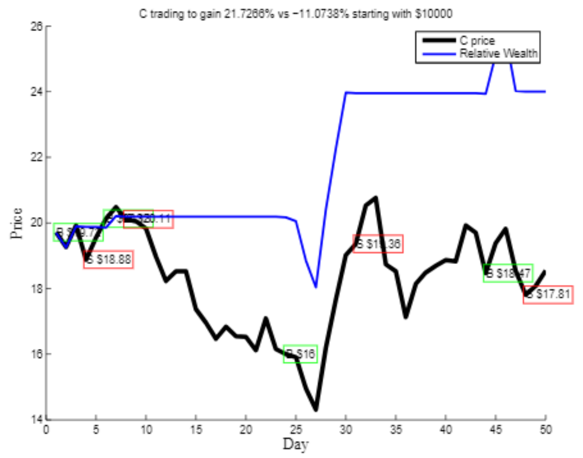


(e) PX 50-day back tracking

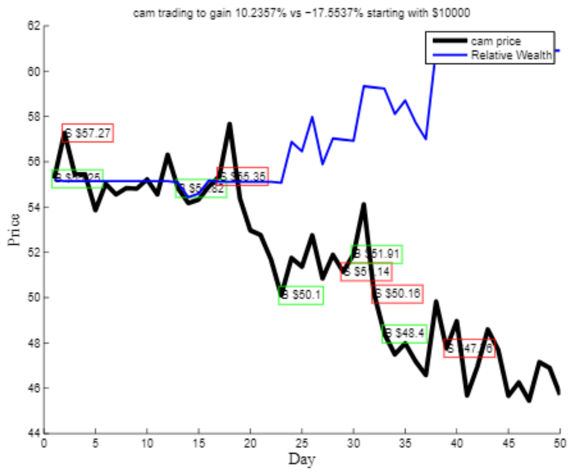


(f) GLW 50-day back tracking

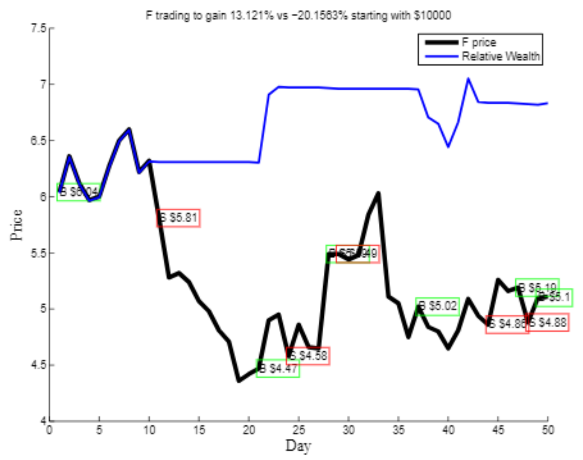
Figure 5.3: In most cases the portfolios gain using the model, but in all cases, the model out-performs the actual stock ticker.



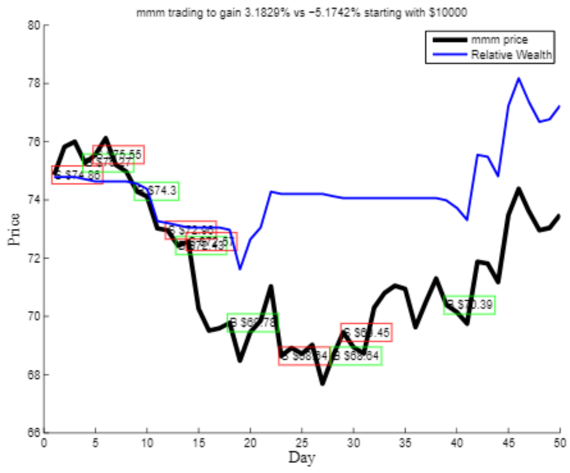
(a) C 50-day back tracking



(b) CAM 50-day back tracking



(c) F 50-day back tracking



(d) TGT 50-day back tracking

Figure 5.4: In these cases, the model does drastically better than the actual ticker.

In figure 5.3, the time-based portfolio fluctuations are shown, but in table 5.2, the end results are tabulated.

ticker	50-day ticker change	50-day change in wealth
DJI	-9.44 %	1.99 %
SLB	-4.21 %	3.57 %
WMT	0.11 %	4.60 %
TGT	-9.62 %	5.60 %
PX	-7.16 %	5.86 %
GLW	-21.86 %	-1.36 %
C	-11.07 %	21.72 %
CAM	-17.55 %	10.23 %
F	-20.15 %	13.12 %
MMM	-5.17 %	3.18 %

As seen from table 5.2, in ALL cases, model-based trading out-performs the actual ticker performance. In only case was a loss experienced, but given the spread between the loss the ticker took and the loss the model-based trading took, there is still a significant advantage to trading with a model.

5.3 Conclusions

In this section, the accomplishments of this research will be considered, as well as future directions this research could take. This research lays the foundation for many projects and developments, as well as provides a benchmark to compare against other model development methods in the field of system identification.

5.3.1 What this research has shown

This research has demonstrated the evolution of signal approximation. First, signals were globally approximated in 1, 2, and n dimensions using Least Squares. Then the Least Squares approach was replaced by an optimization routine which minimized the 1-norm of the weighting coefficients along with the 2-norm of the residual. This led to the advancement of sparsity (minimum number of basis functions used). Then sparsity was shown to be enhanced through regional approximations globally mapped to reconstruct the signal.
To accomplish this, a veritable suite of scripts that can reconstruct a signal using each methods in 1, 2, or n-dimensions was formed. Signals of 1, and 2, dimensions can be plotted - such that the end user has a visual sense of the approximation, while signals of n-dimension can only be observed in terms of error. These scripts were used to analyze five main examples related to the research and personal interest of the author. In section 5.3.2, further development of this script suite and analysis using these tools is discussed.

5.3.2 Further Work

In this section, further examples are suggested that will help demonstrate the capabilities of this research, as well as provide an input output relationship for the system in questions. This section will also discuss comparable methods in system identification, which have know advantages and disadvantages. Finally, further developments of the script suite will be proposed - as well another problem not yet solved by the author.

Examples to consider

There are many real-world systems that could be modeled using input output relationships discussed in this research. Two examples will help further the research of the LAIRs lab and will provide a good sense of the applicability of this research on physical systems.

The two degree of freedom helicopter has two main inputs, main rotor velocity and tail rotor velocity - that provide a 2-dimensional input. It can be argued that this is actually a 6 dimensional problem with initial conditions of pitch and yaw and pitch and yaw rates being the other inputs. The outputs in this model, which are orthogonal, and therefore not considered to be inputs of each other, are the angular velocities of pitch and yaw. These derivatives can later be numerically integrated to know position.

The six degree of freedom helicopter has four rotors that can input a velocity to the system. Position information matters significantly in terms of Hover In Ground Effect (HIGE) or Hover Out of Ground Effect (HOGE) so all position information is considered an input. Also, all velocity information is considered. Thus, a 16 dimensional model will be used to approximate the quad-rotor hovercraft with six degrees of freedom. This problem as 12 outputs as well, and some of these relationships interact due to aerodynamic and dynamic effects. Thus, this problem could potentially be a 27 dimensional problem.

Benchmarks

There are a number of different methods that would provide comparable results to those observed by this research. These models would provide a good benchmark for how well the methods in this research approximate a signal relative to "industry standard" methods.

System ID methods include eigen-realization algorithms using singular value decomposition of the Henkle matrix. This basically develops a discrete time state space model that can help relate inputs to outputs. This allows for every input and output to be related in one state equation. Numerical integration can then allow for the understanding of system behavior.

Neural Networks have risen in popularity - particularly in the stock market examples. Like this research, neural networks weight simple processing elements, which combine to represent global behavior. In relationship to this research, neural networks not only weight basis functions to form a signal approximation, but also approximation the transition of those weights. Each time this secondary type of approximation is formed, it is called a layer. Using this type of approach, very accurate signal approximations can be formed. On the other hand, having many basis functions will lead to computational overload. Thus, it will be tough to form a true comparison, but a relationship between computational time and accuracy tradeoff points can be assessed.

Knowing the actual properties of a physical system can lead to a dynamic model. From this, equations of motions can be developed. The purpose of this research is to approximate the output without knowing the underlying dynamics, so comparing the output of the dynamic model to the sparse approximation, to the response of the physical model, will provide a good sense of the advantages of either approach.

Software Releases

Thus far, the other has only provided a suite for multi-dimensional analysis using least squares, sparsity through optimization, and sparsity through global-local mapping. Further work could lead to any number of the following releases.

A sparsity GUI could help the end-user visually see the transition of the output approximation by varying the cardinality or γ. This would provide a "user optimized" sparse solution through visual aid.

A de-noising package would use the principals of this research to create a smooth output of a measured signal. This will help the end-user better correlate data and allow them to essentially use less expensive equipment for their measurement.

A real time tracking and prediction package could be developed to assist in guidance and navigation, or investment, or a wide array of application. In this case, this is a low dimensional problem, with basic inputs as time or position. If a forecast of data is generated, a controller can respond more rapidly and provide the appropriate inputs for the desired response.

Since complexity of signals was discussed, guidelines for posing the optimization routines should be developed. A table to give basic γ guidelines, or equations for γ should be suggested for future users of the scripts developed by this research.

Adaptive γ

One desire of this research that never saw fruition is an adaptive γ. This would automatically determine the best model based on certain criteria. One criteria could be little improvement in the error subsequent to a decrease in γ. Another could be a metric of cardinality times 2-norm error. These methods would be evaluated and the most successful one would lead to an algorithm that would determine "the best" approximation for any given data.

Appendix A

Why Least Squares?

The least squares problem has an analytical solution - achieving a feasible solution when minimized. Though the general objective is to minimize error, various other strategies have their limitations or simply will not achieve a single feasible solution.

As just described in chapter 2, the Least Squares method attempts to minimize the residual in order to achieve $\hat{\mathbf{x}}$ which achieves the "best fit" approximation. Due to the advantages of matrix math, the Least Squares actually has an analytical solution, which is found by taking the Pseudo-Inverse of the H matrix, which shall be discussed further in A.4.

A.1 Minimizing the Residual sum

To minimize the error, the residual $\mathbf{r} = \sum_{i=1}^{n} \tilde{y}_i - h_i\hat{\mathbf{x}}$ could attempt to be minimized. During a minimization, the derivative is set equal to zero such that $\frac{d\mathbf{r}}{d\hat{\mathbf{x}}} = 0$. However, it is seen that

$$\mathbf{r} = \sum_{i=1}^{n} \tilde{y}_i - h_i\hat{\mathbf{x}}$$
$$\frac{d\mathbf{r}}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} -h_i$$

where i is the current row.

Unfortunately, $\frac{d\mathbf{r}}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} -h_i = 0$ is a useless solution, for it allows for an $\hat{\mathbf{x}}$ that does not exist or is undefined. Also, there is no minimum to $\tilde{\mathbf{y}} - H\hat{\mathbf{x}}$ as $\hat{\mathbf{x}}$ can be infinitely positive to minimize the residual. Therefore it is seen that since error can be negative, minimizing the residual is not a well-posed method to solve for $\hat{\mathbf{x}}$.

A.2 Minimizing the absolute Residual sum

A better option is to minimize the absolute value of the residual.$|\mathbf{r}| = \sum_{i=1}^{n} |\tilde{y}_i - h_i\hat{\mathbf{x}}|$ This has an advantage over the former minimization because the residual sum can't combine positive and negative error that cancel out, but instead any error makes a positive contribution. Thus, the minimization of the absolute residual sum is a better routine for determining X, however it is seen that

$$
\begin{aligned}
|\mathbf{r}| = \sum_{i=1}^{n} |h_i \cdot X - b_i| = \sum_{i=1}^{n} \sqrt{(\tilde{y}_i - h_i\hat{\mathbf{x}})^2} = \sum_{i=1}^{n} \left((\tilde{y}_i - h_i\hat{\mathbf{x}})^2\right)^{\frac{1}{2}} \\
\tfrac{d|\mathbf{r}|}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} \tfrac{1}{2} \left((h_i \cdot \hat{\mathbf{x}} - b_i)^2\right)^{-\frac{1}{2}} \cdot \tfrac{d}{d\hat{\mathbf{x}}} (\tilde{y}_i - h_i\hat{\mathbf{x}})^2 \\
\tfrac{d|\mathbf{r}|}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} \tfrac{1}{2} \left((\tilde{y}_i - h_i\hat{\mathbf{x}})^2\right)^{-\frac{1}{2}} \cdot 2 (\tilde{y}_i - h_i\hat{\mathbf{x}}) \cdot \tfrac{d}{dx} (\tilde{y}_i - h_i\hat{\mathbf{x}}) \\
\tfrac{d|\mathbf{r}|}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} \left((\tilde{y}_i - h_i\hat{\mathbf{x}})^2\right)^{-\frac{1}{2}} \cdot (\tilde{y}_i - h_i\hat{\mathbf{x}}) \cdot h_i \\
\tfrac{d|\mathbf{r}|}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} \tfrac{(\tilde{y}_i - h_i\hat{\mathbf{x}})}{\sqrt{(\tilde{y}_i - h_i\hat{\mathbf{x}})^2}} \cdot h_i \\
\tfrac{d|\mathbf{r}|}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} \tfrac{\tilde{y}_i - h_i\hat{\mathbf{x}}}{|\tilde{y}_i - h_i\hat{\mathbf{x}}|} \cdot h_i
\end{aligned}
$$

such that $|h_i\hat{\mathbf{x}} - \tilde{y}_i| = 0$ leads to a solution that does not exist. For this reason, the minimization of the Residual squared sum.

A.3 Minimization of the Residual squared sum

The Residual squared sum is developed as follows:

$$
J = \frac{1}{2} \sum_{i=1}^{n} |\tilde{y}_i - h_i\hat{\mathbf{x}}|^2 = \frac{1}{2} \sum_{i=1}^{n} \sqrt{(\tilde{y}_i - h_i\hat{\mathbf{x}})^2}^2 = \frac{1}{2} \sum_{i=1}^{n} (\tilde{y}_i - h_i\hat{\mathbf{x}})^2
$$

This does have an analytical minimum, seen by taking the derivative:

$$
\begin{aligned}
J = \tfrac{1}{2} \sum_{i=1}^{n} |\tilde{y}_i - h_i\hat{\mathbf{x}}|^2 = \tfrac{1}{2} \sum_{i=1}^{n} \sqrt{(\tilde{y}_i - h_i\hat{\mathbf{x}})^2}^2 = \tfrac{1}{2} \sum_{i=1}^{n} (\tilde{y}_i - h_i\hat{\mathbf{x}})^2 \\
\tfrac{dJ}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} (\tilde{y}_i - h_i\hat{\mathbf{x}}) \cdot \tfrac{d}{d\hat{\mathbf{x}}} (\tilde{y}_i - h_i\hat{\mathbf{x}}) \\
\tfrac{dJ}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} (\tilde{y}_i - h_i\hat{\mathbf{x}}) \cdot h_i \\
\tfrac{dJ}{d\hat{\mathbf{x}}} = \sum_{i=1}^{n} h_i \cdot (\tilde{y}_i - h_i\hat{\mathbf{x}}) \\
\tfrac{dJ}{d\hat{\mathbf{x}}} = H^T (\tilde{\mathbf{y}} - H\hat{\mathbf{x}}) = 0
\end{aligned}
$$

To set the derivative equal to zero, it is seen that $H\hat{\mathbf{x}} = \tilde{\mathbf{y}}$. The solution ($\hat{\mathbf{x}} = pinv(H) \cdot \tilde{\mathbf{y}}$) is solved using matrix math in section A.4, but later (in chapter 3) optimization routines will be discussed in achieving this solution through minimization. This is also seen as the least squares solution, because it solves for the minimum of the residual squared sum.

A.4 The Pseudo-Inverse

The Pseudo-Inverse [21], also known as the generalized inverse, has the advantage of being able to find an inverse for a non-square matrix and achieve an analytical solution to the Least Squares problem. In general, to set zero the residual $r = \tilde{\mathbf{y}} - H\hat{\mathbf{x}}$, it is seen that $H\hat{\mathbf{x}} = \tilde{\mathbf{y}}$. This is explained in section A. The Pseudo Inverse accomplishes a best fit solution as $\hat{\mathbf{x}} = pinv(H)\tilde{\mathbf{y}}$. Given the equation

$$H_{n\times m}\hat{\mathbf{x}}_{m\times 1} = \tilde{\mathbf{y}}_{n\times 1}$$

where $n > m$, it is seen that a simple matrix inverse cannot be taken due to the fact that H is not square. To achieve an analytical solution using matrix algebra, begin by multiplying both sides by $H^T_{m\times n}$ such that

$$H^T_{m\times n}H_{n\times m}\hat{\mathbf{x}}_{m\times 1} = H^T_{m\times n}\tilde{\mathbf{y}}_{n\times 1}$$

Now that there is a square matrix $(H^T_{m\times n}H_{n\times m})_{m\times m})$, an inverse can be taken such that

$$(H^T_{m\times n}H_{n\times m})^{-1}(H^T_{m\times n}H_{n\times m})\hat{\mathbf{x}}_{m\times 1} = (H^T_{m\times n}H_{n\times m})^{-1}H^T_{m\times n}\tilde{\mathbf{y}}_{n\times 1}$$

It is now seen that this poses a solution as the following identities are seen

$$\underbrace{(H^T_{m\times n}H_{n\times m})^{-1}H^T_{m\times n}H_{n\times m}}_{I=1}\hat{\mathbf{x}}_{m\times 1} = \underbrace{(H^T_{m\times n}H_{n\times m})^{-1}H^T_{m\times n}}_{pinv(H)}\tilde{\mathbf{y}}_{n\times 1}$$

$$\hat{\mathbf{x}}_{m\times 1} = \underbrace{(H^T_{m\times n}H_{n\times m})^{-1}H^T_{m\times n}}_{pinv(H)}\tilde{\mathbf{y}}_{n\times 1}$$

$pinv()$ is the pseudo inverse.

If there are more columns than rows, $pinv\left(H\right)_{m\times n} = H^T_{m\times n}\left(H_{n\times m}H^T_{m\times n}\right)^{-1}_{n\times n}$.

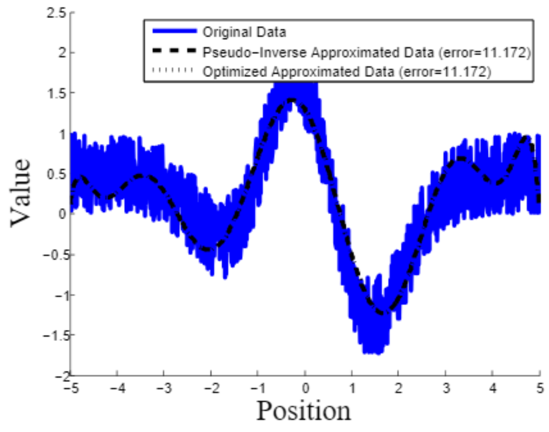
If there are more rows than columns, $pinv\left(H\right)_{m\times n} = \left(H^T_{m\times n}H_{n\times m}\right)^{-1}_{m\times m}H^T_{m\times n}$.

This is commonly called the least squares solution.

A.5 2-norm minimization vs Analytical Least Squares (pseudo-inverse)

The pseudo-inverse is viewed as the analytical solution to the least-squares minimization. Using this 1-D example, it will be shown that this is an accurate assumption. The details of each problem is briefly explained in table A.1. The plots of the least squared minimization solution and the pseudo-inverse approximation actually overlap - as seen in figure A.1, as well as have the same error.

Table A.1 actually shows the same coefficient results and the same error results through posing the Least Squares method as either a minimization or the analytical solution.



(a) Least Squares Minimization and Pseudo Inverse Approximation

Figure A.1: The Least Squares Minimization and the Pseudo Inverse Approximation provide the same result, shown overlapping in this figure and yielding the same error.

Table A.1: A comparison of 2-Norm Minimization to the Pseudo-Inverse Solution

	2-Norm Minimization	Analytical Least Squares (pseudo-inverse)
Construction	variable $\hat{\mathbf{x}}(m)$ $\min \|\tilde{\mathbf{y}} - H \cdot \hat{\mathbf{x}}\|_2$	$\hat{\mathbf{x}}_{m\times 1} = pinv\left(H_{n\times m}\right)_{m\times n} \cdot \tilde{\mathbf{y}}_{n\times 1}$
Coefficients of $\hat{\mathbf{x}}$	$\begin{pmatrix} 0.2237 \\ 0.3905 \\ 1.1316 \\ 2.1766 \\ -4.7654 \\ -18.0085 \\ 14.1469 \\ 68.0966 \\ -32.2813 \\ -202.8808 \end{pmatrix}$	$\begin{pmatrix} 0.2237 \\ 0.3905 \\ 1.1316 \\ 2.1766 \\ -4.7654 \\ -18.0085 \\ 14.1469 \\ 68.0966 \\ -32.2813 \\ -202.8808 \end{pmatrix}$
2-norm error	11.2717	11.2717

Appendix B

Orthogonal Basis Functions

B.1 Introduction to Gram-Schmidt

Gram-Schmidt orthogonalization takes a nonorthogonal set of linearly independent functions and constructs an orthogonal basis over an arbitrary interval. In other words, given the basis function $w_1, w_2, w_3, \ldots, w_n$, construct orthogonal basis function such that $\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_n$ such that $span\left\{w_i\right\} = span\left\{\Phi_i\right\}$.

B.2 The Gram-Schmidt Process

B.2.1 Selection of the Basis Functions

The original basis functions, which shall be orthogonalized, are increasing in order, such that a polynomial basis is developed.

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{n-1} \end{pmatrix}$$

B.2.2 Orthogonalization

First define the inner product as $\langle w, \Phi \rangle = \int\limits_{-1}^{1} w(x)\,\Phi(x)\,dx$

The Gram-Schmidt process takes the general form of $\Phi_n = w_n - \sum\limits_{i=1}^{n-1} \frac{\langle w_n, \Phi_i \rangle}{\langle \Phi_i, \Phi_i \rangle} \Phi_i$

Now begin the Gram-Schmidt process using the basis functions provided above.

$$\Phi_1 = w_1 = 1$$

$$\Phi_2 = w_2 - \frac{\langle w_2, \Phi_1 \rangle}{\langle \Phi_1, \Phi_1 \rangle}\Phi_1 = w_2 - \frac{\int\limits_{-1}^{1} w_2(x)\,\Phi_1(x)\,dx}{\int\limits_{-1}^{1} \Phi_1(x)\,\Phi_1(x)\,dx}\Phi_1 = x - \frac{\int\limits_{-1}^{1} x \cdot 1 dx}{\int\limits_{-1}^{1} 1 \cdot 1 dx} 1 = x - \frac{\frac{1}{2}x^2\big|_{-1}^{1}}{x\big|_{-1}^{1}} 1 = x - \frac{0}{2} = x$$

$$\Phi_3 = w_3 - \frac{\langle w_3, \Phi_1 \rangle}{\langle \Phi_1, \Phi_1 \rangle}\Phi_1 - \frac{\langle w_3, \Phi_2 \rangle}{\langle \Phi_2, \Phi_2 \rangle}\Phi_2 = w_3 - \frac{\int\limits_{-1}^{1} w_3(x)\,\Phi_1(x)\,dx}{\int\limits_{-1}^{1} \Phi_1(x)\,\Phi_1(x)\,dx}\Phi_1 - \frac{\int\limits_{-1}^{1} w_3(x)\,\Phi_2(x)\,dx}{\int\limits_{-1}^{1} \Phi_2(x)\,\Phi_2(x)\,dx}\Phi_2$$

$$\Phi_3 = x^2 - \frac{\int\limits_{-1}^{1} x^2 \cdot dx}{\int\limits_{-1}^{1} 1 \cdot 1 \cdot dx} 1 - \frac{\int\limits_{-1}^{1} x^2 \cdot x \cdot dx}{\int\limits_{-1}^{1} x \cdot x \cdot dx} x = x^2 - \frac{\frac{2}{3}}{2} - \frac{0}{\frac{2}{3}} x = x^2 - \frac{1}{3}$$

The results of this process are expressed in section B.3.

B.3 Resulting Orthogonal Basis Functions

Using the Gram-Schmidt process on the basis functions above, the following orthogonal basis functions were formed:

$$\begin{pmatrix} 1 \\ x \\ x^2 - \frac{1}{3} \\ x^3 - \frac{3}{5}x \\ x^4 - \frac{6}{7}x^2 + \frac{3}{35} \\ x^5 - \frac{10}{9}x^3 + \frac{5}{21}x \\ x^6 - \frac{15}{11}x^4 + \frac{5}{11}x^2 - \frac{5}{231} \\ x^7 - \frac{21}{13}x^5 + \frac{105}{143}x^3 - \frac{35}{429}x \\ x^8 - \frac{28}{15}x^6 + \frac{14}{13}x^4 - \frac{28}{143}x^2 + \frac{7}{1287} \\ x^9 - \frac{36}{17}x^7 + \frac{126}{85}x^5 - \frac{84}{221}x^3 + \frac{63}{2431}x \\ x^{10} - \frac{45}{19}x^8 + \frac{630}{323}x^6 - \frac{210}{323}x^4 + \frac{315}{4199}x^2 - \frac{63}{46189} \end{pmatrix}$$

Appendix C

Multi-Dimensional Relationship Development

C.1 Introduction

The goal is to take a multi-dimensional input signal, expand it using basis functions, and establish various relationship between those basis functions depending on the order of the relationship. A simple 2-D example can be seen as follows:

Order	Relationship Formed			
0		1		
1		x	y	
2	x^2		$x \cdot y$	y^2
3	x^3	$x^2 \cdot y$	$x \cdot y^2$	y^3
higher		$\vdots$		

This gets far more complicated, by even adding a single dimension for a 3-D example.

Order	Relationship Formed								
0				1					
1		x		y		z			
2		x^2	$x \cdot y$	y^2	$y \cdot z$	z^2		$z \cdot x$	
3	x^3	$x^2 \cdot y$	$x \cdot y^2$	y^3	$y^2 \cdot z$	$y \cdot z^2$	z^3	$z^2 \cdot x$	$z \cdot x^2$
higher					$\vdots$				

This suddenly becomes more complicated when each polynomial is orthogonal. Thus, x^2 is really $x^2 - \frac{1}{3}$ and x^3 is really $x^3 - \frac{3}{5}x$ and so on.

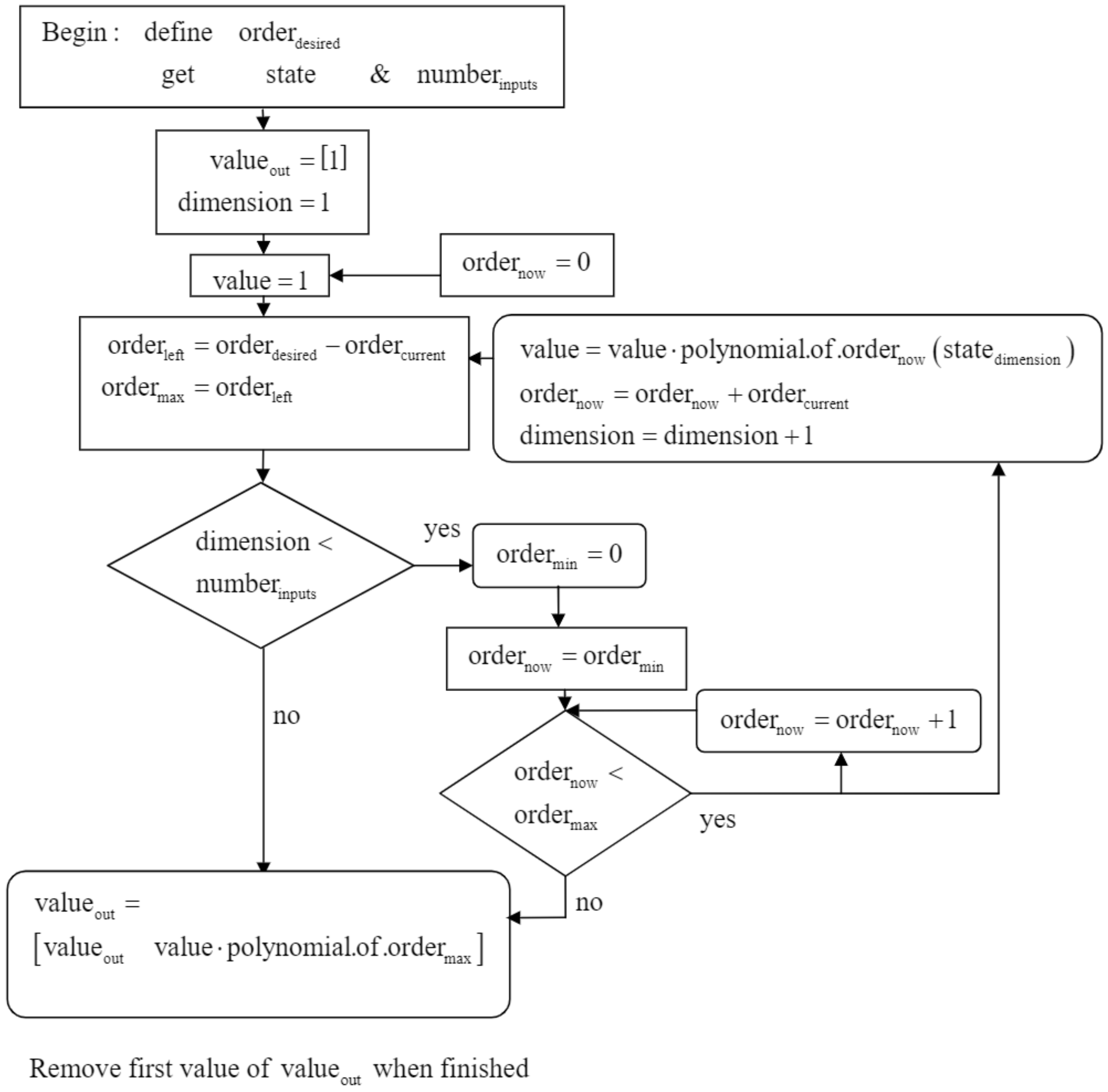
C.2 Approach

To develop every possible combination of a given order for a state in a set of data, using multiple inputs, the desired order must first be known. Beginning with the first state, every possible combination of the subsequent states' orders is found while holding the previous state at a given value which ranges from order 0 to the desired order. The model for this algorithm is shown in figure C.1.

This algorithm was implemented and tested using MATLAB [22], and the results are shown below.

C.3 Results

The following are the results of testing $x = 1$, $y = 2$, $z = 3$ as in the table above.


Begin : define order$_{desired}$
get state & number$_{inputs}$
value$_{out}$ = [1]
dimension = 1
value = 1
order$_{now}$ = 0
order$_{left}$ = order$_{desired}$ − order$_{current}$
order$_{max}$ = order$_{left}$
value = value · polynomial.of.order$_{now}$ (state$_{dimension}$)
order$_{now}$ = order$_{now}$ + order$_{current}$
dimension = dimension + 1
dimension < number$_{inputs}$
yes
order$_{min}$ = 0
order$_{now}$ = order$_{min}$
order$_{now}$ = order$_{now}$ + 1
order$_{now}$ < order$_{max}$
yes
no
no
value$_{out}$ = [value$_{out}$ value · polynomial.of.order$_{max}$]
Remove first value of value$_{out}$ when finished

(a) Implementation of the Every-Possible-Combination Algorithm

Figure C.1: This flow chart describes the development of every combination of a state of multiple dimension for a single order polynomial to be developed.

Order	Relationship Formed									
0				1						
1		1		2		3				
2	0.6667		2.0000	3.6667	3.0000	6.0000		8.6667		
3	0.4000	1.3333	3.6667	6.8000	2.0000	6.0000	11.0000	8.6667	17.3333	25.2000
higher						⋮				

Appendix D

n-Dimensional Smoothing Functions

D.1 Weighting Functions Criteria

An arbitrary set of vertices $\{X_1, X_2, X_3, \ldots X_n\}$ are introduced at a uniform distance h apart. These vertices represent the center of a weighting function, and for this purpose, the center of a region. Now assume a normalized parameter such that $-1 \leq x_I \leq 1$:

$$x_I \triangleq \frac{(X - X_I)}{h}$$

Assume that $\bar{F}_I(x)$ is the smooth approximation of a data set at a point. This is the smoothed approximation blends the functions $F_{I-1}(X)$, $F_I(X)$, and $F_{I+1}(X)$ using a local weighting function $w(x_I)$, that can be introduced as follows:

$$\bar{F}_I(X) = w(x_{I-1}) F_{I-1}(X) + w(x_I) F_I(X) + w(x_{I+1}) F_{I+1}(X)$$

For the smoothing to be un-biased, the derivatives have to match also, such that

$$\frac{d\bar{F}_I(X)}{dx} = w\left(x_{I-1}\right)\frac{dF_{I-1}(X)}{dx} + w\left(x_I\right)\frac{dF_I(X)}{dx} + w\left(x_{I+1}\right)\frac{dF_{I+1}(X)}{dx}$$
$$\frac{dw(x_{I-1})}{dx}F_{I-1}\left(X\right) + \frac{dw(x_I)}{dx}F_I\left(X\right) + \frac{dw(x_{I+1})}{dx}F_{I+1}\left(X\right)$$

This leads to the important property that the function tends to 1 in the center, $w\left(0\right) = 1$, and the function tends toward zero at distance h and beyond. Since h is normalized, $w\left(1\right) = 0$. Further, the derivative must be zero at the center, such that $\frac{dw(0)}{dx} = 0$, and the derivative must be zero at distance h, which again is normalized, such that $\frac{dw(1)}{dx} = 0$. Therefore, the sum of the weighting functions at any point is 1, such that $w\left(x_I\right) + w\left(x_I + 1\right) = 1 = w\left(x_I\right) + w\left(x_I - 1\right)$.

D.2 Weighting Function Design

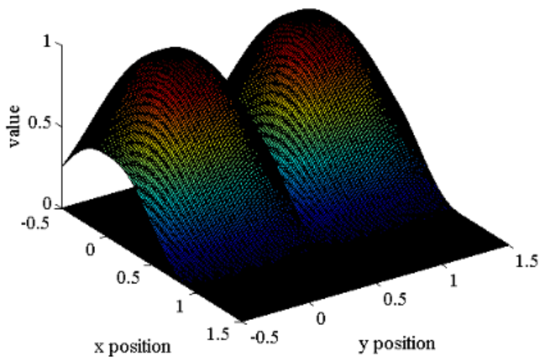
A collection of weighting functions meets the aforementioned criteria of the form $w\left(x\right) = 1 - J\left(x\right)$ with $\frac{dJ(x)}{dx}$ a polynomial of the form $C \cdot x^n \left(1 - x\right)^n$ such that it satisfies the conditions $\frac{d^n w(0)}{d^n x} = 0$ and $\frac{d^n w(1)}{d^n x} = 0$.

With these conditions met, $J\left(1\right) = C \cdot \int\limits_0^1 x^n \left(1 - x\right)^n \cdot dx = 1$ meaning that $C = \left[\int\limits_0^1 x^n \left(1 - x\right)^n \cdot dx\right]^{-1}$ with $n > 0$.

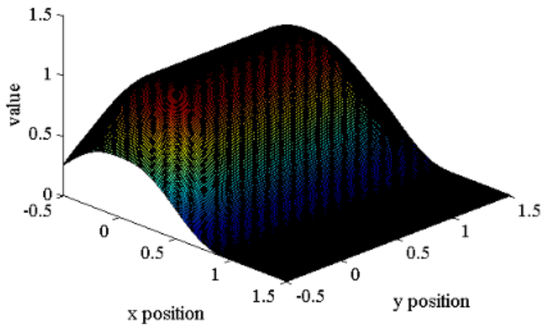
D.3 Resulting Weighting Functions

Using the above procedure, the following weighting functions were developed:

$$\begin{pmatrix} 1 - x \\ 1 + 2 \cdot x^3 - 3 \cdot x^2 \\ 1 - 6 \cdot x^5 + 15 \cdot x^4 - 10 \cdot x^3 \\ 1 + 20 \cdot x^7 - 70 \cdot x^6 + 84 \cdot x^5 - 35 \cdot x^4 \\ 1 - 70 \cdot x^9 + 315 \cdot x^8 - 540 \cdot x^7 + 420 \cdot x^6 - 126 \cdot x^5 \\ 1 + 252 \cdot x^{11} - 1386 \cdot x^{10} + 3080 \cdot x^9 - 3465 \cdot x^8 + 1980 \cdot x^7 - 462 \cdot x^6 \\ 1 - 924 \cdot x^{13} + 6006 \cdot x^{12} - 16380 \cdot x^{11} + 24024 \cdot x^{10} - 20020 \cdot x^9 + 9009 \cdot x^8 - 1716 \cdot x^7 \\ \vdots \end{pmatrix}$$



(a) Two weighting functions shown in 2-D space



(b) The sum of two 2-D weighting functions

Figure D.1: A weighting function evaluates to 1 at its center, 0, and evaluates to 0 at its extent, 1. For higher order weighting functions, the derivative is zero at both the center and the extents.

Bibliography

[1] Joel A. Tropp, Anna C. Gilbert, and Martin J. Strauss. Algorithms for simultaneous sparse approximation: part i: Greedy pursuit. *Signal Process.*, 86(3):572–588, 2006.

[2] Lorenzo Granai and Pierre Vandergheynst. Sparse approximation by linear programming using an l1 data-fidelity term. Technical Report 1403, Signal Processing Institute, Switzerland, 2005.

[3] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems.* Classics in applied mathematics, Englewood Cliffs, New Jersey, 1995.

[4] Joel A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 53:1030, 2006.

[5] Puneet Singla and John L. Junkins. *Multi-Resolution Methods for Modeling and Control of Dynamical Systems.* Texas A & M University, College Station, Texas, 2008.

[6] John L. Crassidis and John L. Junkins.

[7] Eric W. Weisstein. Least squares fitting. From MathWorld–A Wolfram Web Resource., 1999.

[8] Mansfield Merriman. On the history of the method of least squares. *The Analyst*, 4(2):33–36, 1877.

[9] Carl Friedrich Gauss and G. W. Stewart (translator). *Theory of the Combination of Observations Least Subject to Errors: Part One, Part Two, Supplement (Classics in Applied Mathematics)*. Society for Industrial Mathematics; Facsimile edition, Philadelphia, PA, January 1, 1987.

[10] David Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998.

[11] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. CASTLEOFMUSIC, New York, 2002.

[12] Michael Grant. *cvx Users Guide*. Stanford University, Stanford, CA, 2007.

[13] Federico Girosi. An equivalence between sparse approximation and support vector machines. *Neural computation*, 10:1455, 1998.

[14] Howard Karloff. *Linear Programming (Progress in Theoretical Computer Science)*. Edward Brothers Inc., Birkhauser, Boston, 1991.

[15] Jinchao Xu and Jun Zou. Some nonoverlapping domain decomposition methods. *SIAM Review*, 40(4):857–914, 1998.

[16] J. Doyne Farmer and John J. Sidorowich. Predicting chaotic time series. *Phys. Rev. Lett.*, 59(8):845–848, Aug 1987.

[17] Richard Franke. Scattered data interpolation: Tests of some method. *Mathematics of Computation*, 38(157):181–200, 1982.

[18] Joshua B. Tenenbaum. *Global Versus Local Methods in Nonlinear Dimensionality Reduction.* Department of Brain and Cognitive Sciences, Massachusetts Institute of Technolog, Cambridge. MA 02139, 2002.

[19] Andrew D. Back Steve Lawrence, Ah Chung Tsoi. Function approximation with neural networks and local methods: Bias, variance and smoothness. *Australian Conference on Neural Networks*, 96(1):16–21, 1997.

[20] Lu Han and Larry L. Schumaker. Fitting monotone surfaces to scattered data using c1 piecewise cubics. *SIAM Journal on Numerical Analysis*, 34(2):569–585, 1997.

[21] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51:406–413, 1955.

[22] The Mathworks. Matlab and simulink for technical computing., 2007.