HSB
Hochschule Bremen
City University of Applied Sciences

Faculty of Electrical Engineering and Computer Science

IMSAS
Institut für Mikrosensoren, -aktoren und -systeme

Universität Bremen

MASTER THESIS

SMOOTHING AND MODELLING OF SENSOR DATA NETWORK

By

Subramanyam Duvvuri
Matriculation Nr : 367238

SUPERVISED BY

Prof.Dr.Ing.Dieter Kraus
Dr.Ing.Reiner Jedermann

Declaration

I, hereby declare that this thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort has been made to cite and acknowledge this clearly, with due reference to the literature.

Subramanyam Duvvuri
February 2016

Acknowledgements

Firstly,I wish to express deep appreciation to my mentor at IMSAS Dr.Ing.Reiner Jedermann for providing me with this opportunity and all the necessary facilities needed for my research. His continuous guidance and insightful comments, immense knowledge helped me in my research and writing of this thesis.

I would like to express my sincere gratitude to my guide in the university, Prof.Dr.Ing.Dieter Kraus for the continuous support throughout my Master Thesis. His guidance comments and ideas helped me at the time of research and analysis.

None of this would have been possible without the support and strength of my family. I express my heart-felt gratitude to my family and friends for their love and patience.

This thesis is related to the project ‘In-network data analysis of spatially distributed quantities’, founded by the German Research Foundation (DFG). Further information can be found at www.intelligentcontainer.com.

Abstract

The Intelligent Container is used for monitoring food shelf life by monitoring the parameters during transportation. In order to provide an accurate prediction of shelf life of fruits and meat, it is necessary to measure temperature inside food pallets. In order to achieve that, several sensors are mounted on the boxes to capture local temperature deviations inside a truck or container. The sensor node batteries have to survive for several months or even years. Power consumption has to be reduced as much as possible, which can be achieved by limiting the wirelessly transmitted data. The primary purpose of this study is to find algorithms to reconstruct and smooth the data and evaluate prediction error by Root Mean Square deviation. Algorithms for smoothing and modelling should be tested on data related to cooling of fruit containers during ocean transportation. The long term goal is to do in-network processing and send smoothed data with very less parameters. Algorithms for smoothing and modelling should be tested on data related to cooling of fruit containers during ocean transportation.

This report firstly deals with background of the project and the motivation of the approch. In the third chapter the mathematics behind the B-Splines are discussed throughly in one dimension and two dimensions followed by regression using the splines in the fourth chapter. Fifth chapter deals with the smoothing splines and the automated smoothing spline techniques, and their comparisons. Lastly, in the sixth chapter the algorithms are tested on the two data sets. Time to time *incode* section has been added to discuss about the functions and the algorithm used in the code to obtain a particular result. A list of functions has been added at the end of the book.

Contents

1 Background 7

2 B-Splines and its surfaces 9
2.1 Introduction 9
2.2 Bézier curves 10
2.3 B-Splines 10
2.4 B-Spline Surfaces 13

3 Estimation through Regression 16
3.1 Introduction 16

4 Smoothing with Smoothing Spline 22
4.1 Introduction 22
4.2 Theory and Generalization 22
4.3 Implementation 23
4.4 Parameter selection Methods 26
4.5 Timings Tables 27
4.6 New Search Algorithm 28

5 Simulation and Results 29
5.1 Data Set 1 29
5.2 Data Set 2 33

6 Summary and Conclusion 35

List of Figures

1.1 Gas chromatography column and other system components . . . 8
1.2 Java enabled Preon32 sensor node and Preon Cube with CO2 Sensor 8

2.1 B-Spline with Maxima at 2 11
2.2 Open uniform B-Spline - 1D 13
2.3 Calculation through tensor product by keeping one of the directions constant while varing the other 14
2.4 Open Uniform B-Spline 2D 15

3.1 Regression Spline using uniform B-Splines 17
3.2 Regression Spline using Open uniform B-Splines 18
3.3 Regression Spline using Open uniform B-Splines with more knots 19
3.4 Test data clean for reference 20
3.5 Estimate fits good but spiky edges 20
3.6 Estimate gets wiggly with more knots 21

4.1 Smoothing spline: when lambda is 0 tends to interpolation . . . 23
4.2 Smoothing spline: when lambda is very high tends to linear regression 24
4.3 Fit seems to be good 25
4.4 Fit not good for same lambda 25
4.5 Fit not good for same lambda 26

5.1 Data Set one - Clean Data 30
5.2 Data Set one - Regression Spline 30
5.3 Data Set one - using one lambda 31
5.4 Data Set one - using two lambdas 31
5.5 Data Set one - using many Sensor points 32
5.6 Comparision between estimate function and clean data 33
5.7 Data Set two - Clean Data 34
5.8 Data Set two - Clean Data 34

List of Tables

Chapter 1

Background

Due to the environmental variations, a large amount of food produced doesn't arrive in an acceptable state to the consumer. Due to many conditions like different harvest conditions, harvest-to-cooling time and deviating transport temperatures the degradation and deviation of the quality of the food products is inevitable. The telematic units do help in monitoring the temperature but only of the cooling unit, but measuring temperature the of the product is not possible. Moreover, measurements provided by telematics are insufficient for predicting product temperatures and resulting quality deviations, as this can be only possible when the required information is given in real time. So the advantage of the Intelligent container is that it can record spatial temperature deviations using a set of sensors usually between 20-30. The sensors used can measure the temperature of the product.

In this project one fruit i.e the bananas and two meat products (vacuum-packed Irish lamb saddles, pork minute steak in modified atmosphere packaging) have been selected to predict the shelf life and relation to the temperature and other parameters. The actual calculation of the shelf life of the products are beyond the scope of this report , so will be discussed only briefly.

It was found that Ethylene gas is related to the ripening process of several fruits. So the shelf life of bananas depends on the amount of ethylene gas released. Ethylene is a gaseous ripening hormone for fruits and plants. when a fruit ripens, ethylene gas is emitted to the surroundings which in turn ripens the fruits in the surroundings. The amount of ethylene that is emitted depends on the state of ripening. So the measurement of ethylene in a crucial factor. A gas chromatographer has been developed in the laboratory to measures the ethylene gas with time. Several tests under laboratory conditions were conducted.

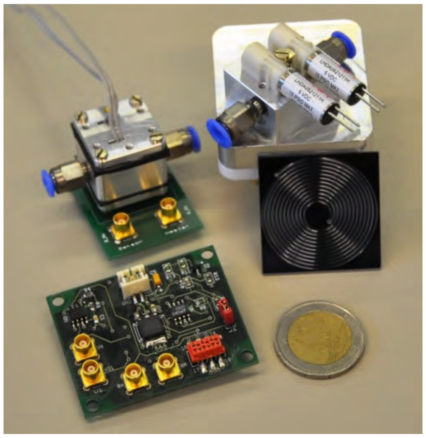


Figure 1.1: Gas chromatography column and other system components

Figure 1.2: Java enabled Preon32 sensor node and Preon Cube with CO2 Sensor

20-30 sensors were mounted to capture local temperature deviations inside a truck or container. Different communication protocols are used for sending the information from the container to the main station. The detailed discussion of the communication protocols are beyond this report.

The main aim of this thesis is to a make a spacial field reconstruction of the received data, using different algorithms smooth the data and send it with minimum parameters. To achieve the same spline regression and smoothing has been used. In particular cubic B-Splines have been used to implement different algorithms. The reason for the usage of the B-Splines has been stated in the next chapter.

Chapter 2

B-Splines and its surfaces

2.1 Introduction

Since the major part of our work is on splines the same will be discussed in detail. In this chapter firstly brief about the theory and mathematics of splines will be discussed. Followed by in-depth details of Bézier curves and B-Splines. It should be noted that though our work is largely based on B-Splines, to understand the mathematics of B-Splines, thorough understanding of Bézier curves is really important.

When a set of points on the plane are considered, to derive a smooth curve that approximates the points, it has to be made sure that the algorithm used is efficient and is suited for computer implementation. To make sure it is not overly sensitive to round off errors, methods involving small number of elementary arithmetic operations should be considered. The sensitivity can be controlled by taking weighted average of the given points in all the operations. To understand the above, let us discuss about the affine and convex combinations initially with two points and then and convex hull of set of points and understand the concepts of numerical stability. So the goal is to develop a numerical method that is insensitive to primary source of problems like round-off errors, in other words the methods should be numerically stable. Let us consider the following equation

$$c = (1 - \lambda)c_1 + \lambda c_2$$

Let c_1 and c_2 be two numbers and λ be a given weight in the range of $[0, 1]$ so the result c will always lie between c_1 and c_2 .One of the main reasons for instabilities in the numerical calculations is the chain of computations containing numbers of large magnitude. The computation of the form of the above equation is known as convex combination. So it can be concluded that if all the computations can be made into convex combinations, then they can be made reasonably stable.

2.2 Bézier curves

Bézier curves are polynomial curves, which when joined smoothly together form more complex curves. Bézier curves not only overcome the problems of wiggles and bulges due to complex chain computations, but also avoids the problem of using high degree polynomials. As mentioned earlier, though Bézier curves are not part of our work, Bézier curves are very important in understanding B-Spline curves, which are main part of this work.

Consider the following equation

$$\sum_{i=0}^{n} a_i x^n = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \tag{2.1}$$

The above equation represents the generalised polynomial form where a are the coefficients and x are the variables. Polynomial equations can be used to draw the curve. The order of the curve is the highest degree used in the polynomial equation .The Higher the degree or the order, the more the changes in direction in the curve.

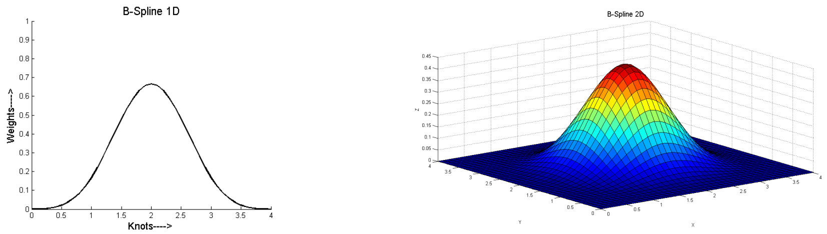
2.3 B-Splines

As discussed above that the order of the Bézier curves determine the change in the direction. The number of changes in the directions will be one less than the order of the curve. So this turns out to be a big disadvantage of Bézier curves because if the need is to form structures with multiple curves in a single line, a very large order of Bézier curve has to be used which can get very complicated and can cause various computational problems.To overcome this problem, one can join several Bézier curves end to end to form a special form of curve called the B-Splines or Basis Splines.A B-Spline curve of degree k defined by n+1 control points will have n-k+1 Bézier curves.

Consider an example of a quadratic B-Spline with 8 control points. Now the number of Bézier curves will be $6-2+1 = 5$ .As in a Bézier curve with degree 2 the maximum number of control points that can be used 3, however, in case of a B-Spline any number of control points can be chosen. Some of the important assumptions that needs to be considered when dealing with BSplines are

- The first derivative at the end of the first Bézier curves is same as the first derivative at the start of the second Bézier curve
- the second derivative at the end of the first Bézier curve is the same as the second derivative at the start of the second Bézier curve
- the final point on the first Bézier curve has the same coordinates as the first point of the second Bézier curve.

The following figures are the B-Splines in 1D and 2D with maxima at 2



(a) 1D B-Spline

(b) 2D B-Spline

Figure 2.1: B-Spline with Maxima at 2

The generalized equation for a B-Spline can be written as

$$s(t) = \sum_{i=0}^{n} N_{i,k}(t)P_i \tag{2.2}$$

where N is the Basis function and P are the control points[**?**] B-Splines can be defined using a very complex formula, Cox de Boor recursion formula which is given as[**?**]

$$N_{i,0} = \begin{cases} 1, & \text{if } \; t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

$$N_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t) \tag{2.4}$$

where t represents the knot sequence which is always in the range of $[t_0, t_m]$ and also determines what effect basis function causes to the curve. t_i represents a knot,and the values of t_i is taken from the knot vector,

$$T = (t_0, t_1, t_2 ... t_m)$$

If we consider a B-Spline of kth order with $n + 1$ control points, then number of knots $m + 1$ is given as

$$m = n + k + 1$$

For example, if we have 6 control points $P_0 ... P_5$ and for cubic B-Spline i.e. with degree 3, then it requires $m = 3 + 5 + 1$.The the Knot vector is given by, $T = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$.

In the above triangular representation, $N_{0,k}(t)($ in the far right) needs to be calculated, which is the sum of $N_{0,k-1}$ and $N_{1,k-1}$ as shown in equation 3.2 which in turn are the sum of previous values. Thsi process is recursive and gives either 0 or 1 when level one is reached as shown in equation 3.3.

The above equations give rise to a very special form of B-Splines i.e. uniform B-Spline. Where the knots are equidistant to each other. If we solve the above equations, we get the required splines. Depending on the position of the t we get a different basis function. Please refer to Appendix A to find the method of calculation.Only one of the calculations have been provided for understand purpose.

So which gives us the figure as of the uniform B-Splines.

The above figure gives two information.Firstly, multiple B-Splines equidistant to each other, i.e. uniform b-Spline gives a flat surface.This is one of the main reasons why we used uniform knots throughout this work.Secondly, for this particular example, it is only $[t_2, t_3]$ where our basis function effecting our B-Spline. So our B-Spline is only defined between $t = 2$ and $t = 3$.

This can be overcome by a new form of B-Splines,open uniform B-Spline. In this,the recurrence of the knots which play the main role.For example we want to define basis function between $[0, 1, 2, 3, 4]$.The first and last knot has to be repeated, which give us the following vector sequences.From now on we will be using cubic B-Splines in our work. One of the special properties of B-Splines with degree 3 is that their first and second derivatives are continuous.And it is because of this property of lower order continuity, splines are very smooth

- [00001]
- [00012]
- [00123]
- [01234]
- [12344]
- [23444]
- [34444]

Calculation of $[3, 4, 5, 5, 5]$ can be seen in Appendex A as an example. In the above list of knot sequences, one with four repetitions give quadruple start and end splines, similarly we get triple recurrence start and end splines, double start and end splines, and basic splines in the middle.Now the basis functions have been calculated for each of the above knot vectors using the $equation - 3.4$.The calculation have not been shown because of their complicity.In the figure shows open uniform B-Splines, in which the splines unaffected by the basis function have been truncated.

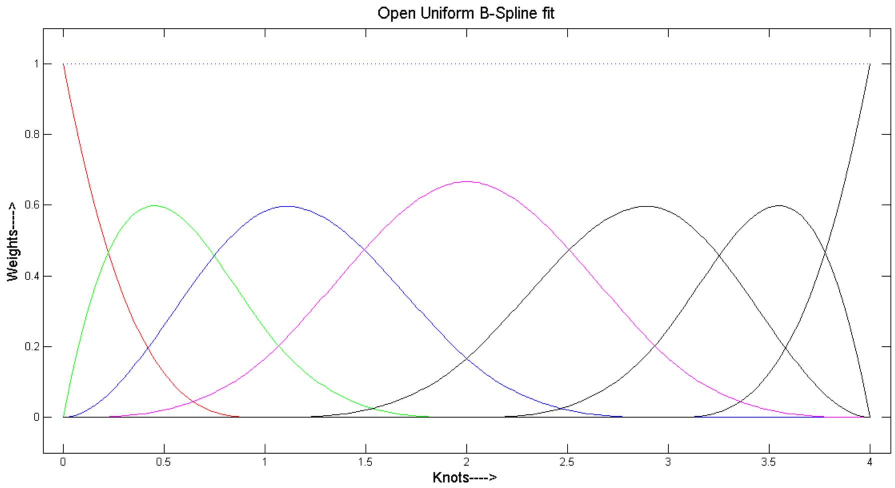


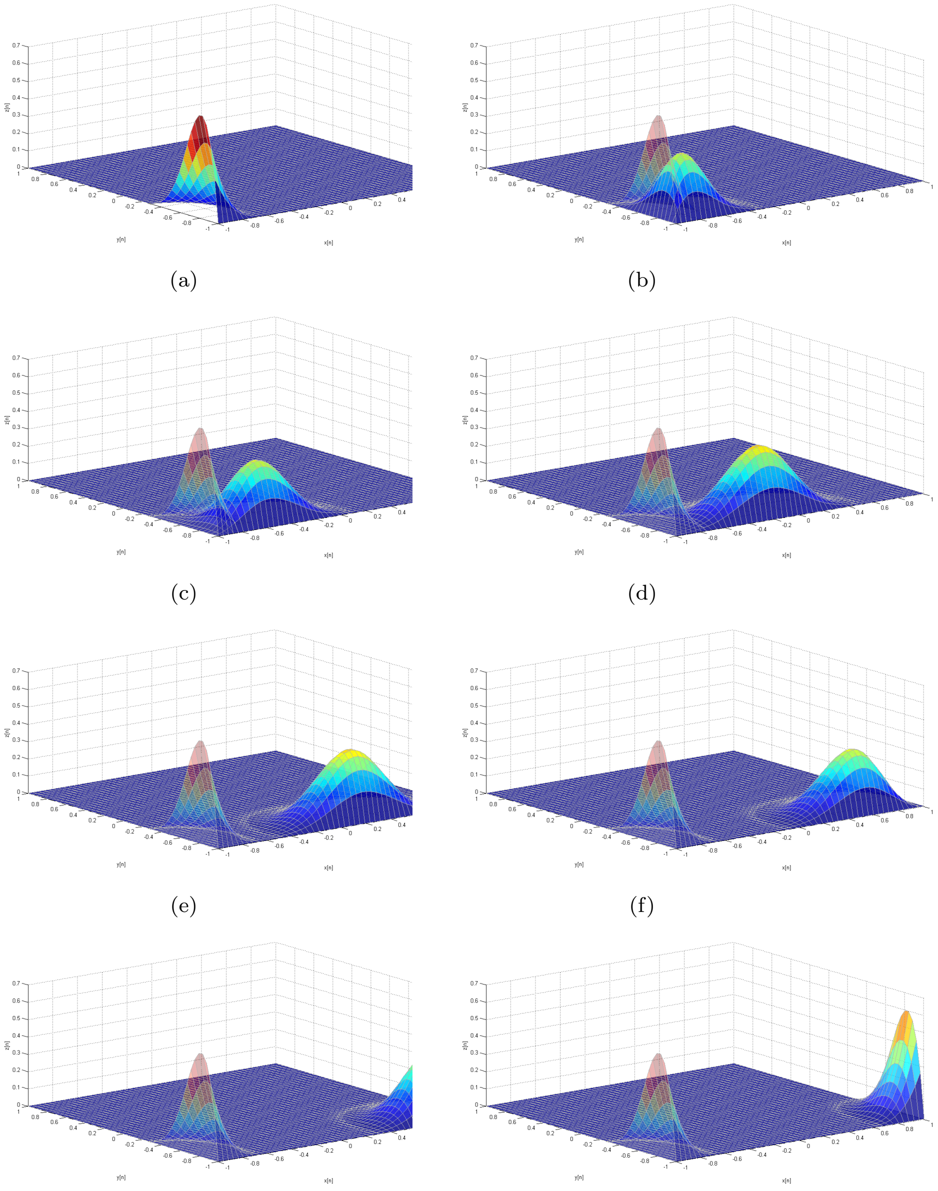
Figure 2.2: Open uniform B-Spline - 1D

2.4 B-Spline Surfaces

In the previous sections B-Splines were constructed using vectors in one-Dimensions. Taking the process to higher dimensions, the B-Splines can also be represented in two-dimensions. This is obtained by tensor product of rectangular set of control points referred to as control mesh in two dimensions. From the above assumptions the B-Spline surface can be defined as

$$s(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,k}(u) N_{j,l}(v) P_{i,j} \tag{2.5}$$

Extending the equation 3.2, $N_{i,k}$ $N_{j,l}$ are the basis functions of B-Splines of degree k, l respectively, where the degree k is in u direction and l in v direction. The following figure shows the plotting of the B-Spline surface. The B-Spline surface is plotted by keeping the basis function in u direction fixed while changing the v direction basis function. The figure shows only one of the directions calculations.[9]



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 2.3: Calculation through tensor product by keeping one of the directions constant while varing the other

Adding all the splines the following open uniform B-spline surface is obtained

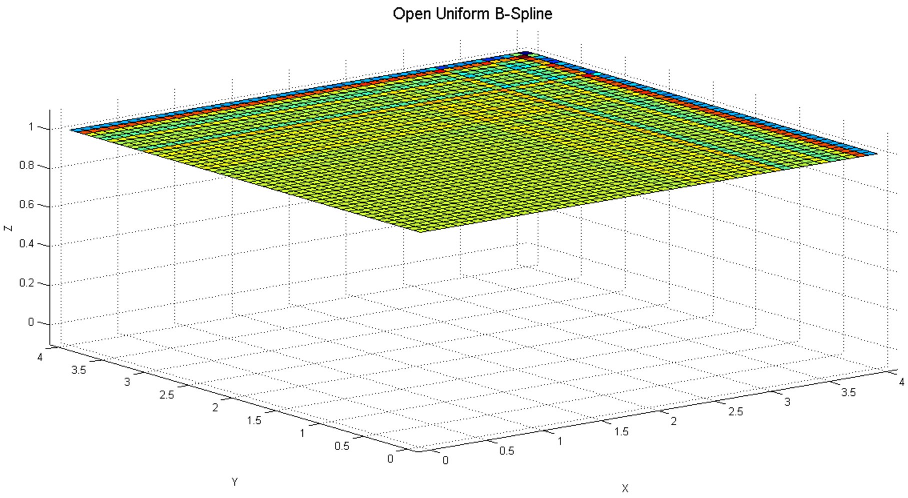


Figure 2.4: Open Uniform B-Spline 2D

It can be seen in the above figures, that recurrence of the knots gives straight edges. Based on these two properties, i.e smooth surface and straight edges we proceed with further implementations and tests. Since basic splines plot has been made ,the next step is to find regression of splines, to find the estimate function for a data set which has been covered in detail in the next chapter.

Chapter 3

Estimation through Regression

3.1 Introduction

In the previous chapter we had a thorough discussion about B-Splines and two of its properties.Two of its main properties have been discussed and implemented, which will be used in for the further process. We proceed further by estimating the curve using regression. A running example will be used through out this chapter as well as this book, to discuss different concepts and motivate different methods for estimation. The data is generated from the function given below with random noise included in the target values

$$y(t) = 2e^{(-0.4(x-2)^2)} + \frac{5}{x+10} + 0.1x - 0.2; \tag{3.1}$$

it is always advisable to use a pre-defined artificial example (as shown above)as one can easily use it for compression with the estimation, and understand the precise process the data is generated.

Firstly, an estimate will be made of the above mathematical function using Uniform B-Splines. The the estimate will be made in 1D as well as 2D using open uniform B-Splines.Let us assume for a given samples set $(x_i; y_i)$, where $i = 1...n$, the regression function

$$r(x) = \mathbb{E}(Y|X = x)$$

has to be estimated. It can be achieved by fitting 3rd order B-splines (cubic spline) at some pre specified locations $t_1...t_m$.

The same regression technique has been used in this process. Least square regression has been achieved by pseudo inverse operator. Firstly, the spline functions has been calculated for each of the spline which were then pseudo in-versed with the sensor values, which gave the weights or the coefficients for each of the splines.

In code

In the figure below the Uniform B-Splines have been calculated using the function $bSpline3$, and the estimate has been made for the function $dummyCurve$. The weights have been calculated using pseudo inververse operator. Since number of splines splines used here are 13, number of weights obtained are 13.

The estimation has been made using the regression splines, as shown in the figure below

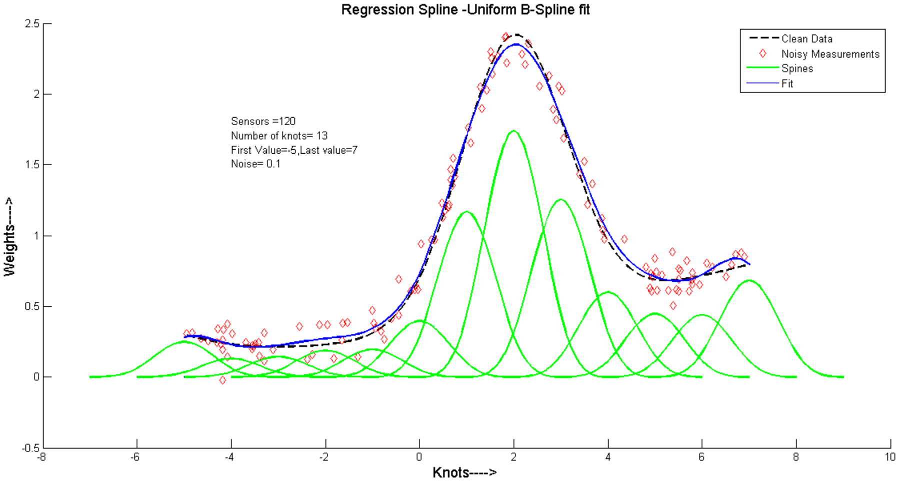


Figure 3.1: Regression Spline using uniform B-Splines

In the above figure we have used uniform B-Splines, to prove its disadvantage discussed in the previous chapter.We can see that the basis function has no effect on the end splines,so this extra part is truncated by using open uniform B-Splines.

In code

The functions used to generate the shown splines are *Basis_Spline_modified*, *Double_reccurence_start_modified*, *Double_reccurence_end_modified*, *triple_reccurence_start_modified*, *triple_reccurence_end_modified*, *quadruple_reccurence_start_modified*, *quadruple_reccurence_end_modified* to make regression on the function *dummyCurve*. Unlike for the uniform B-Splines when weights were calculated, the number of weights will be *two* more than the number of knots. As shown below the number knots are 14 but the number of splines are 16.

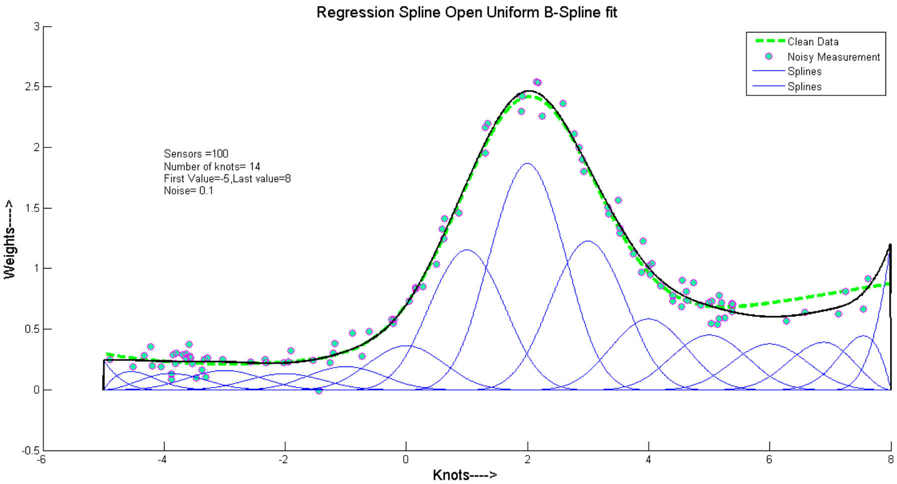


Figure 3.2: Regression Spline using Open uniform B-Splines

In the above figure, with the use of open uniform B-Splines,it can be observed that the spline have smooth edges .i.e. the non effected part has been truncated. Moreover, the fit is almost as expected. We will be using Open uniform B-Splines in one-Dimensional as well as two-Dimensional in our work,for further calculations.

Regression splines are really good for finding an estimate, only when a proper value of knot vector is used.However,using a correct knot vector is really tricky and has been in discussion for decades.Furthermore if we increase the knot points the fit starts getting wiggly, that has been shown in the figure below. We can also see that the splines in the end are very sensitive to the points. Even though there is no point in the end the fit tries to find a point based on the previous point position.Due to which we get an undesirable output.

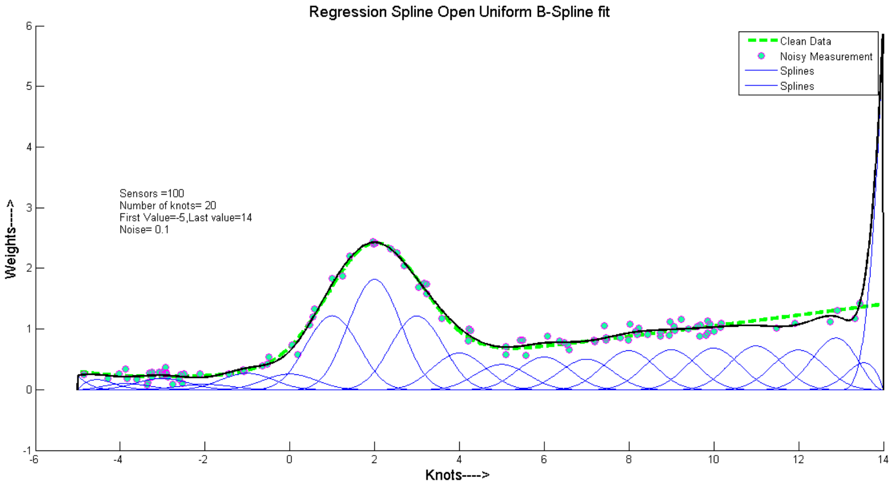


Figure 3.3: Regression Spline using Open uniform B-Splines with more knots

Some similar kind of results were also found when implemented in two dimensional.When the estimate was made for the reference data shown in the first figure below, the estimate was good. But as it can be seen due to the high sensitivity of the splines, the we are getting an unexpected result. Also when we increase the number of knots the fit gets wiggly.

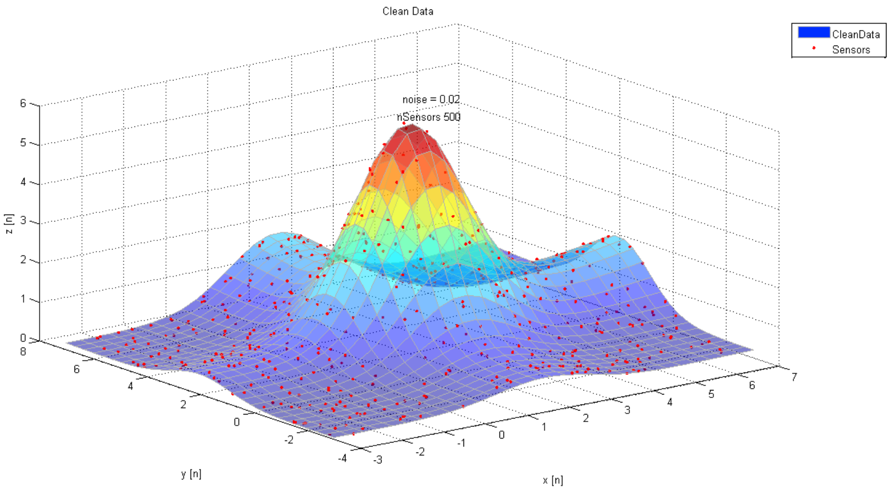


Figure 3.4: Test data clean for reference

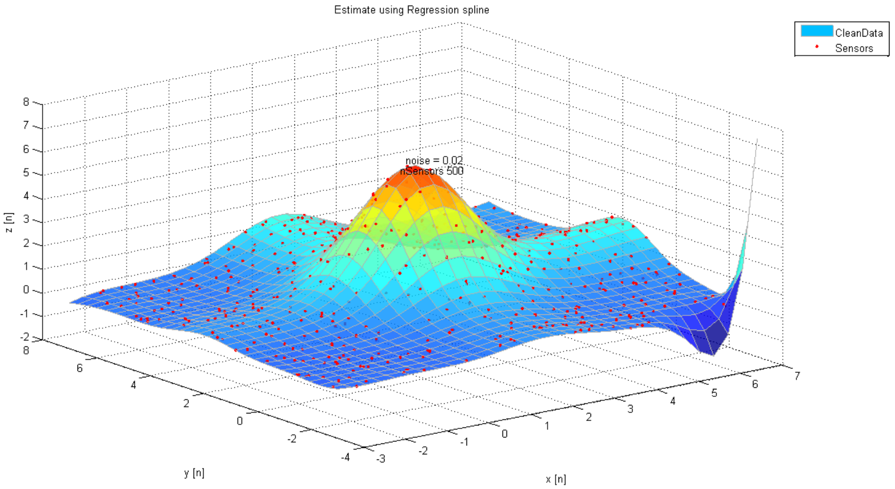


Figure 3.5: Estimate fits good but spiky edges

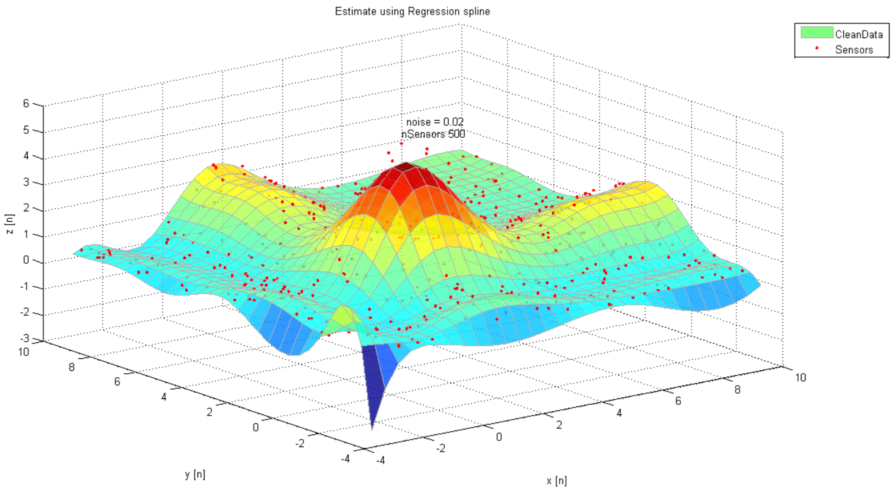


Figure 3.6: Estimate gets wiggly with more knots

Chapter 4

Smoothing with Smoothing Spline

4.1 Introduction

As we have seen in the previous chapter, that the regression splines are very sensitive to data points.Moreover if we increase the number of splines the fit tends to become more and more wiggly.One of the best ways to overcome the problems caused by regression splines is smoothing splines. Smoothing spline can be defined as a method that uses a spline function to fit a smooth curve over a set of noisy measurements.

This chapter introduces and investigates some of the aspects of smoothing splines .Firstly we will build up a concrete theoretical background and then we will concentrate on implementation and simulation with some test data. Then we discuss upon some of the disadvantages of smoothing splines when parameter is chosen manually and how those can be overcome by using automatic smoothing parameter selection algorithms.

4.2 Theory and Generalization

Consider a set of n pair of measurements (x_i, y_i) where $i = 1, \ldots\ldots, n$related to the model where

$$y_i = f(x_i) + \varepsilon_i \tag{4.1}$$

Where x_i is a known sequence, f is an unknown function and ε_i are independent random variables. So the task is to produce an estimate of f .The estimate can be obtained by the minimization of

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(x)]^2 + \lambda \int_0^1 [f^{('')}(x)]^2 du \tag{4.2}$$

In the above equation, the first term represents the RSS, residual sum of squares, which tries to make f(x) match the data at each x_i . The second term in the equation penalizes the roughness of the fit. The second term contains the third derivative of the squared function. The third derivatives pick up wiggles of the function and the integration adds up all the non linearity.

The λ in the above equation is a positive constant known as the smoothing parameter or roughness penalty .The value of λ usually lies in the range of $[0, 1]$.[2] It controls the trade-off between bias and variance of the estimate. Smaller the value of the roughness penalty the more wiggly the function gets[3]. Therefore as the $\lambda \to 0$ the smoothing spline starts to interpolate the data points.Similarly, as the $\lambda \to \infty$ the smoothing spline approaches to a simple linear regression spline

4.3 Implementation

Some more info But as from the theory discussed above, the behaviour smoothing spline can be seen by setting two extreme values.So let us consider the following results

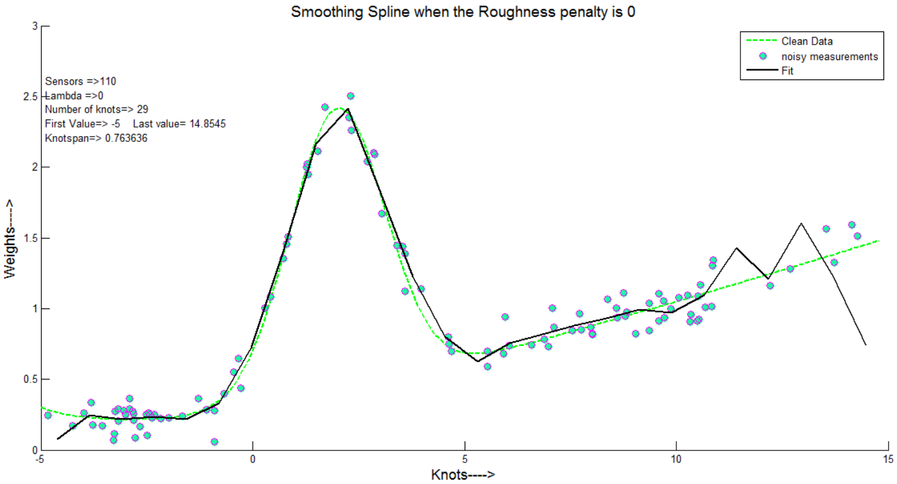


Figure 4.1: Smoothing spline: when lambda is 0 tends to interpolation

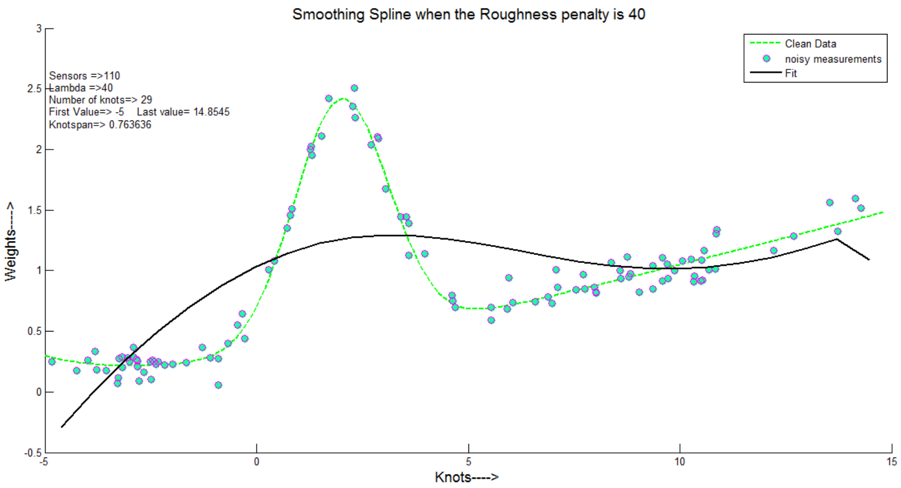


Figure 4.2: Smoothing spline: when lambda is very high tends to linear regression

The first figure shows that when the value of $\lambda = 0$ or very low, the plot almost interpolates and the fit is wiggly, which is nothing but the regression spline shown in the last chapter. So when more knots are taken, the fit gets more wiggly and the becomes more sensitive to the sensor points. In other words, the data are over fitted. However, when the value of lambda is very high i.e λ =40 as in the second figure, the fit tends to linear regression. So a value has to chosen in such a way that one gets a smooth fit[1].

However there are two disadvantages if roughness penalty is chosen manually. Firstly, we cannot really determine the exact value of λ as there can be a very large number of solutions of the smoothing parameter. Secondly, even if a perfect λ is found manually for a function, it may not fit for other functions.

For example, consider the example give below, when lambda is chosen manually as .9. As we can see there is very less wigglyness and fits the clean data let us assume its an optimal value of lambda.However when the same value of λ is chosen for other functions, the estimate tends to be overfitted or underfitted as shown in th following figures

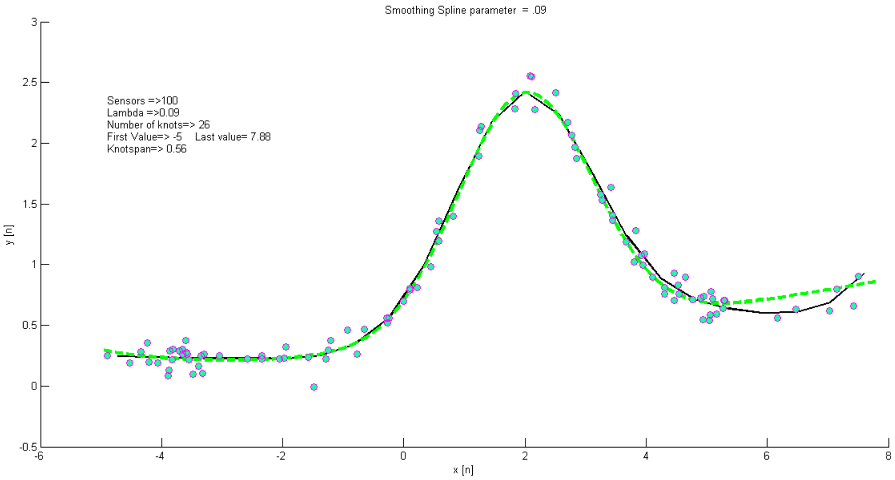


Figure 4.3: Fit seems to be good

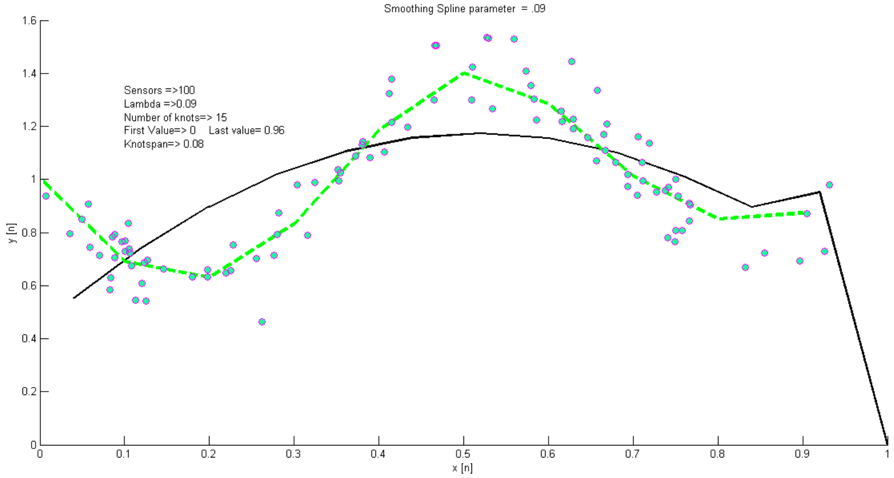


Figure 4.4: Fit not good for same lambda

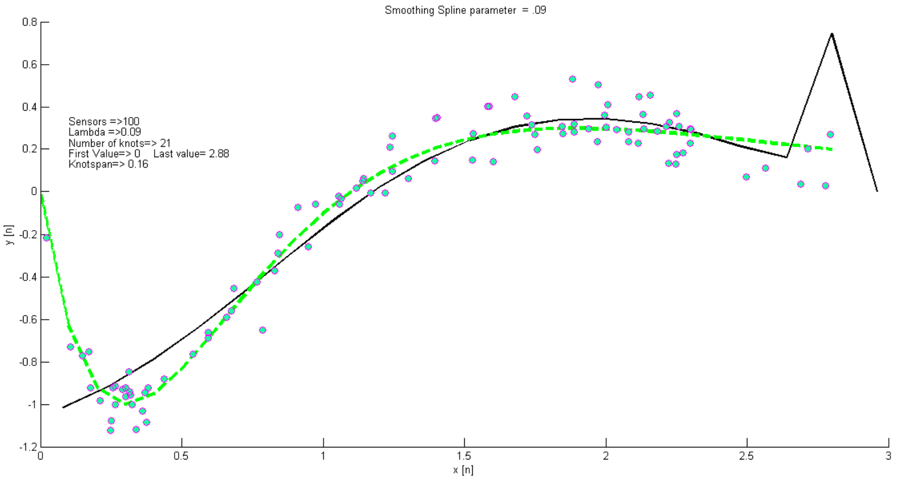


Figure 4.5: Fit not good for same lambda

Therefore it should be noted that obtaining the value of correct λ is really important as it has a crucial effect on the quality of the estimate. So some special algorithms have to be used for choosing the value of λ in a way that not only it fits the data in the best way possible but also it can can be used for different functions for choosing.So the next section describes about different methods used to select the smoothing parameters

4.4 Parameter selection Methods

One of the crucial point in B-Spline smoothing is the selection of smoothing paramerter and selection of the number of knots[4] . Method of model selection has attracted the researchers since decades .Several attempts have been made for the selection of smoothing parameter like cross validation (Stone, 1974), cross validation with the trace of hatmatrix, Generalised Cross validation (G.Wabha, 1979). In this section we will discuss, about the theory behind parameter selection alogrithms, and then test the algorithms on different function.

Cross validation

One of the most well known methods is the method of cross validation. The basic idea of cross validation is to achieve an algorithm and evaluating its behaviour on new set of data points. In its application, the data is split and so one part of the data is used for used for evaluating the performance of the algorithm and other part is used for training the algorithm (Arlot). The major part of this process depends on the way the data samples are split . We have mainly used two types of ordinary cross validation techniques which are leave out one cross validation, and cross validation with the trace of hat matrix.

Let us firstly discuss briefly about the two of the methods mentioned above, used in our work . The first is the leave out one cross validation which corresponds to the choice of n-1 data points. In this process all the data points are trained using a function approximator except for one data point, whose estimation is made. Now the root mean square error is evaluated and the smoothing parameter with the least RMSE value is used to generate the estimation.[1]

The equation for ordinary cross validation function $V_0(\lambda)$ can be given as

$$V_0(\lambda) = \frac{1}{n}\sum_{k=1}^{n}\left[y_k - f_\lambda^{[k]}(x_k)\right]^2 \tag{4.3}$$

Cross validation with Hat Matrix

Let $((H_\lambda))_{ii}$ be the i^{th} diagonal element of H_λ. So the equation for Cross validation using hat matrix is given by [1, 3]

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{y_i - \hat{f}_\lambda(x_i)}{1-(H_\lambda)_{ii}}\right]^2 \tag{4.4}$$

Generalised cross validation

The equation for generalised cross validation is given by [1, 3, 5]

$$GCV(\lambda) = \frac{1}{n}\frac{\sum_{i=1}^{n}\left[y_i - \hat{f}_\lambda(x_i)\right]^2}{\left[1-n^{-1}tr(H_\lambda)\right]^2} \tag{4.5}$$

It can be seen that, the estimate is a good fit with all the three algorithms. The wigglyness is reduced and unlike as seen in the regression splines the ends do not have unexpected edges.The the other figure shows the the behaviour of RMS with change in lambda.

Similar results can be seen in two-Dimensional.

4.5 Timings Tables

It can be seen that all the algorithms used above give a good result, for one dimensional as well as two dimensional data set. But one of the biggest disadvantage of these is their search algorithm, which is time consuming and, optimizing the time is one of the important criteria of our work. Following are the timing table and the graphs. The graph is a Sensor verses time graph. Though the results are good, and gives a good fit, the overall the timing are very high. Moreover, it can be that, the graphs are almost identical, which shows that as the number of sensors are increasing the time of computation is more, but has hardly any effect with the change of noise.

4.6 New Search Algorithm

As of now, the RMSE is calculated for each of the λ in the range, and λ for minimum RMSE is chosen as the optimal Smoothing parameter.For example, if the the λ is in the range of [.001 : .001 : 1] it calculates λ for all the 1000 values and accordingly the λ is chosen. But what if the optimal λ lies between $[.001, .005]$, RMSE will still be calculated till λ is 1. As it be seen that there is a lot of additional calculations in this process, which have to be minimised.

Keeping the above in mind, this algorithm has been developed with the main intention to reduce the number of calculations and to increase over all speed of the process. Firstly all the redundant calculation have been eliminated. As it was seen that RMSE first decreases and then increases, and λ is chosen for smallest value of RMSE .The elimination of the redundant calculations was achieved by stopping the calculations as soon as the RMSE starts increasing. This technique reduces the overall calculations, so eventually amount of computations are decreased.

Secondly,in this method two new parameters have been introduced, resolution, and the step. For example, if the calculation starts at a value of $\lambda = .0001$, it increases at a step size of .04 (which can be defined manually if needed). It checks if the optimal value lies in the range of .0001 and .0401. If yes, it takes .0001 and .0401(instead of 1) and starts the process again, else it keeps on inceaseing the step size to find the window in which the lambda lies.Once it finds new maximum and minimum limits, it increases the resolution.The resolution increases till 5 decimal points(which can be changed manually.

Let us test the new algorithm with the previous test data sets and compare the timings accordingly.The following table compares timing of different algorithms with and without this algorithm.

Chapter 5

Simulation and Results

The use of the algorithms on different test data has been shown in previous chapters. In this chapter the algorithms will be tested on real data provided. Two provided set data sets will be used for testing of the algorithms and and comparison of the results .The algorithms will be tested on each of the data set individually, and then smoothness and the time results will be compared for different algorithms using different noise values and number of sensors.

5.1 Data Set 1

In this section the first data set will be dealt. It should be noted that this data set provided was generated by mathematical function.

The following figure shows the clean data, which has to be predicted using the implemented algorithms. 200 sensors and noise level of .01 has been set as the initial parameters. The parameters will be varied in the further tests.Since,it was already seen that the regression spline have some disadvantages, in this chapter the two will be compared using the RMSE.

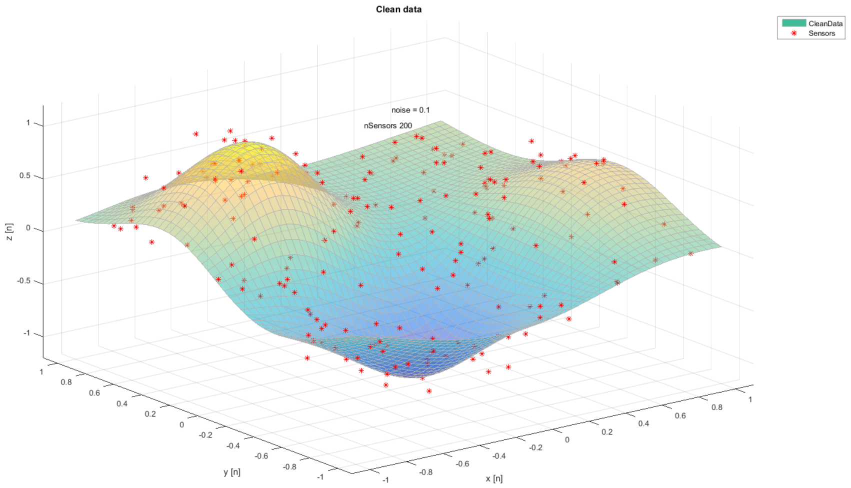


Figure 5.1: Data Set one - Clean Data

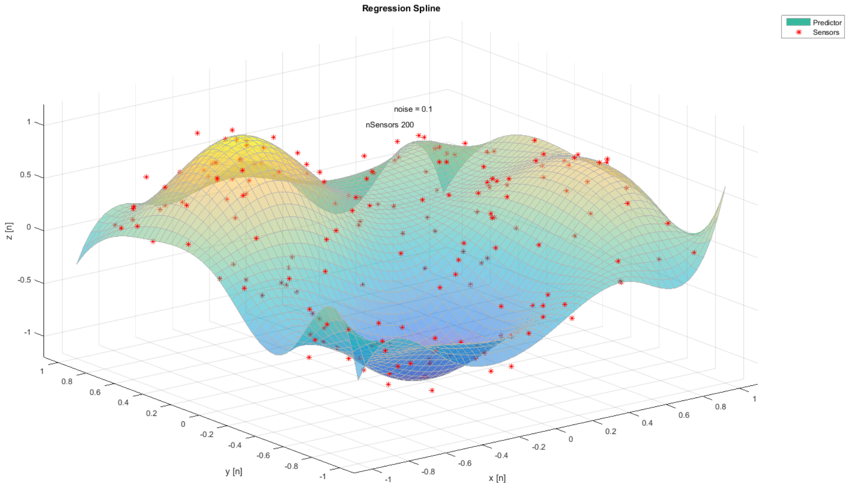


Figure 5.2: Data Set one - Regression Spline

Initially, λ is set maually. Similar to what was seen in previous smoothing spline figures, it can be seen that the edges of the curve are not getting smoothed up. It has been found that the splines are very sensitive to sensors points, and it is happening due to insufficient data points.

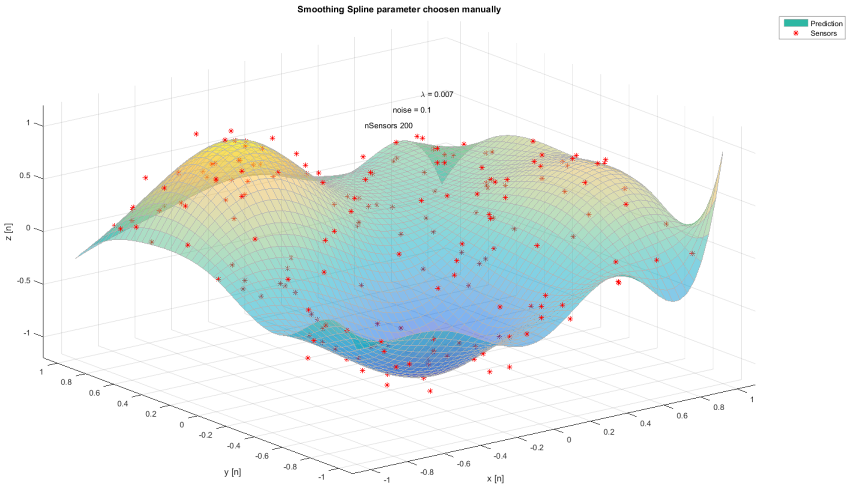


Figure 5.3: Data Set one - using one lambda

This can be demonstrated by using different values of lambdas for splines. The end splines i.e the quadratic triple and double end splines were given higher λ, and other splines were given general value of λ, but the ends still show peak edges

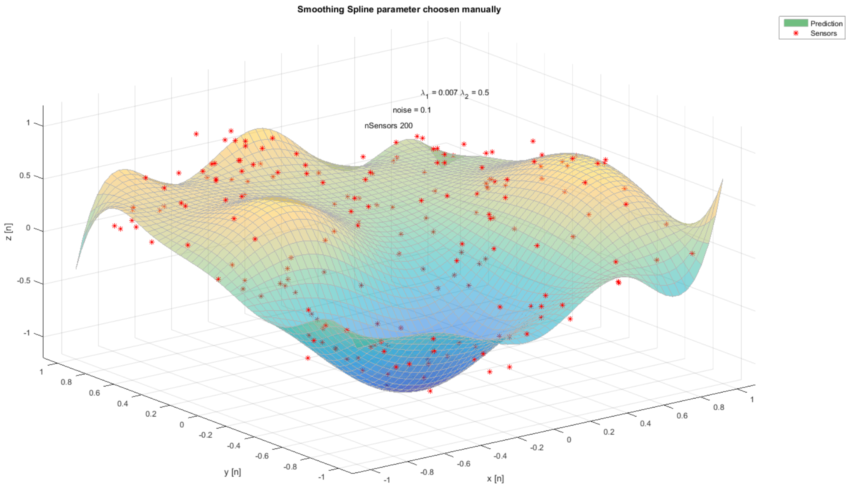


Figure 5.4: Data Set one - using two lambdas

In the following figure it can be clearly seen that the edges are smoothed up even though same value of λ has been used, but increased the number of data points .So it can be concluded that the splines are sensitive to data points.

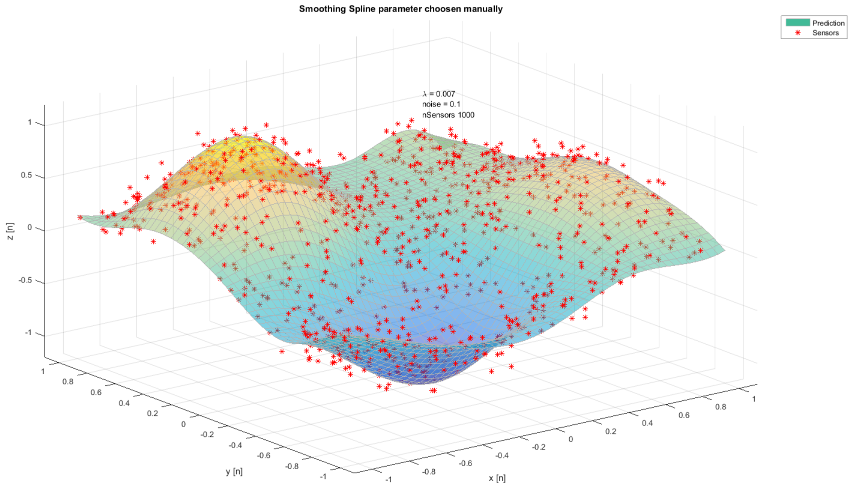


Figure 5.5: Data Set one - using many Sensor points

In the figure the estimate function has been found using ordinary cross validation, and a comparison has been made between clean data and the estimated data.

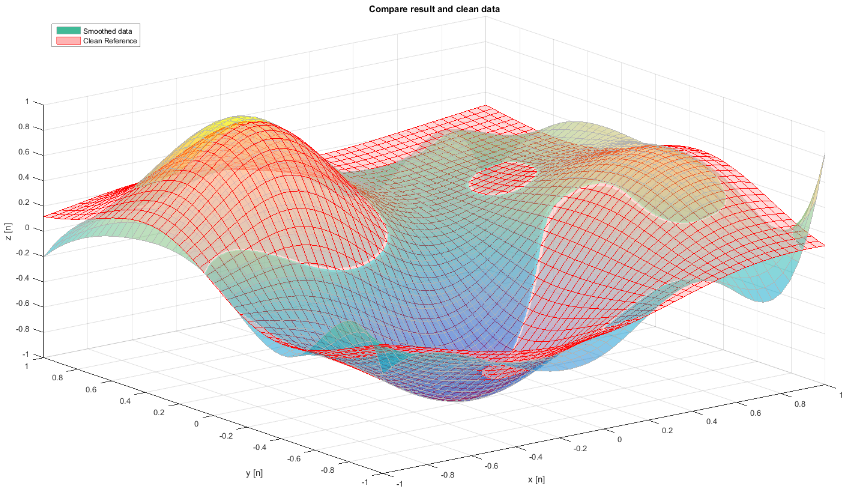


Figure 5.6: Comparision between estimate function and clean data

5.2 Data Set 2

This data set contains a simulation of of 3 heat sources in a water basin, which also includes the effects of diffusion, the influence of walls and the horizontal heat transfer by a small flow. Initially the noise is set to .01 and 400 sensor points. Similar to the data set one, These parameters will be compared to check the accuracy of the estimate. Consider the following figure, which represents the clean data. This can be used as reference to check the accuracy of the estimated field[8] [**?**]

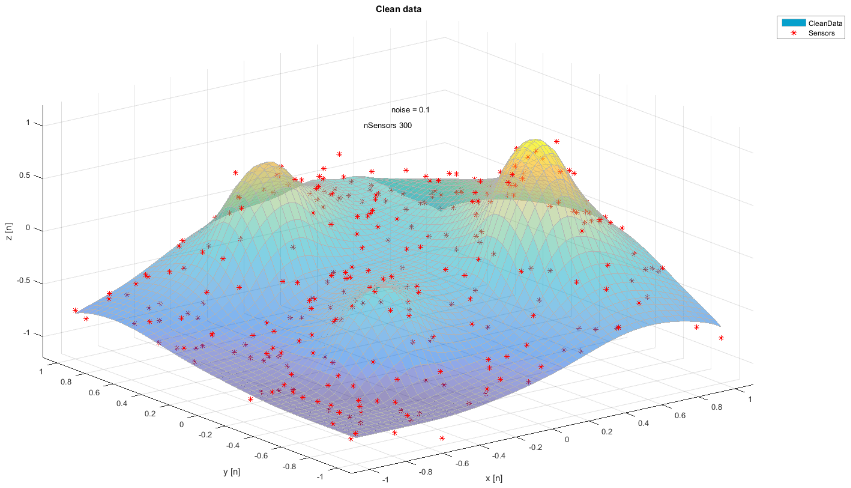


Figure 5.7: Data Set two - Clean Data

Following is the figure of the estimate of data set two, where λ has been chosen manually.

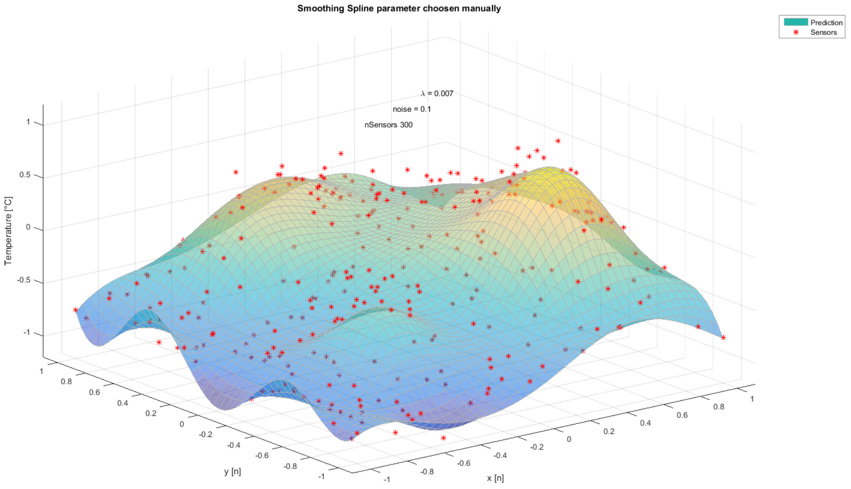


Figure 5.8: Data Set two - Clean Data

Chapter 6

Summary and Conclusion

The complete mathematics related to B-Splines has been shown in One-Dimensional and then was extended to two dimensional B-Spline surfaces for both uniform and open uniform B-Splines. Once the splines were calculated, regression was made using least squares. It was found that though regression was made and prediction a estimate function was found, the regression splines had a problem of wigglyness with more knot insertions.

The alternative way to find the estimate was through smoothing spline. It was found that setting an optimal smoothing parameter was crucial, so cross validation, cross validation with hat matrix, generalised cross validation algorithms were implemented to find the optimal smoothing parameter. It was found that the time take to find the optimal smoothing parameter was almost the same by all the three methods. A special algorithm was also implemented which reduced the number of calculations and reduced the time almost to half.

Finally the implemented algorithms were tested on the given data set. Two data examples were used to test the work.

Bibliography

[1] Thomas C.M. Lee *Smoothing parameter selection for smoothing splines: a simulation study*. Department of Statistics, Colorado State University 2002.

[2] Tatyana Krivobokova *Dissertation zur Erlangung des Grades eines Doktors der Wirtschaftswissenschaften (Dr. rer. pol.) der Fakult at f ur Wirtschaftswissenschaften der Universit at Bielefeld*. 2006.

[3] R.L.Eubank *THE HAT MATRIX FOR SMOOTHING SPLINES*. Department of Statistics, Southern Methodist University, Dallas, TX, USA 1983.

[4] Dennis L Young *Knot selection for least-squares and penalized splines*. Journal of statistical computation ans simulation 2012

[5] Dursun Aydin , Memmedaga Memmedli, Rabia Ece Omayx *Smoothing Parameter Selection for Nonparametric Regression Using Smoothing Spline*. European Journal of pure and applied Mathematics 2013

[6] Roger Koenker *Smoothing Things Over: Some Notes on Cross-Validated Smoothing Splines*. Department of Economics,University of Illinois 2014

[7] Seiya Imoto and Sadanori Konishi *Selection Of Smoothing Parameters In B-Spline Nonparametric Regression Models Using Information Criteria*. The Institute of Statistical Mathematics 2003

[8] Jedermann, R. and Paul, H. *Identifying methods for accurate reconstruction of spatial fields in wireless sensor nets under data transmission limits. International Journal of Distributed Sensor Networks,*. Hindawi 2016

[9] Michael S. Floater
`http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5340/v07/undervisningsmateriale/kap7.pd`
2003

[10] Michael S. Floater
`http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5340/v07/undervisningsmateriale/kap2.pd`
2003

Functions and Files in the code

- **bSpline3**
- **dummyCurve**
- **Basis_Spline_modified**
- **Double_reccurence_start_modified**
- **Double_reccurence_end_modified**
- **triple_reccurence_start_modified**
- **triple_reccurence_end_modified**
- **quadruple_reccurence_start_modified**
- **quadruple_reccurence_end_modified**
- **knot_calculation**
- **calculate_spline**
- **plot_Spline**
- **Plot_Basis**
- **Basis_cal_one_point**
- **calcSpline1D_Single**
- **Calculate_Basis**

Appendix A

triple reccurence end
[34555]

$$\tfrac{x-3}{5-3}[3,4,5,5] \qquad + \qquad \tfrac{5-x}{5-4}[4,5,5,5]$$

$$\tfrac{x-3}{5-3}[3,4,5] + \tfrac{5-x}{5-4}[4,5,5] \qquad + \qquad \tfrac{x-4}{5-4}[4,5,5] + \tfrac{5-x}{5-5}[5,5,5]$$

$$\tfrac{x-3}{4-3}[3,4] + \tfrac{5-x}{5-4}[4,5] + \tfrac{x-4}{5-4}[4,5] \qquad + \qquad \tfrac{x-4}{5-4}[4,5] \text{ Ignoring terms with 0 denominator}$$

$$\tfrac{(x-3)(x-3)(x-3)}{4}[3,4] + \tfrac{(x-3)(5-x)}{4}[4,5] + \tfrac{(x-3)(5-x)(x-4)}{2}[4,5] + \tfrac{(5-x)(x-4)^2}{1}[4,5]$$

The above expression when used as a Matlab function, gives a triple recurrence end B-Spline