MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

A Framework for Fitting Functions with Sparse Data

Reza HOSSEINI, Akimichi TAKEMURA and Kiros BERHANE

METR 2013–33 December 2013

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

A framework for fitting functions with sparse data

Reza Hosseini[1], Akimichi Takemura[1], Kiros Berhane[2]
University of Tokyo[1], University of Southern California[2]
[1]reza1317@gmail.com

Abstract

This paper develops a framework for fitting real functions with domains in the Euclidean space, when data are sparse but a slow variation allows for a useful fit. We measure the variation by Lipschitz Bound (LB) – functions which admit smaller LB are considered to vary more slowly. Since most functions in practice are wiggly and do not admit a small LB, we extend this framework by approximating a wiggly function, f, by ones which admit a smaller LB and do not deviate from f by more than a specified Bound Deviation (BD) across the domain of interest. In fact for any arbitrary positive LB one can find such a BD, thus defining a trade-off function (LB-BD function) between the variation measure (LB) and the deviation measure (BD). We show that the LB-BD function satisfies nice properties such as monotonicity (non-increasing trend) and convexity. We also present a method to obtain it using convex optimization. For a function with given LB and BD, we find the optimal fit and present deterministic bounds for the prediction error of various methods. This result is an extension of the case with no deviation in the literature. Given the LB-BD function, we discuss picking an appropriate LB-BD pair for fitting and calculating the prediction errors. The developed methods can naturally accommodate an extra assumption of periodicity to obtain better prediction errors. Finally we present an application of this framework to air pollution data with sparse observations over time.

Key Words: Lipschitz Bound; Sparse data; Interpolation; Approximation; Linear Interpolation; Convex Optimization; Periodic Function

1 Introduction

This paper investigates the problem of approximating (fitting) functions when data are sparse over time or spatial domains of data (with a special focus on the 1-dimensional domain case). Such “data-sparse” situations are often encountered when collecting large amounts of data is expensive or practically implausible. For example in many air pollution studies including the *Southern California Children Health Study*, (Franklin et al. (2012), Gauderman et al. (2004), Gauderman et al. (2007)), only sparse data are collected over time for concentrations of several elements (metals, gases) in some homes and schools in Southern California to assess the effect of air pollution exposure on children’s lung function. Using such sparse data, we are interested in approximating the exposure for a given location over a time period of interest. In such cases some properties of the data might allow for a good approximation (prediction/fit) despite the sparse data structure. For example for Ozone concentrations in Southern California, bi-weekly measurements (the measuring filters are installed and collected in such periods) are available and therefore the process over time varies “slowly”. Figure 1 depicts the biweekly moving average of Ozone concentration for a central site (in Upland, CA) in Southern California (where complete data are available) during 2004–2007. There might be also other properties of the process that help us fit the function in sparse data situations. For example many processes over time show an approximate periodic pattern on annual scale (e.g. weather and air pollution). The approximate periodicity for the Ozone process in Southern

California can also be seen in Figure 1. This work utilizes such properties to improve the methods of fitting.

When we are working with (at least one-time) differentiable real functions defined on the d-dimensional Euclidean space $\mathbb{R}^d$, we can naturally define a measure of variation of the function f by the supremum of its first-order derivative (or gradient for multidimensional case) on the domain: $\sup ||f'(x)||$, $x \in D$, where $||.||$ is the Euclidean norm. Of course this definition is not useful for most processes we encounter in the real world – even if they show some global slow-variation – because often there are irregular small variations which make the function non-differentiable (see Figure 1). Another problem with this definition is the domain D needs to have sufficient properties for the derivative to be well-defined (one such sufficient property is D being an open subset of $\mathbb{R}^d$).

The key concept we use in this paper is a measure of variation (or roughness) for general non-differentiable functions on a given domain. At first we consider functions which admit a Lipschitz Bound (LB) on the specified domain and assign the variation of the function to be the infimum of all such bounds. A function $f : D \subset \mathbb{R}^d \to \mathbb{R}$, where D is a subset of $\mathbb{R}^d$ is said to have Lipschitz Bound (LB) m if $|f(x) - f(y)| \leq m||x - y||$ where $||.||$ denotes L^2 norm. The interpolation of functions with a given Lipschitz Bound is also considered in Gaffney et al. (1976), Sukharev (1978), Beliakov (2006), Sergeyev and Kvasov (2010) and the optimal interpolator (which is a piece-wise linear function) in terms of the worst-case error is found.

The Lipschitz framework immediately includes piece-wise differentiable functions but this generalization is still not adequate (useful) for processes we encounter in practice because they do not admit a small enough Lipschitz Bound for the fits or the prediction errors obtained may not be reasonable. We call such functions “wiggly” functions. (Note that this is not an accurate mathematical definition.) As one of the contributions of this work, we extend this framework by approximating a wiggly function $f : D \subset \mathbb{R}^d \to \mathbb{R}$ which does not admit a small enough LB by another function g, which does admit a small LB and deviates from f only by a small “Bound Deviation” (BD), σ in terms of the sup norm: $||f - g||_\infty = \sup\limits_{x \in D} |f(x) - g(x)|$. Then we find the optimal approximation of a given function f with known LB and BD and provide the prediction (approximation) errors for the optimal solution and other standard approximation methods, thus extending the results in Gaffney et al. (1976), Sukharev (1978), Beliakov (2006), Sergeyev and Kvasov (2010) to a much more practical class of functions.

Another key observation is: for each given LB=m, (which may not be satisfied by f), we can consider all the functions g which satisfy the LB, m, and calculate the infimum of the distance of all those function from f (in terms of the supremum norm) and denote it by $\gamma_f(m)$. Thus we can construct a generalized concept of LB in which a given function can be considered to have any $m \geq 0$ as LB, albeit up to a “Bound Deviation”, $\gamma_f(m)$, which is the price one pays for getting m as LB. We call this trade-off function, γ_f, the LB-BD function (or curve) of f. We develop methods for approximating functions using this framework and in particular find the optimal methods in this case. We also develop methods for calculating the LB-BD function from data. For the simulation and applications, we mainly focus on the 1-dimensional (1-d) domain case. However our framework can also be considered for the multidimensional domain case which deserves an extensive analysis that is beyond the scope of this paper.

In order to assess the performance of approximation methods, we need to define appropriate error measures. The definition of the error should depend on the specific application. Two important cases are as follows: (1) point-wise approximation; (2) integral approximation. In (1), the goal is to achieve good approximations to the function f at all points in the domain, while in (2) we are

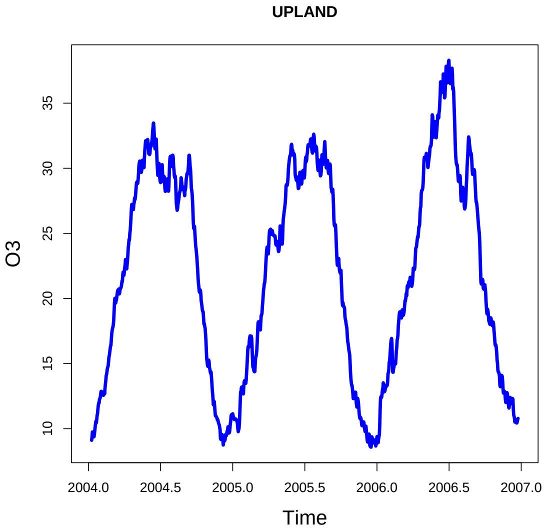

Fig. 1: Biweekly moving average of Ozone concentrations (O_3) plotted at a central station in Upland(UPL) in Southern California. We observe a general pattern which varies slowly over time and some small variations on top of that. The process is also approximately periodic at least when we focus on one given year (i.e. the beginning and end values approximately are the same).

interested in approximating the integral of the function f on the domain D. Therefore different errors should be considered accordingly. For example we can use $\sup_{x \in D} |f(x) - \hat{f}(x)|$ for the first case and $|\int_D f(x)dx - \int_D \hat{f}(x)dx|$ for the second case. In (2) the distinction between interpolation and approximation becomes less important and in fact, simple averaging of the available data (AVG), can be considered a reasonable method for that purpose and is often used in practice for example in the study of air pollution exposure assessment in Franklin et al. (2012). However even for the integral approximation case, we developed methods which outperform the simple averaging (and any other possible method) – a fact we show both by theory and simulations.

Since we assume the data are very sparse over time, the use of classical statistical methods such as regression may not be suitable. This is because with only few data points, it is either impossible to estimate the trends and the error (due to having too many parameters) or the estimates will be extremely poor, resulting in issues such as “over-fitting”. As an example consider an unknown function

$$f : [a, b] \to \mathbb{R},$$

for which only 3 points $x_1, x_2, x_3 \in [a, b]$ are observed:

$$f(x_1), f(x_2), f(x_3).$$

Clearly even a simple regression model of the kind

$$f(x) = a_0 + a_1 B_1(x) + a_2 B_2(x) + \epsilon(x),$$

where $B_i(x),\ i = 1,2$ are basis functions and $\epsilon(x) \sim N(0,\sigma^2)$, cannot be fitted because the number of parameters a_0, a_1, a_2, σ^2 is larger than the number of data points. Therefore one needs to make an assumption such as $\sigma^2 = 0$, in which case the fit will interpolate the points. The problem with such a fit is: in some cases there is no limit on how poorly the approximation performs outside the sampled points and we illustrate this point better in the simulations. Also note that even if we have access to more data, say 8 points, the fit would most likely remain very poor because: (1) the proposed model and in particular the basis functions could be inappropriate; (2) the parameter estimates can be extremely poor.

As another example consider the data consisting of a series of measurements of head accelerations (y-axis) versus *time* (x-axis) (Figure 2) in a simulated motorcycle accident, used to test crash helmets (see Silverman (1985)). These data are available as a part of the R package library, *MASS*. The full data set is depicted (black curve) and we have chosen a subset of 15 points (filled circles). The"locally weighted regression" (LOESS) fit (dotted), Cleveland et al. (1992), and smoothing spline fit (dashed) are also given and we observe that while these methods perform well at the beginning of the series, where more data are available, they fail dramatically at larger values (larger than 30), where less data are available. The LOESS and smoothing spline fits are fitted using R packages which estimate the parameters automatically with the standard available techniques, from which the predicted curves are created. In contrast, the thick curve is created using the *Lipfit* method developed in this paper, performs better by tracking the data closely. The problem with most of the classical curve fitting methods (regression/regularization) – when applied to the data which are sparse in some intervals – is there is nothing to prevent their fits from going well beyond the range of the data as shown in the above example. On the other hand $Lipfit$ is guaranteed to stay within the data range by definition. In general methods which produce fits which stay within the range of the data are desirable and can be useful in data sparse situations. We say a method is *data-range faithful* if it does not go beyond the available data range. More formally we give the following definition and later we will see that the $Lipfit$ method introduced in this paper satisfies this property.

Definition 1.1: Suppose the data $(x_i, y_i),\ i = 1, 2, \cdots, n$ are given, where x_i is the vector of predictors and y_i is the target variable. Also assume the goal is to fit the outcome (target) variable on D. Then let

$$y_{min} = \min\{y_i, i = 1, \cdots, n\},\ y_{max} = \max\{y_i, i = 1, \cdots, n\}.$$

The data-range is defined to be the interval $[y_{min}, y_{max}]$. A method is said to be data-range faithful, if the prediction from the method denoted by $\hat{y}$ satisfies:

$$\hat{y}(x) \in [y_{min}, y_{max}],\ x \in D.$$

Because classical regression and regularization techniques cannot be applied to such sparse data cases, alternative methods must be sought. For example we can consider these three simple methods: (1) take the average of $f(x_1), f(x_2), f(x_3)$ which we denote by AVG and approximate all unknown values $f(x)$ by this average; (2) To every point $x \in D$ find its *nearest neighbor* in $\{x_1, x_2, x_3\}$ and assign to $f(x)$ – a method we denote by NN; (3) for the 1-dimensional case, construct a curve by joining the available points and use that for approximation – a method we call "Linear Interpolation" (LI).

One might claim: at this point, this is the best one can hope for; there is not much difference between solutions (1), (2) and (3); and no further investigation is useful or necessary. However by a

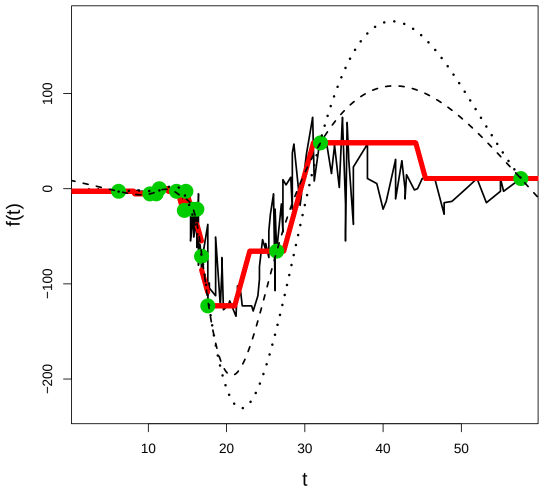

Fig. 2: Motorcycle head acceleration data. The data points (filled circles) are sampled from the full data (thin black curve). The LOESS (dotted) and the smoothing spline (dashed) fits are poor for values bigger than 30 (where the data are sparse) and goes way beyond the range of data points, while *Lipfit* (thick curve) still achieves a reasonable fit.

closer inspection, one notes that method (1) is indifferent to the times we have observed the data in the given period, while methods (2) and (3) utilize that information for possibly improving the interpolations. Moreover we can ask the following questions:

Question 1: Are there data-range faithful methods which perform well in approximating the function, especially in data-sparse situations?

Question 2: Is there a framework in which, using minimal assumptions, we can define what it means for a process to be slow-moving and we can find/compare the accuracy of various methods, such as methods (1), (2) and (3)?

Question 3: Which one of the methods (2) and (3) is better in terms of approximation error? Also are there any other methods that outperform (1), (2) and (3)?

Question 4: Given an approximation method, what is an optimal sampling scheme to get the smallest error in the approximation?

Question 5: In the 1-dimensional case, can we use the extra assumption of periodicity in the approximation and how does this affect the approximation error and optimal sampling scheme?

This paper answers all the above questions. In particular, it shows that under reasonable assumptions a better non-trivial solution exists and indeed is optimal in terms of the appropriate error. For periodic curves, we modify the methods to take advantage of this extra information and quantify exactly the gain in the accuracy and its effect on the sampling scheme.

Many of the definitions and results laid out in this paper are developed for the multidimensional domain case. However our focus, especially for the simulations and the application is on the 1-dimensional case. This is because the multidimensional case deserves an extensive investigation. Also one can get a good intuition from the 1-dimensional case in order to study the multidimensional domain case in which some of the results may be more difficult to prove or do not hold.

The "optimal central algorithm" for Lipschitz functions – which is a special case of $Lipfit$ when there is no deviation – was studied in Gaffney et al. (1976). General formulae in the multidimensional case are provided in Sukharev (1978). Beliakov (2006) developed a fast algorithm for computing central optimal interpolant. Hansen et al. (1992a) and Hansen et al. (1992b) study the problem of finding the maximum of a univariate Lipschitz function on a given domain.

To the best of our knowledge, the new contributions of this paper are the following: (1) We Compared the approximation methods in terms of several appropriately defined approximation (prediction) errors depending on the application, for example we make the contrast between approximating the function point-wise and approximating the integral of the function. (2) We rigorously introduced "data-informed" errors, which are essentially the approximation errors given both the position of the observations and their values. Also we explicitly calculate these errors for the 1-dimensional case and provide a method to calculate them for any dimension. These are useful in practice because we often can obtain smaller errors when considering the data-informed versions. These errors are also useful in theory. For example while the Linear Interpolation (LI) or Nearest Neighbor (NN) methods obtain the same error using the not data-informed errors, the data-informed errors are different. (3) Here we investigate the periodic functions and modify the results for that case. (4) We find optimal sampling schemes for obtaining the smallest possible approximation error for the 1-dimensional case. (5) We show that the LB-BD function enables us to generalize the results to many processes in practice and is a useful characteristic of the variation of processes over time and space. The only other work that considers extension of the interpolation of Lipschitz functions is Beliakov (2007), which considered random independent (gaussian and exponential) noise added to a Lipschitz function. However here our approach allows us to consider infinitely many Lipschitz Bounds up to their corresponding Bound Deviation (BD) and we do not require any assumption on the deviations (including its distribution or independence). In fact the key idea is in the trade-off between these two quantities. (6) For any LB-BD pair, we find the optimal approximating curve and calculate its exact approximation error. (7) We compare the approximation error of the methods we develop here with some well-known statistical smoothing methods in different scenarios of data availability and magnitude of BD. (8) We develop heuristic and exact methods for calculating the LB-BD function from data. The exact method uses convex optimization to find the LB-BD curve. (9) We find the approximation errors for a function with a (partially) given LB-BD curve. (10) We apply the methods developed here to air pollution (Ozone) data observed over time in Southern California.

The remainder of the paper is organized as follows. Section 2 includes a historical background of data fitting and interpolation methods and outlines their connection with this work. Section 3 develops a framework for approximating (fitting) slow-moving curves which are defined using bounds on the Lipschitz Bound. We also propose several loss functions to assess the goodness of approximation methods. We also discuss various approximation methods (some of which are interpolation methods) and find an optimal one $Lipfit$. We find optimal sampling points for getting the best possible approximation error. Section 4 extends the framework to slow-moving "wiggly" functions: function which do not have a small bound on Lipschitz Bound but can be approximated by such functions up to a deviation. Section 5 presents a method to generate functions with a given Lipschitz Bound and then compares the discussed approximation methods using simulations, showing that it does make a difference to choose an appropriate method for the problem at hand.

Section 6 discusses the trade-off between the LB and BD for a given function; defines the LB-BD function associated with a given function; and shows the nice properties of the LB-BD curve such as convexity. Section 7 discusses methods for calculating the LB-BD function, including a convex optimization method. Section 8 discusses finding appropriate parameters (LB-BD) for applying the methods and calculating approximation errors in practice, including a "Prediction Error Minimization Method" (PEM) and validation methods. Section 9 describes the application of the method in measuring Ozone exposure. Finally Section 10 discusses some remaining issues and extensions, for example extending this work to functions of multi-dimensional domains.

2 Background

Throughout this paper we use the words function "approximation", "fitting" and "prediction" interchangeably and to refer to any method which inputs data and outputs values for the function at unknown points. However it is useful to clarify what is usually meant by interpolation here and in the relevant literature because several of the methods discussed in this work are interpolation methods. Interpolation refers to any method which inputs n values of a *target function* f defined on $D \subset \mathbb{R}^d$: $(x_1, f(x_1)), \cdots, (x_n, f(x_n))$ and outputs a function $\hat{f}$ on D which *agrees* with f on the given points (and it is supposed to be close to f values outside the given points). This is in contrast to classical fitting methods in statistics (e.g. linear regression) for which often the estimated curve does not go through the given points. In order to include all possible methods, we use the term *approximation* to refer to any method that given the input data returns a function $\hat{f}$ on D. A very simple example of approximation – which is not an interpolation method – is a method we denote by AVG and simply takes the average of the available values of f and assign that to all the domain: $\hat{f}(x) = \sum_{i=1}^n f(x_i)/n$. We do not think the distinction between interpolation and general approximation is really useful since any approximation method can be slightly tweaked to become an interpolation method by redefining the value of the approximation at the available points to be the same as the data. Moreover there are usually infinitely many "out-of-sample" points as compared to finitely many "in-sample" points. Therefore in practice, we often care mostly about the out-of-sample performance anyway.

Since we consider some interpolation methods in this paper, here we discuss some historical background on this topic. Interpolation of points surprisingly goes back to astronomy in ancient Babylon and Greece when it was all about time keeping and predicting astronomical events (Meijering (2002)). Later Newton and Lagrange studied the problem of interpolating a function f defined on an interval $[a, b]$ with given values on n points: $x_1, \cdots, x_n$, by a polynomial of degree n and arrive at the same solution (with different computational method). The Lagrange method gives this polynomial, p, by defining $l_i(x) = \prod_{j=1; j \neq i}^n \frac{x - x_i}{x_j - x_i}$ and letting

$$p_n(x) = \sum_{i=1}^n l_i(x) f(x_i).$$

Moreover it can be shown (see Cheney and Kincaid (2008)) that if f is $(n+1)$ times differentiable, then for some $a < \zeta < b$:

$$f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\zeta) \prod_{i=1}^n (x - x_i). \tag{1}$$

Therefore if a bound is available on the magnitude of $f^{(n+1)}$, i.e. $|f^{(n+1)}| \leq M$, denoting the interpolation error by $e(x) := |f(x) - p_n(x)|$ we have:

$$e(x) \leq \frac{1}{(n+1)!} M \prod_{i=1}^{n} |x - x_i|.$$

Unfortunately both the existence of $f^{(n+1)}$ and having a small bound are rare in practice.

While a practitioner may have an idea about the first derivative magnitude (or another measure of variation such as Lipschitz Bound) of the process under the study, it is extremely rare to know something about the $(n+1)$th derivative of the process or even believe it exists! To illustrate this point consider the bounded function $f(x) = (1 + x^2)^{-1}$ for which derivatives of all orders are available and suppose n equally spaced points are available for interpolation. One can show

$$\lim_{n \to \infty} \max_{x \in [-5,5]} |f(x) - p_n(x)| = \infty,$$

(Cheney and Kincaid (2008)). This means as more data become available, the accuracy of the interpolation gets worse! The reason the Newton/Lagrange polynomial method fails dramatically in this case is the high-order derivatives of this simple bounded function become very large for some $x \in [-5, 5]$ (or else Equation 1 would guarantee a precise bound). Apparently it was expected that a (continuous) function f will be well-approximated by interpolating polynomials and "in the history of numerical mathematics, a severe shock occurred when it was realized that this expectation was ill-founded" (Cheney and Kincaid (2008)). Based on the discussion above, we seek methods which make the least possible assumptions regarding the properties of function f. In fact we do not require existence of any derivatives and only require the function to have a Lipschitz Bound, which is a weaker assumption than that of the existence of the first derivative. Also we derive the approximation errors merely based on this bound. In later sections, we relax the existence of a (small) Lipschitz Bound by allowing the function to be well-approximated by a function with relatively small Lipschitz Bound up to a deviation defined appropriately.

Interpolation is also considered in (medical) image processing and a good summary is given in Lehmann et al. (1999) and Thévenaz et al. (2000). In that application, it is often assumed to have access to the values of a 2-dimensional image on a equally spaced rectangular grid, which is not the case we consider in this paper.

Another view of approximating functions emerged with the least squares method of Gauss which was followed by various other regression methods. The main difference of this framework (from the interpolation methods previously developed) is to view the function as a combination of a true underlying function and some added "noise". For example a version of linear regression assumes

$$f(x) = \beta_0 + \beta_1 x + \epsilon,$$

where ϵ is independent and identically distributed noise process, for example normally distributed: $\epsilon \sim N(0, \sigma^2)$. This can be generalized in many ways to include more predictors or non-linear trends. For example

$$f(x) = \beta_0 + \beta_1 B_1(x) + \beta_2 B_2(x) + \epsilon,$$

for some basis functions $B_1(x), B_2(x)$. The function can now be seen as an imperfect observation of a true function and the main objective is to infer about the true function. For example if we predict a value at an observed $x = x_i$ for which f is observed to be $f(x_i)$, using the historical interpolation methods, we get back the same observed value $f(x_i)$, while from the regression, we may get back a completely different value, which is supposed to be closer to the noise-free true value.

A more recent view of fitting functions emerged as the so-called regularization methods (Hastie at al. (2009)). As an example, consider the "smoothing splines" method which finds the minimizer of

$$\underset{f}{argmin} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \int_D ||f''(x)||dx, \tag{2}$$

where $f''(x)$ is a second derivative and $\lambda \geq 0$ is a "penalty" term, which creates a trade-off between the deviation of the fitting (approximation) function f and the variation of the function. If we do not include the second term, $\lambda \int_D ||f''(x)||dx$, we end up with a function that necessarily interpolates them and can have chaotic behavior outside the observed points – a similar problem to that of interpolation methods of Legendre and Newton. An appropriate λ is usually chosen by "cross-validation". Solving Equation 2 can be shown to be equivalent to

$$\underset{f}{argmin} \sum_{i=1}^{n} |f(x_i) - y_i|^2 \tag{3}$$

$$\text{subject to } \int_D ||f''(x)||dx \leq \lambda^\star, \tag{4}$$

for some $\lambda^\star \geq 0$, which is determined by λ. In fact the second representation comes from the dual problem of the first one in convex optimization theory.

Conceptually the regularization methods, such as smoothing splines do not explicitly assume and model a noise process but rather do not allow the function to vary too much (thus prevent "over-fitting") through $\int_D ||f''(x)||dx \leq \lambda^\star$. The framework we use in this paper is similar in this sense and does that by controlling the variation through the Lipschitz Bound and the deviation by $\max_{i=1}^n |f(x_i) - y_i|$. Of course the solution in the two cases can be dramatically different. One thing the latter achieves is the fit always remains within the range of the data through controlling the Lipschitz Bound and the severe deviation measure of $\max_{i=1}^n |f(x_i) - y_i|$ which does not let the approximation to deviate from the data much at any individual point. With this background comparison it becomes clear that other solutions to the fitting problem can be considered by choosing various variation measures and deviation measures – each of which may suit different applications. We leave a thorough study of these various choices to future work and discuss more about this idea in a general framework in the discussion section.

The reason for the better performance of $Lipfit$ method over smoothing splines in data-sparse situations can also be seen through the variation-deviation framework. The smoothing spline method only controls the second derivative by penalizing the integral of its magnitude. For example the penalty is zero for a curve with large slope and if the data is sparse. Hence is an opportunity for the fit to deviate a lot from the true values as shown in Figure 2. We observe in that figure for larger values where data are sparse, the fit travels up with high slope but has plenty of opportunity to come back down to the last data point with rather small second derivative (curvature). In this case the $Lipfit$ method, does not allow this chaotic behavior by controlling the LB.

An important idea that we introduce in this work is: for any function f, we consider the trade-off between the variation measure and the deviation measure, which is summarized in the LB-BD function, denoted by γ_f. As we discussed in the introduction, for any LB=m, we can always find a BD=$\gamma_f(m)$ for which a function g exists so that g has LB m and deviates from f (only) as much as $\gamma_f(m)$. This is in contrast to a single penalty term usually considered in other regularization methods such as smoothing splines where this trade-off concept is also considered as referred to as "bias-variance" trade-off (Hastie at al. (2009)). However to best of our knowledge the properties of this trade-off function is not explored closely, as we have done in this work for the LB-BD trade-off.

3 A framework for approximating slow-moving functions

Making inference about an arbitrary function with sparse data is not feasible if we do not have any extra information at our disposal. Here we show that one such useful assumption is having a relatively small *Lipschitz Bound* which is defined formally below. Here we consider real functions defined on a subset D of the d-dimensional Euclidean space, $\mathbb{R}^d$:

$$f : D \subset \mathbb{R}^d \to \mathbb{R},$$

and denote the set of all such functions by $\mathbb{R}^D$. We also consider the supremum norm on this space

$$\|f\|_\infty = \sup_{x\in D}|f(x)|,$$

which induces a metric (and topology) on $\mathbb{R}^D$. We denote the Euclidean norm for a vector v in $\mathbb{R}^d$ by $||v||$. In this paper we mainly restrict ourselves to the case where $d = 1$ and work with functions defined on closed intervals. While many of the definitions and results laid out in this paper are developed for the multidimensional domain case, we leave a more comprehensive study of the general case to future work. Also one can get a good intuition from the 1-dimensional case in order to study the higher dimensional (spatial) case in which some of the results are more difficult to prove or do not hold.

Definition 3.1: (i) Suppose $f : D \subset \mathbb{R}^d \to \mathbb{R}$ is a function. Then f is said to have a Lipschitz Bound (LB), m, if $|f(x) - f(y)| \leq m||x - y||,\ x, y \in D$. We denote the set of all such functions by $\mathcal{LB}(D, m)$ or $\mathcal{LB}(m)$ when the domain is clear from the context.
(ii) We denote the set of all *periodic* functions on $[a, b]$ ($f(a) = f(b)$) and with Lipschitz Bound m by $\mathcal{PLB}([a, b], m)$ or $\mathcal{PLB}(m)$ if the domain is clear from the context.
(iii) Infimum Lipschitz Bound:

$$Lip(f) = \inf\{m \in \mathbb{R},\ \ |f(x) - f(y)| \leq m||x - y||,\ \ \forall x, y \in D\}.$$

(iv) Denote the set of all at least one-time differentiable functions (for multi-dimensional case when the gradient exists) on D by $\mathcal{DIF}$. (Assume also D is an open set). Then let

$$\mathcal{DIF}(D, m) := \mathcal{DIF} \cap \mathcal{LB}(D, m),\ \mathcal{PDIF}([a, b], m) := \mathcal{DIF} \cap \mathcal{PLB}([a, b], m).$$

Lemma 3.1: (Properties of Lipschitz Bound)
(i) Suppose $f : D \subset \mathbb{R}^d \to \mathbb{R}$ is differentiable for an open subset D and $|f'(x)| \leq m,\ \forall x \in D$ then

$$Lip(f) =: \sup_{x\in D}||f'(x)||.$$

In other words $f \in \mathcal{DIF}$ then $f \in \mathcal{LB}(m = \sup_{x \in D} ||f'(x)||)$.
(ii) Suppose f is continuous on a convex set $D \subset \mathbb{R}^d$. Moreover assume $D = \cup_{i=1}^{n} D_i$, for a collection of convex subsets D_i and f has Lipschitz Bound m on each of the D_i. Then f is Lipschitz Bound m on D.
(iii) $\mathcal{LB}(D, m)$ is a convex and closed subspace of $\mathbb{R}^D$.
(iv) $\mathcal{DIF}(D, m)$ is convex but not closed in $\mathbb{R}^D$ and is therefore a strict subset of $\mathcal{LB}(D, m)$.
(v) $f \in \mathcal{LB}(D, Lip(f))$.
(vi) $Lip : \mathbb{R}^D \to\ R$ is a convex function (but not continuous).
(vii) $\mathcal{DIF}(D, m)$ is dense in $\mathcal{LB}[D, m]$ and $\mathcal{PDIF}([a, b], m)$ is dense in $\mathcal{PLB}[(a, b), m]$.

Proof See Appendix. ■

3.1 Slow-moving pattern generation

This subsection presents a method for generating slow-moving patterns over time (1-dimensional case). In particular, we generate patterns with a given Lipschitz Bound. We present this method for both periodic and non-periodic functions. We start by defining classes which satisfy a given LB and it is clear from the definition of the classes how such functions can be simulated.

Definition 3.2: Suppose an interval $[a, b]$ and a LB, m, are given. Consider a collection of k points $a < p_1 < p_2 < \cdots < p_k < b$, a collection of slopes $m_1, \cdots, m_{k+1}$ so that $|m_i| \leq m$, and an intercept y_0. Then define a function $f : [a, b] \to \mathbb{R}$ by first defining $f(a) = y_0$ and drawing a line segment starting from $(a, f(a))$ with slope m_1 until $(p_1, f(p_1))$. Then draw another line segment starting from $(p_1, f(p_1))$ with slope m_2 until the point $(p_2, f(p_2))$ and continue in the same manner for $m_3, \cdots, m_{k+1}$. Then f is a piecewise linear function on $[a, b]$ with LB m. Denote the class of all such functions by $\mathcal{PL}([a, b], m)$.

Definition 3.3: Suppose an interval $[a, b]$ and a Lipschitz Bound m are given. Consider a collection of k points $a < p_1 < p_2 < \cdots < p_k < b$, a collection of slopes $m_1, \cdots, m_{k-1}$ so that $|m_i| \leq m$, and an intercept y_0. Then define a function $f : [a, b] \to \mathbb{R}$ by first defining $f(p_1) = y_0$ and drawing a line segment starting from $(p_1, f(p_1))$ with slope m_1 until $(p_2, f(p_2))$. Then draw another line segment starting from $(p_2, f(p_2))$ with slope m_2 until the point $(p_3, f(p_3))$ and continue in the same manner for $m_3, \cdots, m_{k-1}$ to get to the point $(p_k, f(p_k))$. Then to assure periodicity, connect this point to $(b + (p_1 - a), f(p_1))$ and calculate the slope of this last line and call it m_k. If $|m_k| \leq m$, f is a periodic piecewise linear function on $[a, b]$ with LB m. Denote the class of all such functions by $\mathcal{PPL}([a, b], m)$.

In the following theorem we show that the function spaces $\mathcal{PL}([a, b], m)$ and $\mathcal{PPL}([a, b], m)$ are good approximations of $\mathcal{LB}([a, b], m)$ and $\mathcal{PLB}([a, b], m)$.

Theorem 3.1: $\mathcal{PL}([a, b], m)$ is dense in $\mathcal{LB}([a, b], m)$ and $\mathcal{PPL}([a, b], m)$ is dense in $\mathcal{PLB}([a, b], m)$ in terms of sup norm: $||f||_\infty = \sup\limits_{x \in D} ||f(x)||$.

Proof Suppose $f \in \mathcal{PL}([a, b], m)$ then for any $\epsilon > 0$ consider a grid $a = a_0 < a_2 < \cdots < a_n = b$ such that $a_{i+1} - a_i \leq \epsilon/(2m),\ i = 0, \cdots, (n-1)$. Define $\tilde{f}$ to be the linear interpolation of f on the grid:

$$\tilde{f} = LI[f, (a_1, \cdots, a_n), (f(a_1), \cdots, f(a_n))].$$

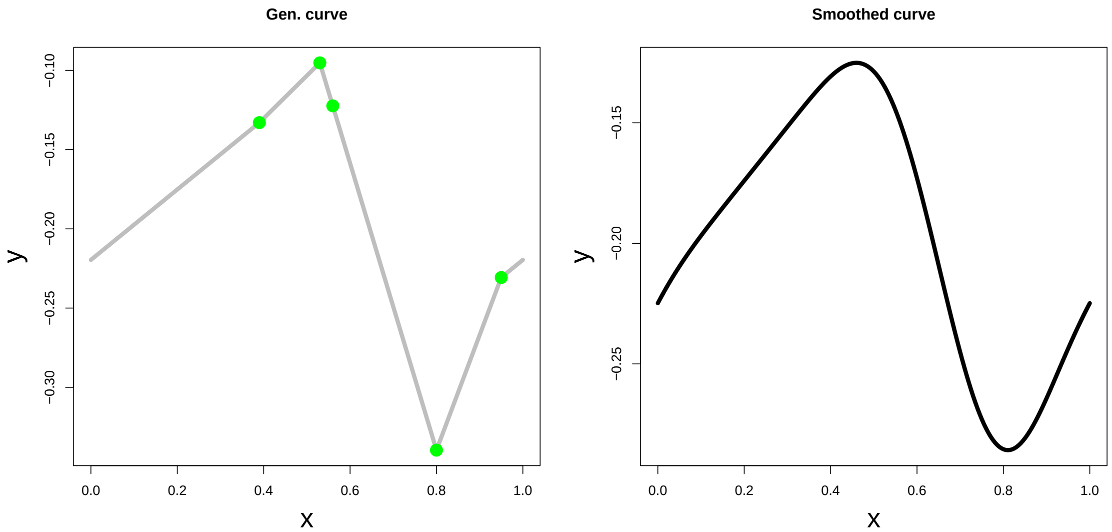

Fig. 3: Left Panel: A simulated periodic function with 5 break points and with LB equal to 1. Right Panel: A smoothed version of the simulated curve using a moving average filter.

Then for $x \in [a, b]$, suppose a_j is the closest element to x on the grid and note:

$$\begin{aligned}
|\tilde{f}(x) - f(x)| &= |\tilde{f}(x) - f(a_j) + f(a_j) - f(x)| \\
&= |\tilde{f}(x) - \tilde{f}(a_j) + f(a_j) - f(x)| \\
&\leq |\tilde{f}(x) - \tilde{f}(a_j)| + |f(a_j) - f(x)| \\
&\leq m|x - a_j| + m|x - x_j| \\
&\leq (2m)\epsilon/(2m) = \epsilon,
\end{aligned}$$

and this proves $||f - \tilde{f}||_\infty \leq \epsilon$.
The same proof works for $\mathcal{PPL}$ by noting $\tilde{f}(x)$ will be periodic if f is periodic. ■

We can modify the above methods to get more smooth functions at the break points by using a filtering method (moving average). An example of the periodic case with 5 break points $p_1, \cdots, p_5$ is given in Figure 3 along with its more smooth version. In order to do several simulations we need to define random procedures to find the break points and the slopes. Later we use uniform distributions for both cases but we also make sure the break points are not too close as discussed later.

3.2 Loss functions

We can consider various loss functions to asses the efficiency of the approximation methods. We use the absolute value of the difference to calculate a distance between functions values at a given point, while other measures of distance such as square of the difference can be considered depending on the application. As we discussed in the introduction, we can consider two general type of losses: losses for approximating the integral of a function; and losses for approximating the point-wise values of the function. Below we introduce various loss functions for such purposes.

- The integral approximation loss:

$$IL(f,\hat{f}) := |\int_D f(x)dx - \int_D \hat{f}(x)dx|.$$

  In fact this loss only depends on $\hat{f}$ through $\int_D \hat{f}(x)dx$. More precisely

$$\int_D g_1(x)dx = \int_D g_2(x)dx \Rightarrow IL(f,g_1) = IL(f,g_2).$$

- The point-wise approximation loss, for which two measures can be considered:

  1. Supremum point-wise loss:

$$SPWL(f,\hat{f}) := \sup_{x\in D} |f(x) - \hat{f}(x)|,$$

  2. Mean point-wise loss:

$$MPWL(f,\hat{f}) := \int_D |f(x) - \hat{f}(x)|/v(D),$$

where $v(D) = \int_D 1dx$. For example $v([a,b]) = b - a$. The error measures defined above are not scale-free and cannot inform us how much of the variation of the function is captured using an approximation method. In order to standardize the above error, we can divide them by the "diameter" of f, which we define to be

$$diam(f) := \sup_D(f) - \inf_D(f).$$

Then we define the standardized supremum point-wise approximation error to be

$$SSPWL(f,\hat{f}) = SPWL(f,\hat{f})/diam(f).$$

It is easy to see that if the approximation method is "scale-free":

$$g = a + bf \Rightarrow \hat{g} = a + b\hat{f},$$

then $SSPWL$ is also scale-free:

$$SSPWL(g,\hat{g}) = SSPWL(f,\hat{f}).$$

Any reasonable approximation method and the ones discussed here are scale-free. Also note that $0 \leq SSPWL \leq 1$.

Suppose we have a family of functions with LB, m, and defined on D. We define the family-standardized approximation error to be

$$FSPWL(f,\hat{f}) = SPWL(f,\hat{f})/v(D)m.$$

Again it is easy to see that if the approximation method is scale-free then $FSPWL$ is also scale-free. It is clear that $0 \leq FSPWL \leq SSPWL \leq 1$ in general. Similarly we define standardized versions of IL, $MPWL$ and denote them by SIL, $SMPWL$. We denote their family-standardized versions by FIL, $FMPWL$.

3.3 Approximations and their performance

Suppose the value of $f : D \subset \mathbb{R}^d \to \mathbb{R}$ is observed at n points $\mathbf{x} := (x_1, \cdots, x_n)$, where each x_i is a column vector of length d; it takes values $\mathbf{y} := (y_1, \cdots, y_n)$; a LB m is available; and we are interested in approximating f at unobserved points x in D. We denote such an approximation by $approx[\mathbf{x}, \mathbf{y}]$, which is a function on the same domain as f. An "approximation method", $approx[.,.]$ is formally a function that inputs data and outputs functions:

$$\begin{aligned} approx : \cup_{n=1}^{n} \mathbb{R}^{n\times d} \times \mathbb{R}^n &\to \mathbb{R}^D, \\ (\mathbf{x}, \mathbf{y}) &\mapsto approx[\mathbf{x}, \mathbf{y}], \end{aligned}$$

where n is the size of the data set. In the previous section, we introduced loss measures for assessing the distance of a given curve to the target function. This cannot directly be used to judge the performance of an approximation method, because the true function is not available in practice. However $SPWL$ and IL introduced previously are useful in simulations where the true curve is known.

In the following definition, we introduce approximation (prediction) error measures which are suitable for comparing approximation methods when the target function is not available. The definition follows by introducing an "ordering" on the set of all approximations and we discuss the connection of this ordering to the error measures.

Definition 3.4: Suppose we are interested in approximating a function $f : D \subset \mathbb{R}^d \to \mathbb{R}$, which belongs to a family of functions $\mathcal{F}$. For example $\mathcal{F} = \mathcal{LB}(m)$ or $\mathcal{F} = \mathcal{PLB}(m)$. Also assume f is observed on $\mathbf{x} = (x_1, \cdots, x_n)$ and takes values $\mathbf{y} = (f(x_1), \cdots, f(x_n))$.
(i) We define the "data-informed supremum point-wise error" to be:

$$DSPWE(approx, \mathbf{x}, \mathbf{y}) = \sup_{f\in\mathcal{F}, f(\mathbf{x})=\mathbf{y}} SPWL(f, approx[\mathbf{x}, \mathbf{y}]).$$

(ii) We define the "supremum point-wise error" to be:

$$SPWE(approx, \mathbf{x}) = \sup_{f\in\mathcal{F}} SPWL(f, approx[\mathbf{x}, \mathbf{f}(\mathbf{x})]).$$

(iii) We define the "data-informed mean point-wise error" to be:

$$DMPWE(approx, \mathbf{x}, \mathbf{y}) = \sup_{f\in\mathcal{F}, f(\mathbf{x})=\mathbf{y}} MPWL(f, approx[\mathbf{x}, \mathbf{y}]).$$

(iv) We define the "mean point-wise error" to be:

$$MPWE(approx, \mathbf{x}) = \sup_{f\in\mathcal{F}} MPWL(f, approx[\mathbf{x}, \mathbf{f}(\mathbf{x})]).$$

(v) We define the "data-informed integral error" to be:

$$DIE(approx, \mathbf{x}, \mathbf{y}) = \sup_{f\in\mathcal{F}, f(\mathbf{x})=\mathbf{y}} IL(f, approx[\mathbf{x}, \mathbf{y}]).$$

(vi) We define the "integral error" to be:

$$IE(approx, \mathbf{x}) = \sup_{f\in\mathcal{F}} IL(f, approx[\mathbf{x}, \mathbf{f}(\mathbf{x})]).$$

(vii) The family-standardized version of the above errors can be obtained by dividing them by $v(D)m$ and we denote them by adding "F" to the title, for example $FDSPWE$ is the family-standardized version for $DSPWE$ and so on.

In the sequel, we obtain both $DSPWE$ and $SPWE$ for well-known approximation methods and the optimal one we develop here. The difference between $DSPWE$ and $SPWE$ is that: $DSPWE$ utilizes the extra information about the actual values of the function at the observed values to calculate the approximation error. While $SPWE$ only uses the position of the points for which f is observed: $(x_1, \cdots, x_n)$ and the approximation error is obtained without using the actual values of the curve $(f(x_1), \cdots, f(x_n))$. Therefore $SPWE$ is useful in the sampling phase – when we decide where to observe the values of a function – in which case we do not have access to the values of the target function a priori. However when we do have access to the values of the function, we should not discard that information in assessing the error in the approximation and that is what $DSPWE$ achieves. To our knowledge most of these types of approximation error measures have not been considered rigorously in the literature while $SPWE$ has been considered for the "Linear Interpolation" method e.g. in Cheney and Kincaid (2008) and for Lipschitz functions e.g. in Beliakov (2006). We show in the sequel that considering $DSPWE$ not only gives us a better error measure, it does make a difference in comparing various methods. In fact, we show that the extra information about the data can guide us to choose among methods which yield the same $SPWE$ but differ in terms of $DSPWE$.

Point-wise error function

The error measures defined above are useful to assess the goodness of various loss functions on their domains. Since these are defined using the whole domain, we can consider them as "overall" measures of error. As we will see various cases of approximation methods have the same overall error in some situations. However, we can compare these approximation methods point-wise by considering the "point-wise error function":

$$pef[approx, \mathbf{x}, \mathbf{y}](x) = \sup_{f \in \mathcal{F}, f(\mathbf{x})=\mathbf{y}} |f(x) - approx[\mathbf{x}, \mathbf{y}](x)|.$$

Note that pef is a function on D in contrast to $DSPWE$ and in fact

$$DSPWE[approx, \mathbf{x}, \mathbf{y}] = \sup_{x \in D} pef[approx, \mathbf{x}, \mathbf{y}](x);$$

$$DMPWE[approx, \mathbf{x}, \mathbf{y}] = \int_D pef[approx, \mathbf{x}, \mathbf{y}](x) dx / v(D).$$

Ordering of approximations

When comparing two approximation methods $approx_1, approx_2$, to show the superiority of $approx_1$ to $approx_2$ (in terms of point-wise error), one ideally wants to show a superiority everywhere on the domain:

$$pef[approx_1, \mathbf{x}, \mathbf{y}](x) \leq pef[approx_2, \mathbf{x}, \mathbf{y}](x), \ \forall x \in D, \tag{5}$$

from which we can conclude:

$$DSPWE[approx, \mathbf{x}, \mathbf{y}] \leq DSPWE[approx_2, \mathbf{x}, \mathbf{y}];$$

$$DMPWE[approx, \mathbf{x}, \mathbf{y}] \leq DMPWE[approx_2, \mathbf{x}, \mathbf{y}].$$

We denote the relation in Equation 5 by $approx_1 \preceq_{pw} approx_2$. If both $approx_1 \preceq_{pw} approx_2$ and $approx_2 \preceq_{pw} approx_1$ hold, we write $approx_1 =_{pw} approx_2$. Also note that this relation is "transitive" i.e.

$$approx_1 \preceq_{pw} approx_2, \text{ and } approx_2 \preceq_{pw} approx_3 \Rightarrow approx_1 \preceq_{pw} approx_3.$$

Also it is "antisymmetric":

$$approx_1 \preceq_{pw} approx_2, \; approx_2 \preceq_{pw} approx_1 \Rightarrow approx_1 =_{pw} approx_2.$$

However it is not a "total ordering" in general. And there may exist a pair $approx_1, approx_2$ for which neither $approx_1 \preceq_{pw} approx_2$ nor $approx_2 \preceq_{pw} approx_1$ is true. In contrast, if we define an ordering using $DSPWE$ or DIE, then clearly we get a total ordering (since the usual ordering of real numbers is a total ordering). Interestingly, this is one of those rare situations that although the ordering is not a total ordering, there is a solution to the approximation problem which minimizes the approximation error in terms of $\preceq_{pw}$.

In the following, we show that for important approximation methods, it is possible to order them using $\preceq_{pw}$ (which immediately gives the ordering for $DSPWE$ and $DMPWE$). Moreover we find a method which is superior to all the methods using $\preceq_{pw}$ and it is the unique method with this property.

In the case of integral losses, we have a similar contrast between DIE and IE as before: DIE is a better measure of error, while IE is useful when the values of the function are not available. However, for approximating integrals, there is no point-wise error function version similar to pef, since by definition we are interested in integrals. Below we formally define various approximation methods.

Important examples of the (1-d) approximation methods are:

1. **Average (AVG)**: $AVG[f](x) := \frac{1}{n}\sum_{i=1}^{n} f(x_i)$. In other words, this method simply calculates the average of the value of f at the available n points and assigns that to all points. As noted before this is not generally an interpolation method. However this method is widely used in approximating the integral of a curve.

2. **Nearest Neighbor (NN)**: to every point, assigns the value of f at the closest available point.

3. **Periodic Nearest Neighbor (PNN)**: This is a variation of NN to use the assumed periodicity in f. We add two points to $x_1, \cdots, x_n$: $x_n^* = x_n - (b-a)$ and $x_1^* = x_1 + (b-a)$. Note that by the periodicity assumption $f(x_1^*) = f(x_1)$ and $f(x_n^*) = f(x_n)$. Then we apply the NN method to the points

$$(x_n^*, f(x_n^*)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)), (x_1^*, f(x_1^*)).$$

   To our knowledge this method has not been considered before.

4. **Linear Interpolation (LI)**: This method draws line segments between each pair of points $(x_i, f(x_i)), (x_{i+1}, f(x_{i+1})),\ i = 1, \cdots, (n-1)$ and approximates $t \in [x_i, x_{i+1}]$ by the corresponding value on the line. For points $[a, x_1]$ and $[x_n, b]$, we assign the nearest neighbor value.

5. **Periodic Linear Interpolation (PLI)**: is a variation of LI to use the assumed periodicity in f. We add two points to $x_1, \cdots, x_n$: $x_n^* = x_n - (b-a)$ and $x_1^* = x_1 + (b-a)$ and use the LI method. To our knowledge this method has not been considered before.

6. **Regression:** For example, using r basis functions to fit a function to the available points. The number of basis functions can be chosen in a way that the fit interpolate the points. For example in the 1-d case, for n points, we can use up to n basis functions. The basis functions for the non-periodic case can be considered to be $1, t, t^2, \cdots, t^r$. If we consider a periodic function then periodic basis functions such as Fourier series can be considered. When the monomials $1, t, \cdots, t^{n-1}$ are considered, we will get the Newton/Legendre polynomial as the solution (since that is the unique polynomial which interpolates the n points).

7. **Regularization:** Examples of regularization methods are smoothing splines and LOESS. The $Lipfit$ method developed in this work can also be considered as a regularization method as we discuss later.

Remark. The above approximation methods, except for the last two, are data-range faithful, i.e. the approximated function is in the range of the data. While, regression methods and the regularization methods in general are not data-range faithful, $Lipfit$ which can also be viewed as a regularization method is data-range faithful.

Remark. All the above approximation methods can be immediately extended to multidimensional input space, except for LI and the periodic cases.

We start by the simplest case where only one point is observed at $t_0 \in [a, b]$ and we are interested in approximating f on each point or its integral.

Lemma 3.2: (Approximation using one point) Suppose $f : [a, b] \to \mathbb{R}$, $f \in \mathcal{LB}(m)$ and the value of f is available at x_1, $f(x_1) = y_1$. Then the constant function $approx[f](x) = y_1$ is uniquely the best approximation to f at any $x \in [a, b]$ in terms of pef and therefore optimal using $DSPWE$ and $DMPWE$ (see Figure 4). It also minimizes the DIE. Moreover we have:
(i) $pef[x_1, y_1](x) = |f(x) - approx[x_1, y_1](x)| = |f(x) - f(x_1)| \le m|x - x_1|$;
(ii) $SPWE[approx, x_1] = DSPWE[approx, x_1, y_1] = \max\{m(x_1 - a), m(b - x_1)\}$;
(iii) $IE[approx, x_1] = DIE[approx, x_1, y_1] = \frac{m}{2}((x_1 - a)^2 + (x_1 - b)^2)$.

Proof See Figure 4 for the idea and the Appendix for a proof. ■

Then we move to the case for which two points are available. Below we describe a method (denoted by $Lipfit$) which we prove to be optimal in terms of pef ordering and therefore in terms of $DSPWE$ and $DMPWE$. This method was also considered and proved to be optimal in Sukharev (1978) and Beliakov (2006). The optimal method can also be found for the multidimensional domains and is presented in Beliakov (2006). Our contribution in this paper about the optimal solution are: finding closed-form solutions for the approximation errors for the optimal solution and other classical methods such as NN and LI; simulations for comparing the methods; extending the framework and finding the optimal solution for the wiggly functions case (the ones for which the Lipschitz Bound is too big to be useful); finding closed-form errors for the optimal solution and the closed-form prediction error for the optimal solution and other standard methods in the wiggly case.

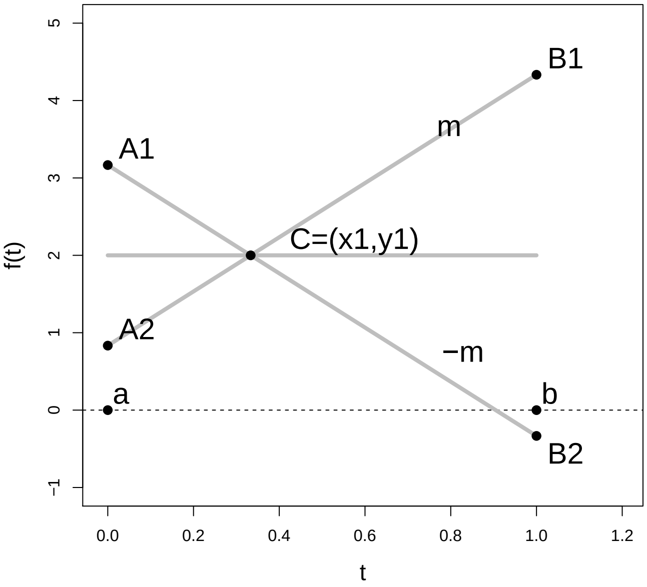

Fig. 4: One point interpolation using only one observed point (x_1, y_1) denoted by C in the Figure. The constant function is given by solid line. Two broken lines on $[a, b]$ with slopes $(m, -m)$ and going through C are also shown in the Figure. The function on $[a, b]$ with trajectory A_1C, CB_1 takes the highest possible values and the function with trajectory A_2C, CB_2 takes the lowest.

Lipfit **Method (1-d) case:** Suppose $f : [a, b] \to$ has Lipschitz Bound m and the value of f is available at $x_A < x_B,\ f(x_A) = y_A,\ f(x_B) = y_B$. Let $A = (x_A, y_A)$ and $B = (x_B, y_B)$ to be points in the cartesian plane as shown in Figure 5. Draw two lines starting from each of A and B with slopes equal to $m, -m$. Each of the two lines starting from A is parallel with one of the lines starting from B and will meet the other lines. Call the intersection points C, D with C denoting the top point and D the bottom point. One of the C or D will be closer to A and the other to B (depending on the sign of slope of AB). In Figure 5 the point closer to A is C and the point closer to B is D (since sign of slope of AB is positive). Define the point F to be $F = (x_F, y_F) := (x_D, y_A)$ and $G = (x_G, y_G) = (x_C, y_B)$. Define $Lipfit[f]$ to have the trace AF, FG, GB.

Remark. Note that in this case $Lipfit$ indeed interpolates the points and we will prove that it is optimal. We will also see that the optimal method for “wiggly” curves (which we will still call $Lipfit$) also interpolates the observed points but can have sudden discontinuity at the (and only at the) observed points. Even in that case strictly speaking the method is still an interpolation. However it is different from usual interpolation methods which are continuous on the domain.

Remark. The $Lipfit$ method is data-range faithful, i.e. the approximated curve is in the range of the data.

$Lipfit$ **general (multidimensional) case:** Suppose a function f is given at $\mathbf{x} = (x_1, \cdots, x_n)$ where each x_i is a column vector of length d denoting a point in $D \subset \mathbb{R}^d$, with values equal to $\mathbf{y} = (y_1, \cdots, y_n)$. Then suppose we are interested in approximating f at a point $x \in D$. Applying the Lipschitz Bound to x and x_i, for $i = 1, \cdots, n$, we get

$$|f(x) - f(x_i)| \leq m||x - x_i|| \Rightarrow f(x_i) - m||x - x_i|| \leq f(x) \leq f(x_i) + m||x - x_i||,$$

from which we conclude

$$H^{lower}(x) \leq f(x) \leq H^{upper}(x),$$

where,

$$H^{lower}(x) = \max_{i=1,\cdots,n}(f(x_i) - m||x - x_i||),$$
$$H^{upper}(x) = \min_{i=1,\cdots,n}(f(x_i) + m||x - x_i||).$$

Then optimal solution which minimizes $|f(x) - \hat{f}(x)|$ is given by

$$\hat{f}(x) = (H^{lower}(x) + H^{upper}(x))/2.$$

In order to see that it is sufficient to note that (1) both $H^{lower}(x), H^{upper}(x)$ interpolate the data; (2) satisfy the Lipschitz Bound of m; (3) it is impossible for $f(x)$ to be outside the range $[H^{lower}(x), H^{upper}(x)]$. The pef can also be expressed in terms of $H^{lower}(x), H^{upper}(x)$:

$$pef(x) = (H^{upper}(x) - H^{lower}(x))/2.$$

For more details see Beliakov (2006), which also provides a fast computational method for obtaining the solution, since calculating $\hat{f}(x)$ point wise is not computationally plausible for many values of x. Also note that our solution in the 1-d case matches this solution and is computationally fast. For the 1-d case here we present a simple proof of optimality using essentially the same steps but with a geometric argument in Lemma 3.3 and Theorem 3.2.

Lemma 3.3: Suppose $f : [a, b] \rightarrow$ has Lipschitz Bound, m, and the value of f is available at $x_A < x_B$, $f(x_A) = y_A$, $f(x_B) = y_B$. Then $Lipfit$ uniquely minimizes the pef when approximating the function f on $[x_A, x_B]$ and therefore minimizes $DSPWE$. It also minimizes the integral approximation error: DIE.

Proof See the Appendix. ■

Remark. $Lipfit, NN, LI$, all assign $(y_B + y_A)(x_B - x_A)/2$ to the integral of f on $[x_A, x_B]$. Therefore they all achieve the same integral error (DIE).

The method $Lipfit$ above was introduced for two points. We can extend that to n points: $x_1, \cdots, x_n$ by:
(i) considering the intervals $[a, x_1), [x_1, x_2), \cdots, [x_{n-1}, x_n), [x_n, b]$;
(ii) assigning constant values $f(x_1)$ to $[a, x_1]$ and $f(x_n)$ to $[x_n, b]$;
(iii) applying the $Lipfit$ for two points to each of the remaining intervals.

In the above theorem, we showed the optimality of $Lipfit$ for two points and this immediately generalize to n points.

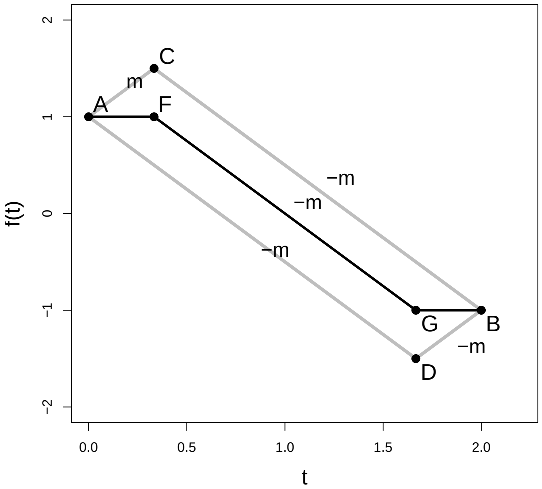

Fig. 5: Definition of the $Lipfit$ for 1-d case. We assume the target function goes through A and B. We draw two lines with slopes $(m, -m)$ from each of A and B to form the parallelogram $ACBD$. The optimal solution is given by the function with trajectory AF, FG, GB.

Theorem 3.2: (Optimal Approximation Theorem) $Lipfit$ is uniquely optimal when n points are available in terms of $\preceq_{pw}$ and therefore $DSPWE$ and $DMPWE$.

Proof Any $x \in [a, b]$ belongs to one of the intervals

$$[a, x_1), [x_1, x_2), \cdots, [x_{n-1}, x_n), [x_n, b],$$

and by definition $Lipfit$ is optimal in all intervals. Therefore using Lemma 3.3 and Lemma 3.2, we are done. ■

For each of the LI and NN methods, we introduced periodic versions: PLI and PNN respectively and the same can be done for the $Lipfit$ method which we denote by $PLipfit$. Again it is true that $PLipfit$ is uniquely optimal when n points are available for a periodic function in terms of $\preceq_{pw}$ and therefore in terms of $DSPWE$ as well as $DMPWE$.

Optimal methods to approximate integrals:

For the error in approximating the integrals – as we saw for the one point case – the constant function is optimal and for the two points and each of $NN, LI, Lipfit$ give rise to the same approximation. This approximation is done by taking the average of the two available values, $(y_B + y_A)/2$, and multiplying that by the length of the interval $(x_B - x_A)$. We can define an equivalence relation on the set of the curve approximations for functions on $[a, b]$ by

$$approx_1 \sim approx_2 \iff \int_a^b approx_1[\mathbf{x}, \mathbf{y}](t)dt = \int_a^b approx_2[\mathbf{x}, \mathbf{y}](t)dt.$$

Then we have $NN \sim LI \sim Lipfit$ and we denote their common equivalence class by $WAVG$ to stand for "weighted averaging" method which we describe below for n points:

$WAVG$ **Method:**
(i) consider the intervals $[a, x_1), [x_1, x_2), \cdots, [x_{n-1}, x_n), [x_n, b]$;
(ii) let $s_0 = y_1(x_1 - a)$ and $s_n = y_n(b - x_n)$;
(iii) let $s_i = (y_{i+1} + y_i)(x_{i+1} - x_i)/2$;
(iv) return $s = \sum_{i=0}^{n} s_i$ as the approximation for the integral.

The periodic version is given by:

$PWAVG$ **Method:**
(i) let $x_{n+1} = b + (x_1 - a)$ and consider the intervals $[x_1, x_2), \cdots, [x_{n-1}, x_n), [x_n, x_{n+1}]$;
(ii) let $s_i = (y_i + y_{i+1})(x_{i+1} - x_i)/2,\ i = 1, \cdots, n$;
(iii) return $s = \sum_{i=1}^{n} s_i$ as the approximation for the integral.

We have the following result regarding the optimality of $WAVG$ and $PWAVG$.

Theorem 3.3: (Integral Approximation Optimal Method)
(i) $WAVG$ is the unique optimal integral approximation method.
(ii) $PWAVG$ is the unique optimal integral approximation method for periodic functions.

Proof See Figure 6 for the idea and the Appendix for a proof. ■

Remark. The integral methods can be applied to the multidimensional domain case as well by calculating the integral of the $Lipfit$ method for the multidimensional case.

3.4 The approximation error for various methods

This subsection gives the approximation error for the 1-d case of the various methods introduced in this work in closed form. We start by the approximation error for the integral approximation and then move to the point-wise case.

Theorem 3.4: Suppose a function with Lipschitz Bound smaller than m is considered and the function trajectory goes through points $A = (x_A, y_A)$ and $B = (x_B, y_B)$. Denote the slope of the line from A to B by $m^\star$. Then the (sharp) data-informed integral error for $WAVG$ on $[x_A, x_B]$ is given by:

$$DIE[WAVG, (x_A, x_B), (y_A, y_B)] = \frac{m^2 - m^{\star 2}}{4m}(x_B - x_A)^2.$$

Proof See Appendix. ■

Corollary 3.1: The (sharp) integral error IE for $WAVG$ on $[x_A, x_B]$ is given by:

$$IE[WAVG, (x_A, x_B), (y_A, y_B)] = \frac{m}{4}(x_B - x_A)^2.$$

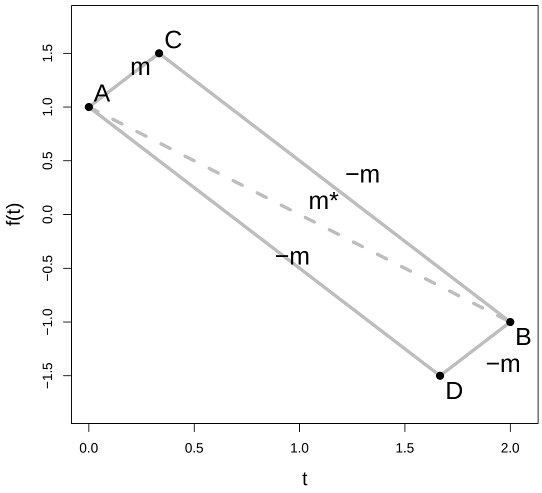

Fig. 6: This figure shows the derivation of the optimal integral approximation method and its error. The target function goes through A, B. The LI method's approximation has exactly an integral between (1) the function f_1 with trajectory along AC, CB, which is the function with the largest possible integral; and (2) f_2 with trajectory along AD, DB, which is the function with the smallest possible integral. Therefore LI and any other method in its equivalence class minimize the integral error.

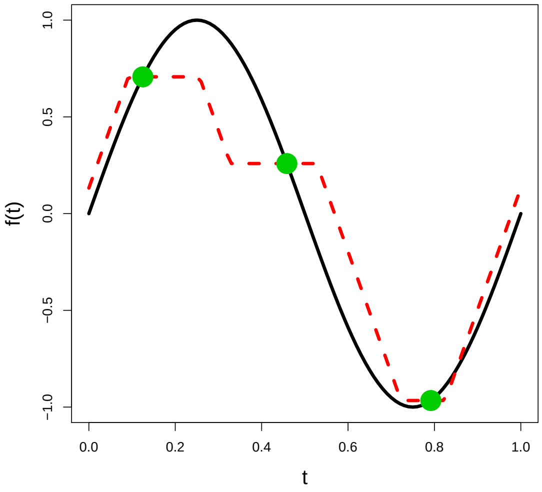

Fig. 7: Example for the $Lipfit$ Method with three points and for the periodic case.

Proof

$$DIE[WAVG, (x_A, x_B), (y_A, y_B)] = \frac{m^2 - m^{\star 2}}{4m}(x_B - x_A)^2,$$

is maximized by letting $m^\star = 0$. Figure 8 depicts the worst case. ■

The following theorem finds the errors for the aforementioned methods used for point-wise approximation.

Theorem 3.5: Suppose a function with Lipschitz Bound, m, is considered and the function trajectory goes through points $A = (x_A, y_A)$ and $B = (x_B, y_B)$. Denote the slope of the line from A to B by $m^\star$. Also define $\Delta_x = (x_B - x_A)$, $\Delta_y = (y_B - y_A)$, and $\Delta = (\Delta_x - |\Delta y/m|)/2$. Then the (sharp) data-informed supremum point-wise error ($DSPWE$) for various methods to interpolate the curve on $[x_A, x_B]$ are as follows:

(a) NN: $DSPWE[NN, (x_A, x_B), (y_A, y_B)] = |m\frac{\Delta_x}{2}|$.

(b) LI: $DSPWE[LI, (x_A, x_B), (y_A, y_B)] = \Delta(m + |m^\star|)$.

(c) $Lipfit$: $DSPWE[Lipfit, (x_A, x_B), (y_A, y_B)] = \Delta m$.

Moreover $Lipfit \preceq_{pw} LI \preceq_{pw} NN$. Also $Lipfit =_{pw} LI$ if and only if $m = m^\star$ or $m^\star = 0$ and $LI =_{pw} NN$ if and only if $m^\star = 0$.

Proof See Appendix. ■

Remark. Note that Δ is half of the difference between: (1) the variation in the x axis Δ_x; and (2) the variation in the y-axis Δ_y divided by the largest possible slope the curve can obtain (m). Therefore $|\Delta_y/m| \leq \Delta_x$ and $\Delta \geq 0$. Also obviously $\Delta \leq \Delta_x/2$ and therefore the NN method is inferior to $Lipfit$ in terms of $DSPWE$.

Remark. The result above extends to n points: $x_1, \cdots, x_n$ by considering the subintervals $[a, x_1), [x_1, x_2), \cdots, [x_{n-1}, x_n), [x_n, b]$ and taking the maximum error over all the subintervals.

Corollary 3.2: Suppose a function with Lipschitz Bound m is given. Also suppose we have access to the functions values on x_A and x_B and use that to interpolate the function on the interval $[x_A, x_B]$. Let $\Delta_x = (x_B - x_A)$. Then the (sharp) point-wise error ($SPWE$) for various methods to interpolate the curve on $[x_A, x_B]$ are as follows:

$$SPWE[Lipfit] = SPWE[LI] = SPWE[NN] = |m\frac{\Delta_x}{2}|.$$

Remark. The worst case for all cases happens in the same situation where $y_A = y_B$ and f travels with a slope of magnitude m from A to B and the slope sign changes in the mid point (Figure 8).

Corollary 3.2 implies that if we allow the function with Lipschitz Bound m on $[x_A, x_B]$ to vary and take the supremum over all the $DSPWE$ errors, then all of the methods mentioned above

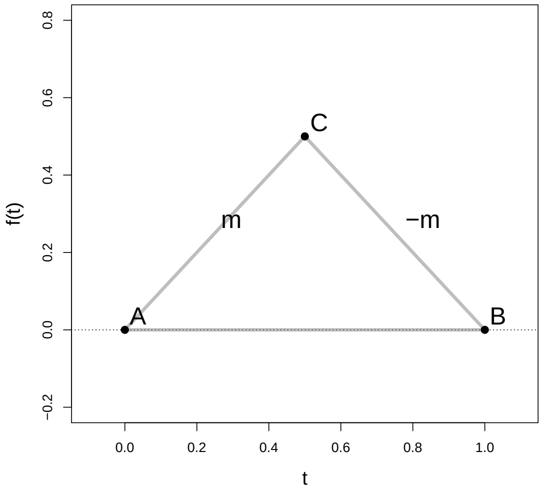

Fig. 8: Worst case error for both point-wise and integral error happens when $y_B = y_A$ and the curve moves with maximum slope magnitude (m or $-m$) until midpoint C and with the negative of the previous slope ($-m$ or m) to go back to B.

achieve the same overall error $SPWE$. This is also true for a collection of n observed points. In the following, we find the error in predicting a periodic or non-periodic function on an interval when n points are observed.

Theorem 3.6: (Approximation errors on an interval)
Suppose $f : [a, b] \to \mathbb{R}$ has Lipschitz Bound m and the value of f is available at $x_1 < \cdots < x_n,\ f(x_i) = y_i$. Also suppose *approx* is any of the methods $NN, LI, Lipfit$. Define $e_i := \frac{x_{i+1}-x_i}{2}; i = 1, \cdots, n-1$. Also let $e_0 := |a - x_1|, e_n := |b - x_n|$ and $e_{max} = \max\{e_0, \cdots, e_n\}$ (Figure 9). We have

$$SPWE[approx, \mathbf{x}] = me_{max}.$$

Proof See Appendix. ■

Theorem 3.7: (Approximation errors on an interval: periodic case)
Suppose $f : [a, b] \to \mathbb{R}$ is periodic, $f \in \mathcal{LB}(m)$ and the value of f is available at $x_1 < \cdots < x_n,\ f(x_i) = y_i$. Also suppose *approx* is any of the methods $PNN, PLI, PLipfit$. Then let $e_i := \frac{x_{i+1}-x_i}{2}; i = 1, \cdots, n-1,\ e_0 := |a - x_1|, e_n =: |b - x_n|$ and $e_{0,n} := \frac{e_0+e_n}{2}$ (Figure 10). Let $e_{max} = \max\{e_{0,n}, e_1, \cdots, e_{n-1}\}$, then we have:

$$SPWE[approx, \mathbf{x}] = me_{max}.$$

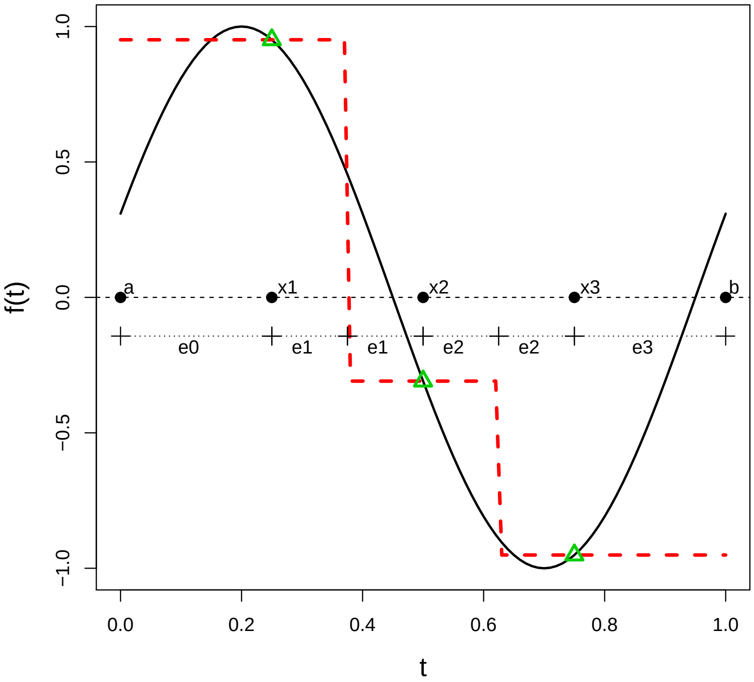

Fig. 9: Figure illustrates the calculation of the NN error bound on an interval. The observed points at (x_1, x_2, x_3) are denoted by triangles.

Proof See Appendix. ■

3.5 Optimal sampling

This subsection finds the optimal sampling times for both the non-periodic and periodic cases for each of the discussed methods.

Theorem 3.6 implies that the optimal sampling times for a function using any of the methods $(NN, LI, Lipfit)$ are the same. Similarly Theorem 3.7 implies the same for the periodic case.

Theorem 3.8: (Optimal sampling)
Suppose $f : [a, b] \to \mathbb{R}$, $f \in \mathcal{LB}(m)$ and the value of f is available at $x_1 < \cdots < x_n$, $f(x_i) = y_i$. Also suppose we use one of $NN, LI, Lipfit$ methods for approximation. Let $e_i := \frac{x_{i+1}-x_i}{2}; i = 1, \cdots, n-1$, $e_0 := |a - x_1|, e_n := |b - x_n|$. The approximation bound is minimized be letting $e_0 = e_n = e_i = \frac{(b-a)}{2(n+1)}$. This implies $x_1 = a + (b - a)/2(n + 1)$, $x_{i+1} = x_i + (b - a)/(n + 1)$, $i = 1, 2, \cdots, n - 1$.

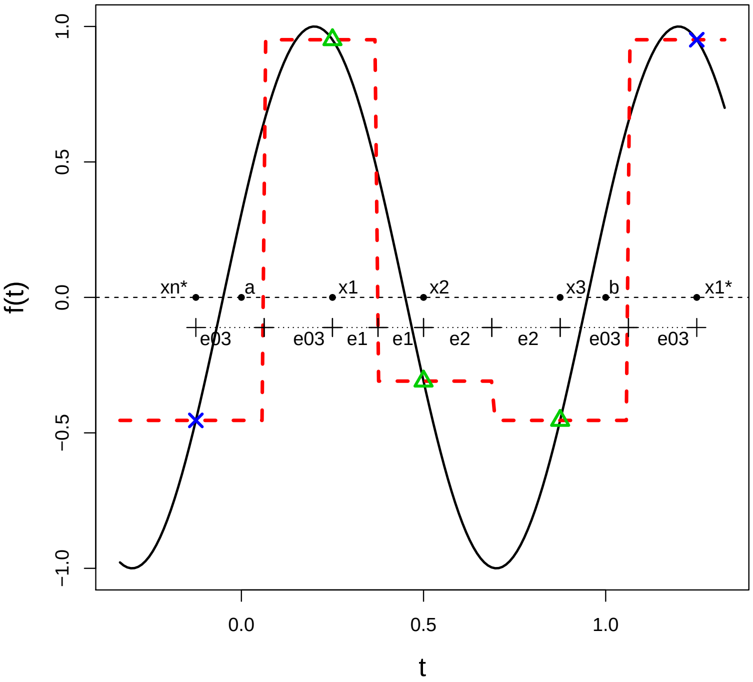

Fig. 10: Figure illustrates the calculation of the PNN error on an interval. The observed points at (x_1, x_2, x_3) are denoted by triangles and the crosses are the points added by the periodicity assumption to apply the PNN method.

Proof Note that the maximum error in this case is $\max\{e_0, \cdots, e_n\}$ and moreover $e_0 + e_n + 2\sum_{i=1}^{n-1} e_i = (b-a)$. Therefore we conclude $e_0 = e_n = e_i = \frac{(b-a)}{2(n+1)}$ minimizes the error. ■

Theorem 3.9: (Optimal sampling: periodic case)
Suppose $f : [a, b] \to \mathbb{R}$, $f \in \mathcal{PLB}(m)$ and the value of f is available at $x_1 < \cdots < x_n$, $f(x_i) = y_i$. Also suppose we use one of $PNN, PLI, PLipfit$ methods for interpolating f. Let $e_i := \frac{x_{i+1}-x_i}{2}; i = 1, \cdots, n-1$, $e_0 := |a - x_1|, e_n := |b - x_n|$ and $e_{0,n} := \frac{e_0+e_n}{2}$. Then the approximation error is minimized by letting $e_{0,n} = e_i = (b-a)/(2n)$, $i = 1, \cdots, n-1$.

Proof This is because $e_{0,n} + \sum_{i=1}^{n} e_i = (b-a)/2$, $i = 1, \cdots, n-1$. ■

4 Extension to wiggly functions

Many processes do not posses a reasonably small Lipschitz Bound and therefore the approximation errors achieved by the above theory are very big. This is actually true even in the case of the temporal Ozone process shown in Figure 1. However in this figure, we observe that despite the high variation in the small scale, a slow-moving pattern is present in a larger scale. This section provides a method to extend the theory developed before to this case. The idea for this extension lies in the fact that in such cases the process can be well-approximated by a function with a reasonably small Lipschitz Bound. To formally introduce this idea, we start by the following definitions.

Definition 4.1: Suppose $f : D \subset \mathbb{R}^d \to \mathbb{R}$. Then f is said to have Lipschitz Bound, m, up to a “Bound Deviation” (BD), σ, if there exist a function g such that $|f(x) - g(x)| \leq \sigma,\ \forall x \in D$ and g has Lipschitz bound m. The class of all such functions is denoted by $\mathcal{ALB}(D, m, \sigma)$ or by $\mathcal{ALB}(m, \sigma)$ when the domain D is clear from the context.

Remark. Note that $f \in \mathcal{ALB}(m, \sigma)$ does not even need to be continuous. Therefore we have generalized this method to functions which are not continuous but well-approximated (in the sense that σ is small) by a continuous function g.

Similar to the simple case with no deviation, we can define a periodic family for the case with deviations and we denote that by $\mathcal{PALB}([a, b], m, \sigma)$.

Definition 4.2: Suppose $f : [a, b] \to \mathbb{R}$. Then f is said to have periodic Lipschitz Bound, m, up to a “Bound Deviation” (BD), σ, if there exist a function g such that $|f(x) - g(x)| \leq \sigma,\ x \in [a, b]$ and g is periodic with Lipschitz Bound, m. The class of all such functions is denoted by $\mathcal{PALB}([a, b], m, \sigma)$ or by $\mathcal{PALB}(m, \sigma)$ when the domain $[a, b]$ is clear from the context.

Remark. Note that this definition lets us include functions which are “approximately periodic”. In other words, it allows a function f for which $0 \leq |f(b) - f(a)| \leq \sigma$.

Just in the same way that along with $\mathcal{LB}(D, m)$ we defined the function families $\mathcal{DIF}(D, m)$ and $\mathcal{PL}([a, b], m)$; along with $\mathcal{ALB}([a, b], m, \sigma)$, we define the function families $\mathcal{ADIF}(D, m, \sigma)$ and $\mathcal{APL}([a, b], m, \sigma)$ to be the version with BD.

The following theorem shows the convexity of the set $\mathcal{ALB}(D, m, \sigma)$.

Theorem 4.1: $\mathcal{ALB}(D, m, \sigma)$ is a convex set.

Proof Let $f_1, f_2 \in \mathcal{ALB}(D, m, \sigma)$ then there exist $g_1, g_2 \in \mathcal{LB}([a, b], m)$ such that $SPWL(f_i, g_i) \leq \sigma,\ i = 1, 2$. Then we need to show that any function of the form $f = \theta f_1 + (1 - \theta) f_2,$ for a $\theta \in (0, 1)$ is in $\mathcal{ALB}(D, m, \sigma)$ as well. Let $g = \theta f_1 + (1 - \theta) f_2$ and note that $g \in \mathcal{LB}(D, m)$ by convexity of $\mathcal{LB}(D, m)$. Also note that

$$\forall x \in D,\ |f(x) - g(x)| \leq \theta |f_1(x) - g_1(x)| + (1 - \theta)|f_2(x) - g_2(x)| \leq \sigma,$$

which proves $SPWL(f, g) \leq \sigma$ and we are done. ■

The extension of the results obtained before for $\mathcal{LB}(m)$, $\mathcal{PLB}(m)$ to the $\mathcal{ALB}(m, \sigma)$, $\mathcal{PALB}(m)$ families is not straight-forward for all the methods and deviations. This is because: while for $\mathcal{LB}$, $\mathcal{PLB}$ families it was impossible to have points A, B observed on the trace of the target function so that the slope of AB is strictly greater than m, it is possible in the $\mathcal{ALB}$, $\mathcal{PALB}$ case due to the existence of the BD. We start first by extending the approximation methods we discussed before to the case with deviations. The approximation for the one point case is trivial and all methods still assign the constant value available. Therefore we only discuss the two point case from which the general multiple point case can be obtained by considering appropriately defined subintervals as before.

Approximation methods for wiggly functions (1-d):

- $NN, PNN, LI, PLI, WAVG, PWAVG$ and their periodic version extend with no change.
- $Lipfit$ method:
  If $|m^\star| \leq m$ then the method is defined as before (Figure 11, top left panel). While if $|m^\star| > m$ as shown in Figure 11 (top right and bottom left panels), then we define the approximating curve as follows.
  Define $\Delta' = \Delta_x(|m^\star| - m)/2$ and the points
  $$F = (x_A, y_A + sign(m^\star))\Delta';$$
  $$G = (x_B, y_B - sign(m^\star))\Delta'.$$
  Then the $Lipfit$ method is given by the line segment FG.

Remark. Note that the value of BD is not needed for applying the $Lipfit$ method.
Remark. The $Lipfit$ method is data-range faithful, i.e. the approximated curve is in the range of the data.
Remark. The $Lipfit$ method is an interpolation method in the sense that on the observed points returns the observed values. However it may have sudden discontinuity at the observed values.

The $Lipfit$ method can also be generalized to multidimensional case easily as we discuss below.

Lipfit **general (multidimensional) case for wiggly functions:** Suppose a function f is given at $\mathbf{x} = (x_1, \cdots, x_n)$ where each x_i is a column vector of length d denoting a point in $D \subset \mathbb{R}^d$, with values equal to $\mathbf{y} = (y_1, \cdots, y_n)$. Then suppose we are interested to approximate f at a point $x \in D$. Applying the Lipschitz Bound to x and x_i for $i = 1, \cdots, n$, we get

$$|f(x) - f(x_i)| \leq m||x - x_i|| + \sigma \Rightarrow f(x_i) - m||x - x_i|| - \sigma \leq f(x) \leq f(x_i) + m||x - x_i|| + \sigma,$$

from which we conclude

$$H^{lower}(x) \leq f(x) \leq H^{upper}(x),$$

where

$$H^{lower}(x) = \max_{i=1,\cdots,n} (f(x_i) - m||x - x_i||) + \sigma(1 - 1_{\{x_1,\cdots,x_n\}}(x)),$$
$$H^{upper}(x) = \min_{i=1,\cdots,n} (f(x_i) + m||x - x_i||) - \sigma(1 - 1_{\{x_1,\cdots,x_n\}}(x)),$$

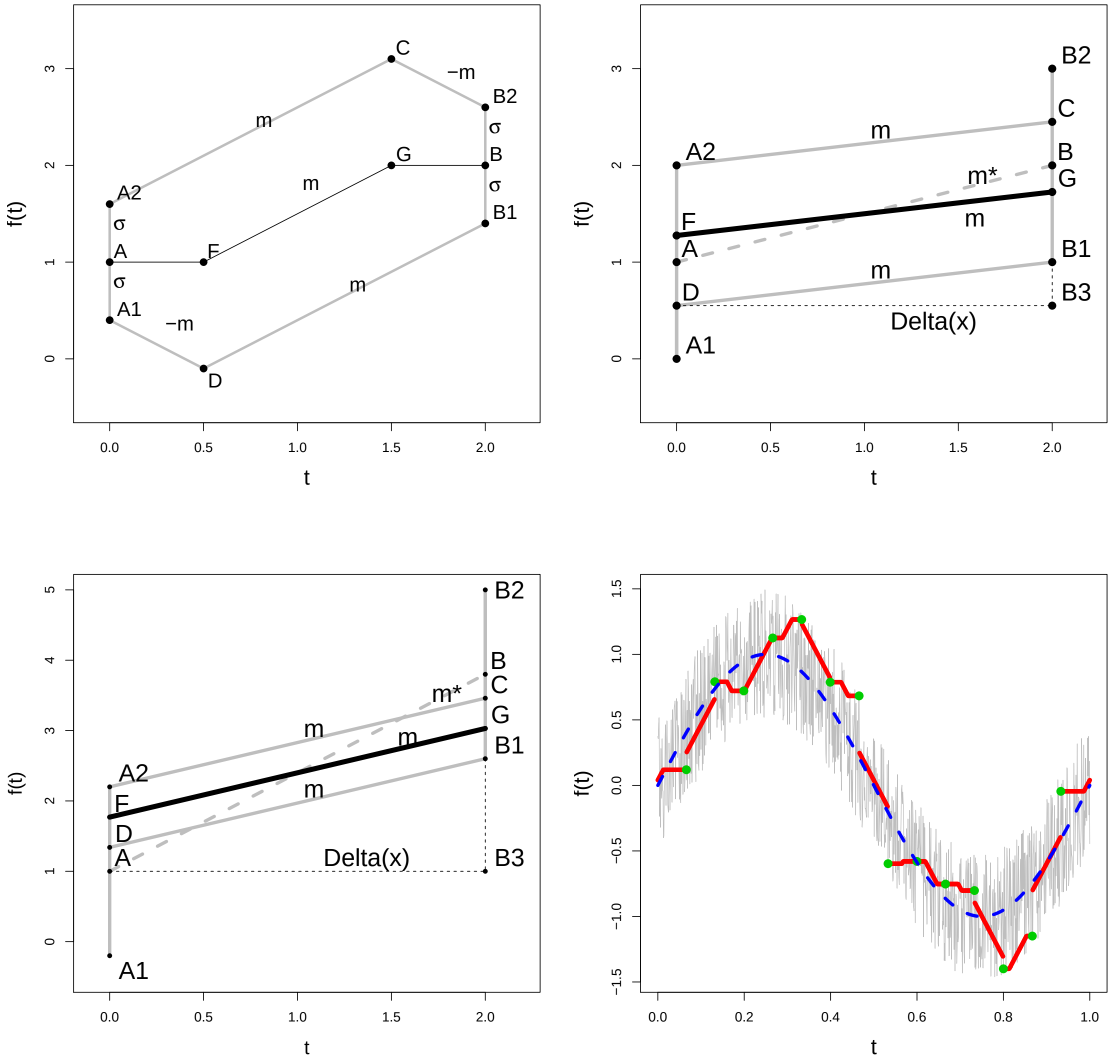

Fig. 11: Top Left Panel: $Lipfit$ method with deviation, Case 1 ($|m^\star| \leq m$). Top Right Panel: $Lipfit$ method with deviation, Case 2 ($|m^\star| > m$). Bottom Left Panel: $Lipfit$ method with deviation, Case 3 (also $|m^\star| > m$). Bottom Right Panel: The curve (grey) is generated by adding uniform noise from the interval $[-\sigma, \sigma]$ to a curve with given LB. The data (filled circles) are fitted with $Lipfit$ method (thick curve).

where $1_{\{x_1,\cdots,x_n\}}(x) = 1$ if x is an observed point and zero otherwise. Then optimal solution which minimizes $|f(x) - \hat{f}(x)|$ is given by

$$\hat{f}(x) = (H^{lower}(x) + H^{upper}(x))/2.$$

Note that σ cancels out and we see that for the general solution it also does not appear in the solution (as it was the case for the 1-d case we discussed before). In order to see the optimality, it is sufficient to note that: (1) both $H^{lower}(x), H^{upper}(x)$ interpolate the data; (2) they belong to $\mathcal{ALB}(m,\sigma)$; (3) it is impossible for $f(x)$ to be outside the range $[H^{lower}(x), H^{upper}(x)]$. Thus we have extended the results in Sukharev (1978) and Beliakov (2006). Also note that our solution in 1-d case match this solution and is computationally fast. The point-wise error function can be expressed in terms of $H^{lower}(x), H^{upper}(x)$):

$$pef(x) = (H^{lower}(x) - H^{upper}(x))/2,$$

from which $DSPWE$ and DIE can be calculated for the multidimensional case.

The prediction errors for the (1-d) case given in Theorems 3.5 and 3.4 extend as follows.

Theorem 4.2: Suppose a function with Lipschitz Bound, m, is considered and the function trajectory goes through points $A = (x_A, y_A)$ and $B = (x_B, y_B)$. Denote the slope of the line from A to B by $m^\star$ and let $\Delta_x = (x_B - x_A)$. Then the data-informed integral error for $WAVG$ on $[x_A, x_B]$ is given by:
(i) If $|m^\star| \leq m$:

$$DIE[WAVG, (x_A, x_B), (y_A, y_B)] = \frac{m^2 - m^{\star 2}}{4m}(\Delta_x)^2 + \sigma\Delta_x.$$

(ii) If $|m^\star| > m$, define $\Delta' = \Delta_x(|m^\star| - m)/2$:

$$DIE[WAVG, (x_A, x_B), (y_A, y_B)] = (\sigma - \Delta')\Delta_x.$$

Proof See proof of Theorem 4.3. ■

Theorem 4.3: Suppose a function belongs to $\mathcal{ALB}(m,\sigma)$ with trajectory going through points $A = (x_A, y_A)$ and $B = (x_B, y_B)$. Denote the slope of the line from A to B by $m^\star$. Also define $\Delta_x = (x_B - x_A)$, $\Delta_y = (y_B - y_A)$, and $\Delta = (\Delta_x - |\Delta_y/m|)/2$. Then the data-informed point-wise error ($DSPWE$) for various methods to interpolate the curve on $[x_A, x_B]$ are given below.

(i) If $|m^\star| \leq m$:

- $DSPWE[NN, (x_A, x_B), (y_A, y_B)] = |m\frac{\Delta_x}{2}| + \sigma.$
- $DSPWE[LI, (x_A, x_B), (y_A, y_B)] = \Delta(m + |m^\star|) + \sigma.$
- $DSPWE[Lipfit, (x_A, x_B), (y_A, y_B)] = \Delta m + \sigma.$

(ii) If $|m^\star| > m$, define $\Delta' = \Delta_x(|m^\star| - m)/2$:

- $DSPWE[NN, (x_A, x_B), (y_A, y_B)] = |m\frac{\Delta_x}{2}| + \sigma.$

- $DSPWE[LI, (x_A, x_B), (y_A, y_B)] = \sigma$.
- $DSPWE[Lipfit, (x_A, x_B), (y_A, y_B)] = \sigma - \Delta'$.

Proof See Appendix. ■

Theorem 4.4: (Optimal Approximation Theorem for wiggly functions)
Suppose $f : [a, b] \to \mathbb{R}, f \in \mathcal{ALB}(m, \sigma)$ and the value of f is available at $x_A < x_B$, $f(x_A) = y_A$, $f(x_B) = y_B$. Then $Lipfit$ method uniquely minimizes pef when approximating the function f on $[x_A, x_B]$ and therefore it minimizes $DSPWE, DMPWE$ and DIE.

Proof It follows from the proof of Theorem 4.3. ■

Optimal sampling (1-d): Corollary 3.2 as well as Theorems 3.6 and 3.7 also extend to the wiggly case by simply adding an σ to the corresponding errors. Therefore the optimal sampling schemes remain the same as the case without error.

5 Simulation studies

This section uses simulations to investigate approximating functions in the framework developed in this work for the 1-dimensional domain case. Subsection 3.1 discussed methods to generate functions with given Lipschitz Bound for both non-periodic and periodic cases. These methods are used here for simulating appropriate functions to compare approximation methods. First we compare the methods for deviation-free case and then move to the wiggly function case. In the wiggly case we consider the simulations for functions that are generated with a given Lipschitz Bound and a deviation which is generated from uniform distribution. This is a special case, because in general the deviations can also have some remaining patterns. However even in this special case, we show that the performance of different approximation methods depend on the magnitude of the deviation and the data sparsity structure. Some remaining work in this area include the multidimensional domain case and the case with more complex deviations and we leave that for future work.

5.1 Comparison of the methods

Here we compare these methods to interpolate curves with 3 points of data available between [0,1]: (1) Average of 3 points: AVG; (2) Nearest neighbor: NN; (3) Periodic nearest neighbor: PNN; (4) Linear Interpolation: LI; (5) Periodic Linear Interpolation: PLI; (6) $Lipfit$; (7) Periodic $Lipfit$: $PLipfit$; (8) Regression with 1 basis function $(\cos(2\pi t))$; (9) Regression with 2 Basis functions $(\cos(2\pi t), \sin(2\pi t))$.

For the simulations, we generate periodic curves with 5 break points with Lipschitz Bound of 1 for functions defined on [0,1]. We also use filtering (moving average) to make sure the change of the derivative is at most 1 at the break points. (Note that the change in the derivative can be 2 if we do not do any filtering because the Lipschitz Bound is 1.) The 3 data points are taken from the interval [0,1] uniformly subject to the condition that: when we construct a circle of circumstance 1 from the interval by joining the end points 0 and 1, the distance of every pair of points is at least $1/(3+1)$. Note that by the optimal sampling result, Theorem 3.9, the distance for the best sampling approach is equal to 1/3.

The result of the fits for one simulation are given in Figure 12. We repeat the simulations 10^5 times and calculate the family-standardized loss,

$$FSPWL = \sup_{t\in[a,b]} |f(t) - \hat{f}(t)|/m(b-a).$$

Various quantiles for the prediction errors are given in Table 1 in which we observe: there is a major gain in using the periodic version of the methods; for higher values of the quantiles of the error, the regression using basis functions perform very poorly; for higher values of the quantiles, the best method is $PLipfit$ and PLI is performing very closely with PNN coming third. We have repeated the analysis with 3,4,6,7 break points and the results were consistent with this case.

Tab. 1: Comparing methods to fit 10^5 periodic curves generated from 5 break points with Lipschitz Bound of 1. The 3 data points are taken from the interval [0,1] uniformly subject to the condition that when we construct a circle of circumference 1 from the interval by joining the end points 0 and 1, the minimum distance of each pair is $1/(3+1)$. The smallest values in each column are denoted in bold. Regr. in the table stands for regression with 1 or 2 basis functions.

Model	median err.	q(0.75) err.	q(0.95) err.	q(0.99) err.	max err.
AVG	0.12	0.15	0.19	0.22	0.29
NN	0.12	0.14	0.2	0.24	0.4
PNN	0.11	0.13	0.17	0.19	0.24
LI	0.1	0.13	0.19	0.24	0.37
PLI	0.078	0.1	0.14	0.17	0.23
$Lipfit$	0.1	0.13	0.19	0.24	0.4
$PLipfit$	0.076	0.099	**0.13**	**0.16**	**0.22**
Regr. (1 basis)	0.095	0.12	0.17	0.2	0.3
Regr. (2 basis)	**0.069**	**0.095**	0.14	0.18	0.35

Table 2 repeats the same analysis for wiggly functions with a very small BD. The difference is: we add a uniform error $(min = -0.02, max = 0.02)$ to each of the 10^5 simulated functions. (Therefore $f \in \mathcal{ALB}(m = 1, \sigma = 0.02)$.) The errors are increased slightly but similar results as before are seen for this case. We discuss the case for which BD is not negligible later.

Tab. 2: Comparing the methods for wiggly functions with small deviation.

Model	median err.	q(0.75) err.	q(0.95) err.	q(0.99) err.	max err.
AVG	0.13	0.16	0.2	0.23	0.29
NN	0.12	0.15	0.2	0.25	0.37
PNN	0.12	0.14	0.18	0.2	0.25
LI	0.11	0.14	0.2	0.25	0.38
PLI	0.088	0.11	**0.15**	**0.17**	0.24
$Lipfit$	0.11	0.14	0.2	0.25	0.37
$PLipfit$	0.088	0.11	**0.15**	**0.17**	**0.23**
Reg. (1 basis)	0.1	0.13	0.18	0.21	0.32
Regr. (2 basis)	**0.08**	**0.1**	0.15	0.19	0.31

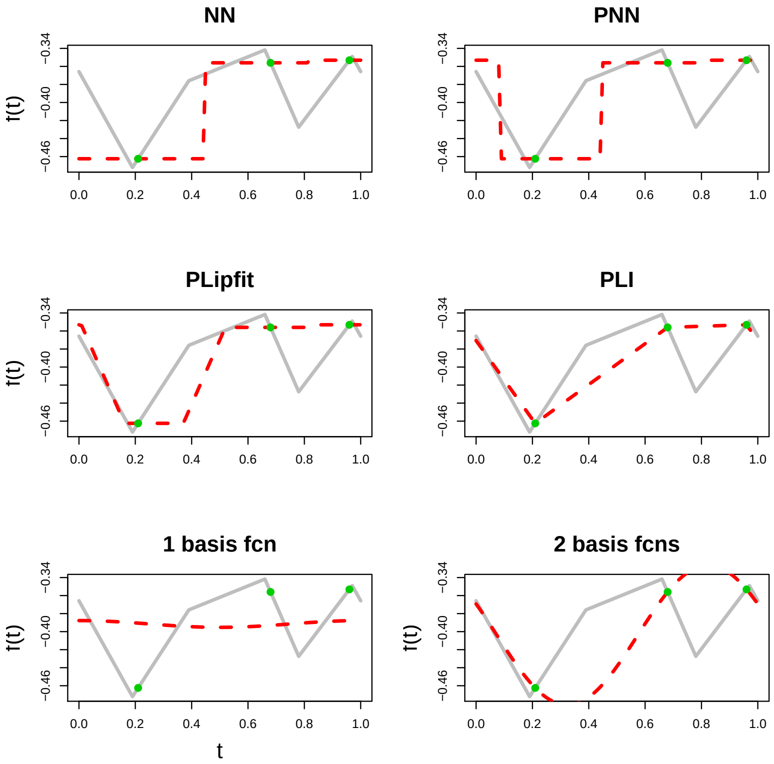

Fig. 12: Fitting the periodic curves with LB=1 generated by 5 break points. The 3 data points are taken from the interval [0,1] uniformly subject to the condition that when we construct a circle of circumstance 1 from the interval by joining the end points 0 and 1, the distance of every pair is at least $1/(3+1)$.

Comparison of methods in terms of integral approximation

In some applications, for example in studying the long-term effects of air pollution on health, we are more interested in the integral of the curve over a period instead of point-wise values. We introduced the family-standardized integral approximation error:

$$FIL(f,\hat{f}) = |\int_a^b f(t)dt - \int_a^b \hat{f}(t)dt|/(m(b-a)).$$

We showed before that $WAVG$ is the optimal method for non-periodic case and $PWAVG$ is optimal for the non-periodic case. Other methods that can be considered are AVG or using basis functions. Table 3 compares these methods. It shows that in fact $PWAVG$ performs the best; it is followed by the non-periodic version; and AVG is inferior to both methods. Moreover the regression using basis functions still perform the worst in higher values of the quantiles.

Tab. 3: Comparing the methods for the integral approximation error.

Model	median err.	q(0.75) err.	q(0.95) err.	q(0.99) err.	max err.
AVG	0.013	0.024	0.044	0.06	0.11
WAVG	0.014	0.025	0.047	0.066	0.14
PWAVG	**0.012**	**0.021**	**0.038**	**0.053**	**0.093**
Reg. (1 basis)	0.013	0.024	0.048	0.073	0.17
Reg. (2 basis)	**0.012**	**0.021**	0.045	0.07	0.19

5.2 The effect of BD magnitude on method performance

Here we perform some simulations to study the effect of the magnitude of the Bound Deviation (BD) on the method performance. We compare these methods: (1) $Lipfit$ with the same LB the curves were simulated from; (2) $Lipfit$ with LB larger than the one the curves were simulated from (denoted by $Lipfit.big$); (3) $Lipfit$ with LB smaller than the one the curves were simulated from (denoted by $Lipfit.sm$); (4) LI; (5) regularization/regression methods such as $LOESS$ and smoothing splines (See Cleveland et al. (1992) and Hastie at al. (2009)).

For simulating the curves, we pick LB=10 with 5 break points. Also the distance between each pair of the break points is taken to be at least 1/(5+2). For the sampling scheme, we consider a sparse data case: From each of [0,1/4], (1/4,1/2], (1/2,3/4], (3/4,1], we take 2 points uniformly at random. Therefore there are 8 points available from [0,1] and there is some assurance to cover the whole interval due to the sampling scheme. In contrast to the previous simulations for which we assume BD to be very small, here we consider larger BDs to study the effect of its magnitude. Figure 13 depicts the fits of various methods to the 8 points for one out of 1000 simulations for BD=1.

To investigate the method performance dependence on the BD magnitude, we consider two cases: Case 1, BD=0.5; Case 2, BD=1.5. Figure 14 presents $(25\%, 50\%, 75\%)$ quantiles of the $MPWL$ for the methods (1) $Lipfit$; (2) $Lipfit.big$; (3) $Lipfit.sm$; (4) LI; (5) $LOESS$. Below we summarize the results:

- In Case 1, where BD=0.5 and smaller than Case 2, we observe that the methods $Lipfit$ and LI perform almost equally well and outperform the other methods.

- In Case 2, in contrast to Case 1, we observe that $Lipfit$ and $Lipfit.sm$ perform almost equally well, outperforming LI in particular.

- In both cases $Lipfit.big$ performs poorly since assuming a too big $Lipfit$ will make the approximation tend to the NN method which is a poor method.

- The intuition that LI is performing better in contrast to $Lipfit.sm$ in Case 1 and this is reversed in Case 2 is as follows: In Case 1 the BD is relatively small and therefore joining the available points using the LI method does not introduce a large approximation error; while in Case 2 it could introduce a large error. In contrast $Lipfit.sm$ works by moderating the slope of the joining line between two available data points too aggressively – especially in Case 1 – believing much of the slope is due to the deviation (BD) rather than a pattern (LB). This is because $Lipfit.sm$ is supposing a much smaller LB, (LB=1), than the one the curve was generated from (LB=10).

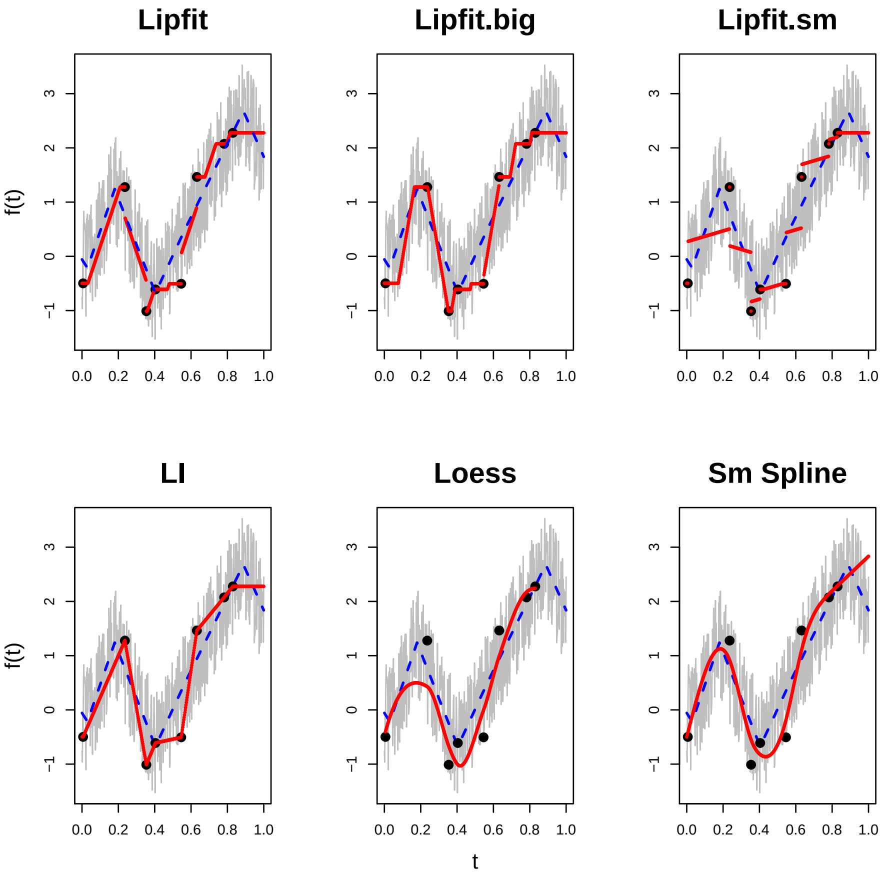

Fig. 13: The fits using various methods using 8 available points are given where the target function is given grey and the fits are given in dark. The deviation-free generated curves are given with dashed lines. The simulations are done by using 5 break points and LB=10, BD=1.

Remark. If we choose smaller number of data points, for example 5 points, similar results are obtained and the LOESS method inferiority to the other methods is magnified. We do not include those simulations here for brevity.

5.3 The effect of data sparsity on method performance

This subsection further investigates the data size and sparsity effect on the method performance. In the previous sections, we showed that when the data are sparse over all the interval of interest, the *Lipfit* method performs better in contrast to NN, LI and standard smoothing methods. We also studied the effect of using a too big LB or too small LB in the data sparse case and for various magnitudes of error. Here we consider two new cases: (1) The data is dense over all the interval; (2) the data size is large however, the data is sparse in some subintervals due to non-uniformity of

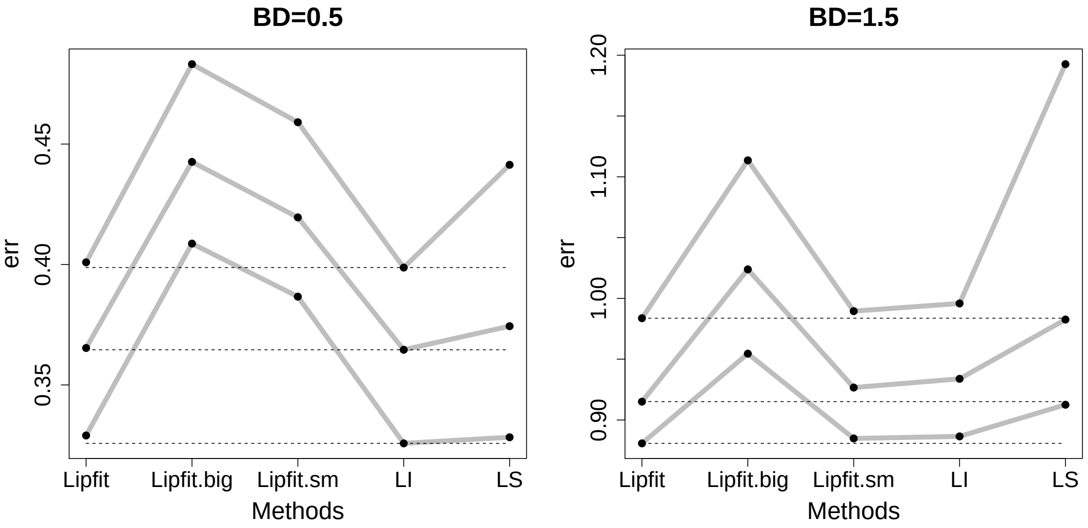

Fig. 14: The plots depict the $(25\%, 50\%, 75\%)$ quantile of the MPWL by various method for two cases: left panel is for BD=0.5; right panel is for BD=1.5

the data locations. We call such data "locally sparse". For both cases, we simulate curves with 5 break points and with LB=10, BD=0.5.

- Case 1: From each of the intervals [0,1/4],(1/4,1/2],(1/2,3/4],(3/4,1], we take 10 points uniformly at random.
- Case 2: From each of the interval [0,1/4], we sample 50 points uniformly at random and only one point from each of the intervals (1/4,1/2],(1/2,3/4],(3/4,1], uniformly at random.

Figure 15 depicts the error quantiles for the two cases and we summarize the results as follows:

- In the data dense case (Case 1, left panel), the smoothing method ($LOESS$) has performed optimally for the lower quantiles but still inferior to the $Lipfit$ method with the correct or small LB in higher quantiles.
- In the locally sparse data case (Case 2, right panel), we observe that the result is almost identical to the data sparse case over the entire interval. In other words having a large data set is not necessarily going to change the results if the data is still sparse in large subintervals and the smoothing methods will continue to perform poorly.

6 Trade-off between LB and BD

For a given function $f : D \to \mathbb{R}$, LB and BD are not unique. In fact for any BD, $\sigma \in \mathbb{R}^{\geq 0}$ one can find, LB, $m \in \mathbb{R}^{\geq 0}$ (non-negative numbers) such that $f \in \mathcal{ALB}(m, \sigma)$. This is the motivation for the following definition.

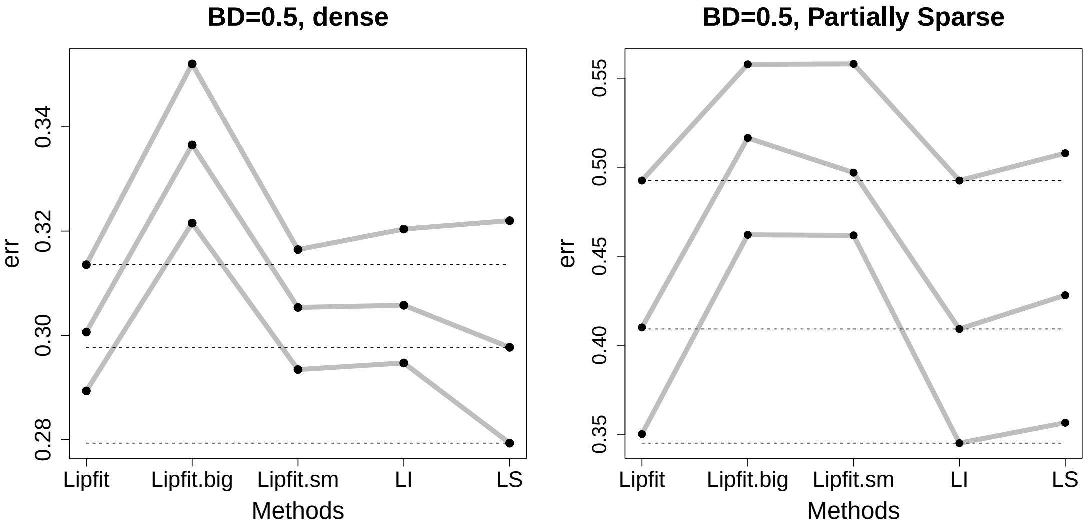

Fig. 15: The figure depicts $(25\%, 50\%, 75\%)$ quantiles of the MPWL for two cases. (Left Panel) $BD = 0.5$ and dense data of size 40. (Right Panel) $BD = 0.5$ with data size 53 and non-uniform data, dense in some interval $[0, 1/4]$ with 50 data points and one data point in each of (1/4,1/2],(1/2,3/4],(3/4,1].

Definition 6.1: Suppose $f : D \subset \mathbb{R}^d \to \mathbb{R}$. Then the LB-BD function (curve) associated with f – denoted by γ_f – is defined as follows:

$$\gamma_f : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0};$$

$$\gamma_f(m) = \inf\{\sigma \mid f \in \mathcal{ALB}(D, m, \sigma)\}.$$

Remark. We can also consider an "inverse" for γ_f :

$$\gamma_f^{-1} : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0};$$

$$\gamma_f^{-1}(\sigma) = \inf\{m \mid f \in \mathcal{ALB}(D, m, \sigma)\}.$$

We call the inverse also the LB-BD curve by slight abuse of naming. In Figure 16 (Right Panel) the LB-BD curve for the function $f(x) = \sin(2\pi x)$ is given. The LB for f is equal to 2π. However if we allow for a deviation of σ, as depicted by the grey curves (Left Panel), there is a function inside the area defined the grey curves which has a smaller LB.

We can also define a LB-BD curve for the periodic case as follows.

Definition 6.2: Suppose $f : [a, b] \to \mathbb{R}$. Then the periodic LB-BD curve associated with f, γ_f^p, is defined as follows:

$$\gamma_f^p : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0};$$

$$\gamma_f^p(m) = \inf\{\sigma \mid f \in \mathcal{PALB}([a, b], m, \sigma)\}.$$

In the following lemma, we give the LB-BD curve for some simple functions.

Lemma 6.1: Below we give the LB-BD curve, γ_f^{-1}, for various functions $f : [a, b] \to \mathbb{R}$.

(a) $f(x) = mx$:

$$\gamma_f^{-1}(\sigma) = \max\{0, m - 2\sigma/(b-a)\}.$$

(b) $f(x) = m|x - (b-a)/2|$:

$$\gamma_f^{-1}(\sigma) = \max\{0, m - 4\sigma/(b-a)\}$$

(c) $f(x) = \sin(2\pi x)$ and $[a, b] = [0, 1]$ then $\gamma_f^{-1}(\sigma) = |2\pi \cos(2\pi x_B)|$, where x_B is the unique solution of the following equation:

$$\sin(2\pi x_B) - 2\pi(x_B - 1/2)\cos(2\pi x_B) - \sigma = 0,\ 1/4 \leq x_B \leq 1/2. \tag{6}$$

(See Figure 16.)

Proof

(a) Consider the left panel of Figure 17 for the proof. Without loss of generality we assume the slope of AB, m, is positive. If we allow σ to be the deviation, and the y-value of B_2 is larger than A_1, then the line connecting A_2 and B_1 minimizes the LB and the slope of that line is equal to $m - 2\sigma/(b-a)$. If the y-value of B_2 is less than or equal to A_2, then a horizontal line starting from A_2 will minimize LB. We conclude the infimum LB is equal to $\max\{0, m - 2\sigma/(b-a)$.

(b) Consider the right panel of Figure 17 for the proof. An argument similar to above can be used.

(c) Consider the left panel of Figure 16 for the proof. Let $f(x) = sin(2\pi x)$, $f_1(x) = f(x) - \sigma$, $f_2(x) = f(x) + \sigma,\ x \in [0, 1]$. Define

$$A = (x_A, y_A) = (1/4, f_1(1/4)),\ C = (1/2, f(1/2) = 0),\ E = (3/4, f_2(3/4))$$

Also let $B = (x_B, f_1(x_B))$ be a point on the trajectory of $f_1(x),\ x \in [1/4, 1/2]$ such that BC is tangent to $f_1(x)$ trajectory. Then let D be the symmetric image of B with respect to C. Then DB is also tangent to f_2 trajectory by symmetry.

Then consider the curve (dashed) that goes along $f_1(x)$ trajectory from A to B; then goes along the line segment BD; then goes along the $f_2(x)$ trajectory to reach E. We claim that this curve has the minimum possible LB while satisfying the deviation σ. First note that such a curve satisfies the deviation, σ, and can be extended in the same manner to $[0, 1]$ (dashed curve). We denote this curve by g. Then note that $Lip(g)$ is the same when applied to the domain $[x_A, x_E]$ or when applied to $[0, 1]$. In fact $Lip(g)$ equals to slope of BC line which we denote by l. Therefore it only remains to show no other curve achieves this and obtain a strictly smaller LB on $[x_A, x_E]$.

Suppose h is another curve defined on $[x_A, x_E]$ which satisfies the deviation σ and has a smaller LB than g on $[x_A, x_B]$. Without loss of generality (and by symmetry), we can assume

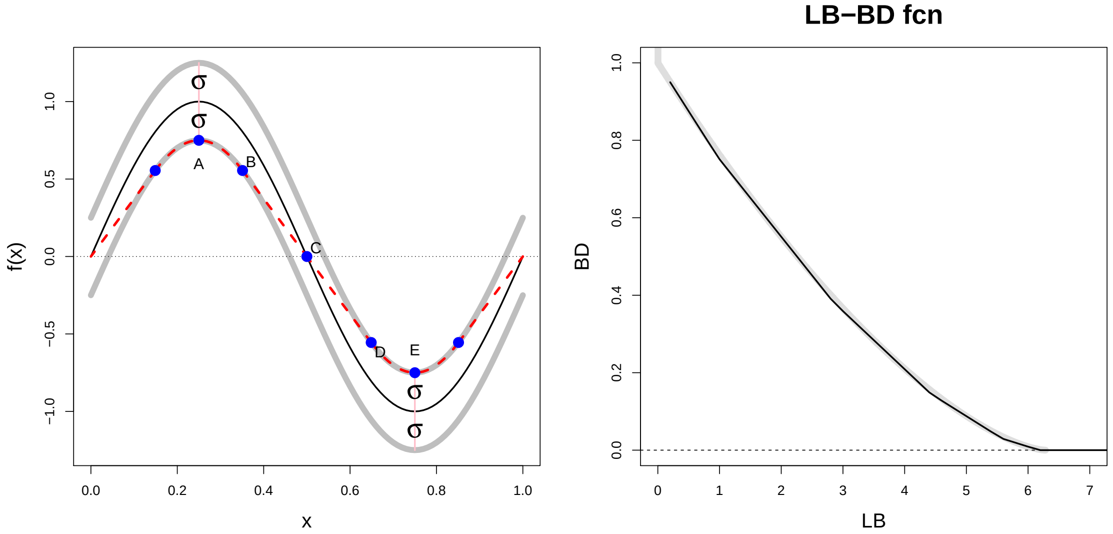

Fig. 16: (Left Panel): The function $f(x) = sin(2\pi x)$ is give in black with the grey curves mark the boundaries for the curves which deviate from f at most as much as σ. A curve which is inside the boundaries and attains the smallest possible LB is also given. (Right Panel): LB-BD curve for $f(x) = \sin(2\pi x)$, $x = [0, 1]$. Black curve is obtained by analytic solution and the grey curve is obtained by solving a convex optimization problem.

that $h(x_C) \leq y_C$ and we focus on the $[x_B, x_C]$ interval. (If $h(x_C) > y_C$ we repeat the following proof by focusing on $[x_C, x_D]$.)

Then note that h must satisfy $h(x_B) \geq f_1(x_B) = g(x_B)$ since h satisfies the deviation σ. Now the line segment from $(x_B, h(x_B))$ to $(x_C, h(x_C))$ will have a slope more than l and this is a contradiction to h having a smaller LB. To complete the proof it remains to calculate the magnitude of the slope of the BC line segment. This can be found by letting the derivative of $f_1(x_B)$ equal to the slope of BC for $1/4 \leq x_B \leq 1/2$ and solve that equation for x_B:

$$2\pi \cos(2\pi x_B) = \frac{0 - (\sin(2\pi x_B) - \sigma)}{1/2 - x_B}.$$

Then we calculate $|2\pi \cos(2\pi x_B)|$ to get the magnitude of the slope.

■

6.1 Properties of LB-BD function

This subsection discusses the basic properties of LB-BD function. These properties are useful in giving intuition about the LB-BD curve, as well as calculating it for given functions.

Lemma 6.2: (Elementary Properties of LB-BD function) Suppose $f : D \to \mathbb{R}$ is a bounded function and $diam(f) = d$. Then the LB-BD curve of f has the following properties.

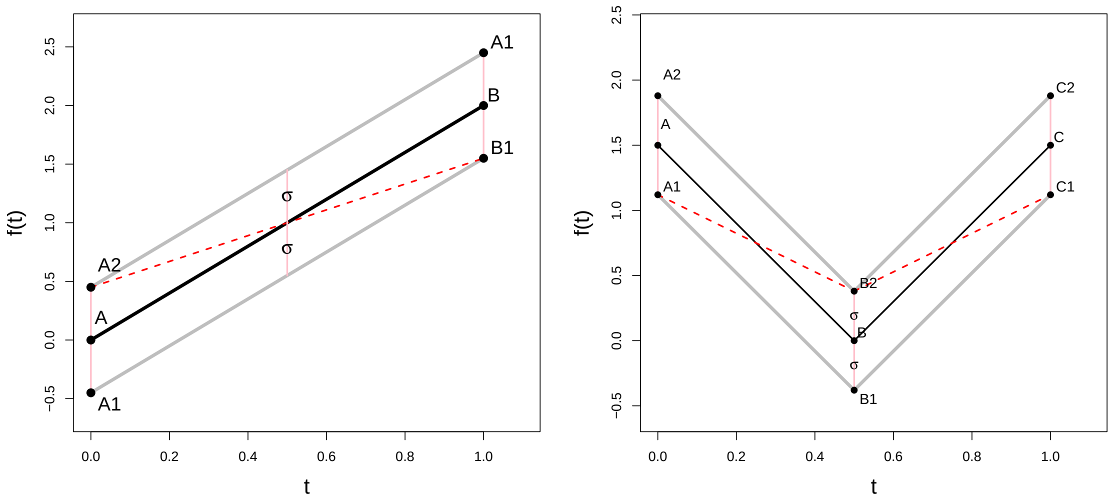

Fig. 17: The figure gives the idea of finding the LB-BD curve for a straight line (left panel) or an absolute value function (right panel). The functions are given in black, the grey curves mark the boundaries for the functions within σ deviation. The dashed curves give a function which minimizes the LB with the given σ.

(a) γ_f and γ_f^{-1} are both decreasing functions.

(b) $\gamma_f(+\infty) = 0$ and $\gamma_f(0) = d/2$.

(c) $\gamma_f^{-1}(+\infty) = 0$ and $\gamma_f^{-1}(0) = Lip(f)$.

(d) Suppose $f : D \to \mathbb{R}$ and $f_1 : D_1 \subset D \to \mathbb{R}$ is a restriction of f from domain D to $D_1 \subset D$. Then $\gamma_{f_1}(m) \le \gamma_f(m),\ m \ge 0$.

(e) Suppose $k > 0$ and define $f_1(x) = f(kx)$ for $x \in [a/k, b/k]$. Then

$$\gamma_{f_1}(m) = \gamma_f(m/k).$$

(f) $\gamma_{kf}(m) = |k|\gamma_f(m/|k|)$.

Proof See Appendix. ■

Theorem 6.1: (Summation Bound on LB-BD Curve) Suppose $f = f_1 + f_2$.

(a) If $m = m_1 + m_2$ where $m_1, m_2 \ge 0$ then

$$\gamma_f(m) \le \gamma_{f_1}(m_1) + \gamma_{f_2}(m_2).$$

(b) If $\sigma = \sigma_1 + \sigma_2$ where $\sigma_1, \sigma_2 \geq 0$ then

$$\gamma_f^{-1}(\sigma) \leq \gamma_{f_1}^{-1}(\sigma_1) + \gamma_{f_2}^{-1}(\sigma_2).$$

Proof See Appendix. ■

Theorem 6.2: Both γ_f and γ_f^{-1} are convex functions.

Proof A corollary of the Summation Bound Theorem. ■

Corollary 6.1: Suppose $SPWL(f, g) \leq \sigma$. Then $|\gamma_f(m) - \gamma_g(m)| \leq \sigma/2$, $\forall m \geq 0$.

Proof Let $h = g - f$ then $SPWL(h, 0) \leq \sigma$. Therefore $diam(h) \leq \sigma$ and we conclude $\gamma_h(0) \leq \sigma/2$. Now by applying the Decomposition Theorem to $f = g + h$ and for $m_1 = m,\ m_2 = 0$:

$$\gamma_f(m) \leq \gamma_g(m) + \gamma_h(0) \leq \gamma_g(m) + \sigma/2,$$

$$\Rightarrow \gamma_f(m) - \gamma_g(m) \leq \sigma/2.$$

Similarly we can show that: $\gamma_g(m) - \gamma_f(m) \leq \sigma/2$, and thus the proof is complete. ■

The following theorem provides a link between the LB-BD of a function and a grid approximation of the function for the 1-dimensional case.

Theorem 6.3: (LB-BD Grid Approximation) Suppose $f : [a, b] \rightarrow \mathbb{R}$ and consider a grid approximation given by $\mathbf{x} = (x_1, \cdots, x_n)$ and $\mathbf{y} = (f(x_1), \cdots, f(x_n))$ and denote the grid function by g. Denote the linear interpolation of g on $[a, b]$ by $LI(g)$ and suppose $SPWL(f, LI(g)) \leq \sigma$. Then $0 \leq \gamma_f(m) - \gamma_g(m) \leq \sigma$. Note that γ_g is calculated with respect to the domain of g which is $\mathbf{x} = (x_1, \cdots, x_n)$ and not $[a, b]$.

Proof See Appendix. ■

Remark. For most functions (even simple smooth ones) obtaining the LB-BD curve analytically is not possible. However using this theorem, we can find a grid for which the grid approximation is arbitrarily close to the original function. Then if we are able to find the LB-BD curve for the gridded function, we can approximate the LB-BD curve of the original function closely. This is also useful from a computational point of view when we are working with data or gridded functions. For example if we are working with data with $\mathbf{x} = (x_1, \cdots, x_N)$ and $\mathbf{y} = (y_1, \cdots, y_N)$ where N is large as we show in the following the LB-BD curve calculation becomes computationally intensive. However we may be able to find sub-grids of $\mathbf{x}$ and $\mathbf{y}$: $\mathbf{x}' = (x_{i_1}, \cdots, x_{i_n}); \mathbf{y}' = (y_{i_1}, \cdots, y_{i_n})$, for $1 \leq i_1 < \cdots < i_n \leq N$, such that $n << N$ (n is much smaller than N) and $SPWL(\mathbf{y}, LI(\mathbf{x}; \mathbf{x}', \mathbf{y}')) \leq \sigma$. We can approximate LB-BD curve of $(\mathbf{x}, \mathbf{y})$ by calculating that of $(\mathbf{x}', \mathbf{y}')$ and noting that $\gamma_{(\mathbf{x},\mathbf{y})} - \gamma_{(\mathbf{x}',\mathbf{y}')} \leq \sigma$.

6.2 Prediction errors given the LB-BD curve

For a function $f : D \subset \mathbb{R}^d \to \mathbb{R}$ observed at given points $\mathbf{x} = (x_1, \cdots, x_n)$ and with values equal to $\mathbf{y} = (y_1, \cdots, y_n)$, we found the errors for estimating it over the entire domain D: $IE, DIE, SPWE, DSPWE, DSPWE$ for each of the methods (e.g. LI and $Lipfit$) and for a given fixed pair of LB and BD. Now suppose instead of one single pair, a "partial LB-BD" curve

$$\gamma_f : U \subset \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0},$$

is given. We can think of U as a subset of $\mathbb{R}^{\geq 0}$, where "information" is available about f. Then we can extend the above errors of estimating f on D by taking the infimum over all the available pairs of LB-BD. Suppose *approx* denote the method (for example *approx* $= Lipfit$) and E the error measure (for example $E = DSPWE$), then the minimal error of estimating f given γ_f is defined as follows:

$$\Upsilon\{E, approx, f, D, x, y \mid \gamma_f\} = \inf_{m \in U} E\{approx, f, D, x, y \mid m, \gamma_f(m)\}.$$

Since f belongs to all $(m, \gamma_f(m))$, when the infimum is obtained by some $m_0 \in U$, we can apply the method *approx* with that $(m_0, \gamma_f(m_0))$ to get the error $E = \Upsilon$, therefore minimizing the error on D as much as possible. If the infimum is not obtained for any small ϵ there is $m_0 \in U$ so that E is within a radius of ϵ of Υ.

7 Calculating LB-BD function

This section assumes we have access to gridded data and using that we develop methods to calculate the LB-BD curve. Theorem 6.3 then can be applied to make a connection to a full curve or a curve defined on a more fine resolution. This may seem contradictory to the sparse data situation at first but as we discuss in more details later the LB-BD curve for many applications does not vary much from one time period to another or we may use the LB-BD curve of a temporal process in one location with dense data for another close location with sparse data. As an example we show that the LB-BD curve is similar for the temporal process of several central sites for Ozone process in Southern California.

Below we start with a heuristic moving average filtering method for calculating the LB-BD curve for 1-d case. Then we proceed to an exact method by representing the LB-BD calculation as a convex optimization problem. This convex optimization method also works for the multidimensional data case. However for the 1-d case ,we also present a faster method by representing the problem as a different convex optimization method.

7.1 Filtering method

We start with the case for which we have access to a complete curve over the period $[a, b]$ and would like to find pairs m, σ for which the curve belongs to $\mathcal{ALB}(m, \sigma)$. In other words we are interested in estimating the LB-BD curve using data.

Filtering method for calculating LB-BD curve for equally-spaced grid: Here we present a method of finding LB-BD curve for a given function by calculating "weighted moving averages". The idea is to "smooth" the curve and use the smoothed version as an approximation.

Suppose the value of the function is available on an equally-spaced grid $\mathbf{x} = (x_1, \cdots, x_N)$ where $x_1 = a$ and $x_N = b$. Note that we can approximate a LB for a function f given on a grid by

$$m_f = \max\{\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i},\ i = 1, \cdots, N-1\}.$$

Now let $w = (1/2, 1/3)$ be a "weight vector" and let $w^\star = (1/3, 1/2, 1, 1/2, 1/3)$ be its "extended symmetric" form. Then we can define a new function g_0 on the same grid as follows:

$$g_0(x_i) := \frac{1}{\sum_{k=1}^{5} w_k^\star}[f(x_i) + \sum_{j=1}^{2} w_j(f(x_{i-j}) + f(x_{i+j}))].$$

For this to be well-defined, we need to define values for $x_{-1}, x_0, x_{N+1}, x_{N+2}$ and we let

$$x_{-1} := x_1,\ x_0 := x_1;\ x_{N+1} = x_N,\ x_{N+2} := x_N.$$

Then g_0 will approximate f on $[a, b]$ and we expect g_0 to have a smaller variation (LB) due to the averaging we performed. In other words, we expect

$$m_{g_0} = \max\{\frac{g_0(x_{i+1}) - g_0(x_i)}{x_{i+1} - x_i},\ i = 1, \cdots, N-1\},$$

to be smaller than m_f. The price we pay for this would be the introduced deviation

$$\sigma_{g_0} = \max\{|f(x_i) - g_0(x_i)|,\ i = 1, \cdots, N\}.$$

By construction, we have $f \in \mathcal{ALB}(m_{g_0}, \sigma_{g_0})$. Moreover we can apply this process iteratively by applying the filtering described above to g_0 and find a more smooth function g_1 and obtain a smaller m_{g_1} for which $f \in \mathcal{ALB}(m_{g_1}, \sigma_{g_1})$. Again we may pay the price that σ value becomes larger. After repeating this process several times, we obtain a sequence of pairs

$$(m_{g_0}, \sigma_{g_0}), (m_{g_1}, \sigma_{g_1}), (m_{g_2}, \sigma_{g_2}), \cdots$$

for which $f \in \mathcal{ALB}(m_{g_i}, \sigma_{g_i}),\ i = 0, 1, 2, \cdots$.

We can also modify this method to calculate the LB-BD curve for $f \in \mathcal{PALB}(m, \sigma)$. The difference for the calculation of m and σ is we need to calculate them on an extended grid as explained above.

We know by definition that the LB-BD curve is decreasing in BD (or LB). In most cases that we used the above procedure, it produced a decreasing curve. However there were some instances for which, at the start of the curve, an increasing trend was observed for the periodic case. Therefore we use a "monotonization" method to create a decreasing curve from a given curve: Let $\gamma'(t)$ be a given curve. Then we define the decreasing monotonized curve to be

$$\gamma(t) = \inf\{\gamma'(u) | u \geq t\}.$$

In order to test this method, we perform a simulation study which is presented in Figure 18. Left panel shows the true curve $f_0(x) = -2x^2 + (4/3)x;\ x \in [0, 1]$ in black; the version with added uniform deviation ($\min = -0.05, \max = 0.05$) is shown in hallow circles. We denote this function by

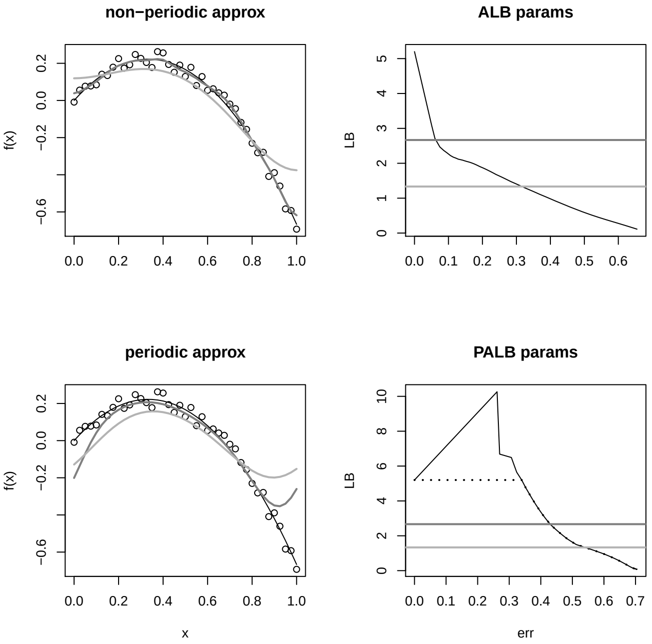

Fig. 18: This figure presents the result of calculating LB-BD curve for equally-spaced grid data using the filtering approach. The top panels correspond to the case for the $\mathcal{ALB}$ family and the bottom panel for the $\mathcal{PALB}$ family. The right panels depict the gridded data (circles) with smoothed curves found in the process of creating the corresponding LB-BD curve. The right panels depicts the LB-BD curve calculated using the filtering approach (black) and the dotted depict the monotonized versions. For each LB-BD curve two specific LB values are picked by straight lines (grey and dark grey) and their corresponding BD. For each LB the BD is the x-value where the corresponding line intersects the LB-BD curve.

f. The top panels correspond to finding the parameters (m, σ) such that $f \in \mathcal{ALB}(m, \sigma)$ and the bottom panels correspond to finding parameters such that $f \in \mathcal{PALB}(m, \sigma)$. Two grey curves have bounded derivative of at most 8/3 (dark grey) and 4/3 (light grey) and are found by smoothing the curve sequentially. The right panel shows the results of the sequential filtering by plotting the BD in each iteration against the LB. The lines corresponding to derivatives 8/3 and 4/3 are also given. Note that 8/3 coincides with the bound for the derivative of the true function for which the deviation has turned out to be approximately 0.2 for the $\mathcal{ALB}$ case as we expect. Also the deviations are larger for the $\mathcal{PALB}$ case as we expect. For the $\mathcal{PALB}$ case the LB-BD curve is not decreasing at the beginning and therefore a monotonization is applied to the curve to get the dotted curve in the figure.

Below we present an extension of the filtering method to the case where the data grid is not equally-spaced.

Filtering method for calculating LB-BD curve for general grid:

(i) Get as input an ordered but possibly non-equally spaced grid $\mathbf{x} = (x_1, \cdots, x_n)$ on $[a, b]$ and corresponding values $\mathbf{y} = (y_1, \cdots, y_n)$.

(ii) Let $m' = \max\{\frac{y_{j+1}-y_j}{x_{j+1}-x_j},\ j = 2, \cdots, n\}$ so that m' is the maximum slope if the data is fit by the LI method.

(iii) Apply the $Lipfit$ method with $LB = m'$ to $\mathbf{x}$ and $\mathbf{y}$. Using the resulting fit create equally-spaced gridded data on $[a, b]$.

(iv) Apply the filtering method for the equally-spaced data to the data obtained in the last step.

Remark. For the periodic case, we simply apply the $PLipfit$ instead.
Remark. In Step (iii), we can also use LI to create the fit from which the equally-spaced data grid is created.

In Figure 19 non-gridded data shown in filled points is used to find the LB-BD curve. The curve in black is created using the complete data; the dotted and dashed curves are created using the above method with LI and $Lipfit$ respectively.

7.2 Exact convex optimization method

This subsection discusses exact methods for calculating the LB-BD function for gridded functions. Suppose $f : D \subset \mathbb{R}^d$ is a given function for which we like to find the LB-BD curve. To calculate $\gamma_f(m)$, we need to solve:

$$\inf_{g \in \mathcal{LB}(m)} \sup_{x \in D} |f(x) - g(x)|. \tag{7}$$

Here we present an exact method for estimating the LB-BD function by solving Equation 7 using convex optimization, when D is a finite subset.

Exact method (convex optimization) for calculating LB-BD function: Suppose $f : D \subset \mathbb{R}^d \to \mathbb{R}$, is defined on a finite domain $\mathbf{x} = (x_1, \cdots, x_n)$ (D is the set defined by the elements of $\mathbf{x}$) and takes the values $\mathbf{y} = f(\mathbf{x}) = (y_1, \cdots, y_n)$. Consider an approximation of $\mathbf{y} = f(\mathbf{x})$ by $y^\star$:

$$y_i = y_i^\star + r_i,$$

where r_i is the deviation from the true value at y_i. This approximation belongs to $\mathcal{LB}(m)$ if and only if

$$y_i^\star - y_j^\star \leq m||x_i - x_j||,\ \forall i, j \in \{1, \cdots, n\},$$

(Beliakov (2006)). We conclude that finding the value of $\gamma_f(m)$ is equivalent to minimizing $\max\limits_{i=1,\cdots,n} |r_i|$.

Now we pose the convex optimization method:

(a) For finding γ_f:

$$\begin{aligned} &\text{minimize} && \max_{i=1,\cdots,n} |r_i|, \\ &\text{subject to} && r_i - r_j \leq m||x_i - x_j|| + (y_j - y_i), \\ &&& \forall i, j \in \{1, \cdots, n\}. \end{aligned}$$

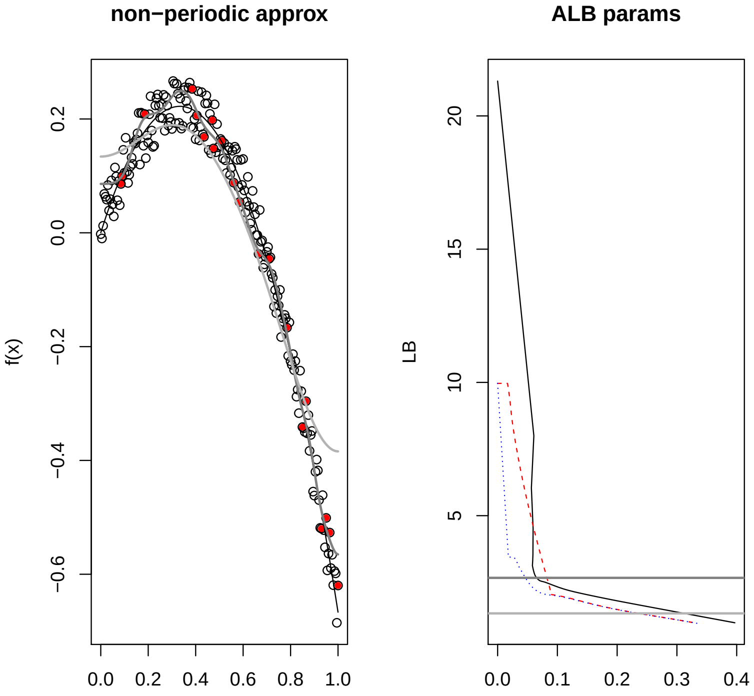

Fig. 19: This figure presents estimating the LB-BD function for a general grid in 1-d case. Left panel depicts the full gridded data set by hallow circles and a subset of it by filled circles. In the right panel, the black curve depicts the LB-BD function constructed using the full data set. We have also depicted the LB-BD curve for the subset using the filtering approach for general grid (dashed and dotted). In Step (iii) of the LB-BD calculation, we have used both LI (dotted) and $Lipfit$ (dashed) for interpolation.

(b) For finding γ_f^{-1} :

$$\begin{aligned}
&\text{minimize} && \max_{i=1,\cdots,n} |(y_j - y_i + (r_i - r_j))|/||x_i - x_j||,\\
&\text{subject to} && |r_i| \leq \sigma, i \in \{1, \cdots, n\}\\
& && \forall i, j \in \{1, \cdots, n\}.
\end{aligned}$$

We can write the above problems in matrix form by defining

- $\mathbf{1}_n := (1, \cdots, 1)$ a vector of ones
- $r = (r_1, \cdots, r_n)^T$ column vector of deviations

- $n \times n$ matrix E : $E(i,j) = m||x_i - x_j|| + (y_j - y_i)$
- $n \times n$ matrix Z : $Z(i,j) = |(y_j - y_i + (r_i - r_j))/||x_i - x_j||$

Then for example we can write the restrictions in (a) in the following compact form:

$$r\mathbf{1}_n - \mathbf{1}_n^T r^T \leq E.$$

These problems can then be implemented in CVX package of Matlab (See Grant and Boyd (2008) and Grant and Boyd (2013)). (a) is the minimization of a maximum of absolute values of n^2 affine functions and with n affine constraints. (b) is the minimization of a maximum of absolute values of n affine functions and with n^2 affine constraints.

Fast exact method (convex optimization) for 1-d case: Suppose we want to calculate $\gamma_f(m)$ where f is defined on $[a, b]$ and is equal to the linear interpolation of $a \leq x_1 < x_2 < \cdots < x_n \leq b$ with values $(y_1, \cdots, y_n)$. Then

$$\gamma_f(m) = \inf_{g \in \mathcal{LB}(m)} SPWL(f, g),$$

which is equal to

$$\gamma_f(m) = \inf_{g \in \mathcal{PL}(m)} SPWL(f, g),$$

because $\mathcal{PL}(m)$ is dense in $\mathcal{LB}(m)$. Now suppose a $g \in \mathcal{PL}(m)$ attains $SPWL(f, g) = \sigma$. Because g is piece-wise linear, g has breakpoints at $a \leq z_1 < z_2 < \cdots < z_k \leq b$. Clearly we can assume z_is include the x_is as we do not require g to change slope at every break point. Moreover we claim that there is always a $h \in \mathcal{PL}(m)$ which is as close to f as g, $SPWL(f, h) \leq \sigma$, and only requires break points at x_is. We define such a h by modifying g. We define h to be the linear interpolation of $x = (x_1, \cdots, x_n)$ with values at $y = (g(x_1), \cdots, g(x_n))$. Then it is clear that $h \in LB(m)$ (because g is) and $SPWL(f, h) = SPWL(f, g) = \sigma$. Any such h can be written as a linear combination of

$$1, 1_{\{x > x_1\}}(x - x_1), 1_{\{x > x_2\}}(x - x_2), \cdots, 1_{\{x > x_{n-1}\}}(x - x_{n-1});$$

$$h(x) = c_0 + \sum_{i=1}^{n} m_i 1_{\{x > x_i\}}(x - x_i)$$

where $1_{x > x_i} = 1 \iff x > x_i$. Now using this definition at the breakpoints $\mathbf{x} = (x_1, \cdots, x_n)$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & x_2 - x_1 & 0 & 0 & \cdots & 0 \\ 1 & x_3 - x_1 & x_3 - x_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n - x_1 & x_n - x_2 & x_n - x_3 & \cdots & x_n - x_{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \end{pmatrix} + \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{pmatrix}, \tag{8}$$

where c_0 is the value of h at y_1; $m_1, \cdots, m_{n-1}$ are the slopes at the break points. Then f belongs to $\mathcal{ALB}(m, \sigma)$ if and only if

$$\max_{i=1,\cdots,n-1} |m_i| \leq m, \quad \max_{i=1,\cdots,n-1} |r_i| \leq \sigma. \tag{9}$$

For simplicity in exposition let us denote the key matrices used here as follows:

$$\mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}, \; \mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & x_2 - x_1 & 0 & 0 & \cdots & 0 \\ 1 & x_3 - x_1 & x_3 - x_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n - x_1 & x_n - x_2 & x_n - x_3 & \cdots & x_n - x_{n-1} \end{pmatrix}, \; \mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{pmatrix},$$

$$\mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-1} \end{pmatrix}.$$

Then we can write Equation 8 in the more compact form:

$$\mathbf{y} \quad = \quad \mathbf{X} \begin{pmatrix} c_0 \\ \mathbf{m} \end{pmatrix} + \mathbf{r}$$

Additionally if we define $\mathbf{1_n}$ to be a column vector of all 1s and of length n, for all natural numbers n, we can also write the conditions in 9 in the matrix form:

$$-m\mathbf{1_{n-1}} \leq \mathbf{m} \leq m\mathbf{1_{n-1}}, \;\; -\sigma\mathbf{1_n} \leq \mathbf{r} \leq \sigma\mathbf{1_n}.$$

Or if we use the definition of maximum norm(infinity norm): $||(x_1, \cdots, x_n)||_\infty = \max_{i=1,\cdots,n} |x_i|$, we can write them as

$$||\mathbf{m}||_\infty \leq m, \; ||\mathbf{r}||_\infty \leq \sigma.$$

For the periodic case, $\mathcal{PALB}(m, \sigma)$, we need an extra condition which assures that the magnitude of the slope of the line going from the last point $(x_n, h(x_n))$ to $(b + (x_1 - a), h(x_1) = c_0)$ is also less than m:

$$-m \leq \frac{\sum_{i=2}^{n} (x_i - x_{i-1}) m_{i-1}}{(b - a) - (x_n - x_1)} \leq m$$

Now we pose the convex optimization method:

(a) For finding γ_f:

$$\begin{aligned} \text{minimize} \;\; & ||\mathbf{r}||_\infty, \\ \text{subject to} \;\; & -m\mathbf{1_{n-1}} \leq \mathbf{m} \leq m\mathbf{1_{n-1}} \end{aligned}$$

(b) For finding γ_f^{-1} :

$$\begin{aligned} \text{minimize} \quad & ||\mathbf{m}||_\infty, \\ \text{subject to} \quad & -\sigma\mathbf{1_{n-1}} \leq \mathbf{r} \leq \sigma\mathbf{1_{n-1}} \end{aligned}$$

Table 4 compares the computation time for 1-d functions using both the fast (1-d) and general method. The computational gain is very significant and grows with the data size.

Tab. 4: Comparison between the optimization methods (to calculate LB-BD curve) for 1-d and general method. The methods are applied to an equally-spaced gridded version of the function $\sin(2\pi x)$, $x \in [0,1]$ with various resolutions determined by data size. In each case, the BD value is calculated for LB in $\{0, 0.1, 0.2, \cdots, 10\}$.

data size	General method time(s)	Fast method (1-d) time(s)	Time Ratio
10	38	19	2.0
20	58	21	2.7
30	49	175	3.6
40	68	606	8.9
50	87	2139	25

8 Choosing appropriate parameters using data

In order to be able to use the prediction errors of various methods or for applying the *Lipfit* method, one needs to find appropriate LB and BD. In some applications, this can come from the expert knowledge of the practitioner. It is often unreasonable to assume that high-order derivatives exist and also require the practitioner to know about its magnitude. On the other hand, for many applications, using physical/chemical/biological properties of the process, the practitioner may obtain a bound on the rate of change of the process as measured by LB and a small-scale deviation (BD). The small-scale deviation may refer to the accuracy of the measurement device or small-scale variations of the process. However one does not need to merely rely on the expert knowledge or the properties of the processes. In the following we show how one can use available data to get an estimate of these parameters m and σ.

In the above, we presented a method to calculate LB-BD curve when we have sufficient data from the process under the study. One question is given an LB-BD curve which pair should be used for fitting the *Lipfit* method and calculating the prediction errors. The main method that we discuss here is picking the pair which minimizes the given errors. We also provide other "validation" methods (either using multiple instances of the process or cross-validation). Also after all the goal is to approximate curves when enough data is not available and it may look such a method is not useful in practice. Here we discuss under what situations this method may be useful by giving concrete situations where the methods can be applied. Also in Section 9, we apply the methods to air pollution data.

8.1 Prediction Error Minimization Method (PEM):

As we discussed before by definition for any function, there are infinitely many pairs of LB-BD, (m, σ), for which the given function belongs to $\mathcal{ALB}(D, m, \sigma)$. Then we face the problem of choosing a specific pair (m, σ) for calculating the prediction errors and applying the *Lipfit* method. If the estimated LB-BD curve is close to the true curve (or we have some expert knowledge to confirm that), then for given data, we can choose the pair which minimizes the (data-informed) prediction errors. Clearly the chosen pair will depend on the data sparsity and the values of the observed function if they are available. In other words PEM maps any given dataset, $(\mathbf{x}, \mathbf{y})$, to a specific pair which minimizes the appropriate prediction error of interest:

$$PEM: \ (\mathbf{x}, \mathbf{y}, \gamma_f) \mapsto (m, \gamma_f(m)).$$

We will use and expand on this method in Section 9. Below we outline other methods to pick a pair which are based on validation.

8.2 Validation using multiple instances

Example 1 (Multiple instances of the process): Suppose $\{Y(t)\},\ t \in [a, b]$ is a slow-moving process (for example with $m \leq 20$ and $\sigma \leq 1$) and many, say 50, instances of this process are observed on a grid of size 200. Also assume we have 3 observations of this process for a new instance and the goal is to approximate this new instance.
In this case the LB-BD can be estimated from each of the available 50 instances and in fact even a distribution for LB-BD curve can be derived and we can apply the methods discussed in this paper.

To assess this method, we simulate 50 instances for 1-d process defined on [0,1] with $m \leq 20$ and $\sigma \leq 1$. We allow 8 break points in the [0,1] interval and at each break point, a slope magnitude from the interval $[m/2, m]$ and a slope sign (with equal probability for positive and negative) are sampled. Figure 20 depicts the simulation results for 50 LB-BD curves obtained from each instance in grey and the quantile value curves at 25% and and 75% are given in black. We observe that there is good consistency among the LB-BD curves and if we were to use the LB-BD curve from one instance of the series for another instance for calculating the errors or applying the $Lipfit$ method, the results will be acceptable.

Our goal is to interpolate a new instance for which very few data points are available. Thus standard time-series methods do not apply without making strong assumptions. Also standard smoothing methods such as smoothing splines are susceptible to have very poor out-of-sample performance due to data-sparsity.

Validation Method: Suppose data are available for 50 instances of a time series defined on $[0, 1]$ with m=10 and σ=1 for 200 data points on a grid defined on $[0, 1]$. In order to choose LB, consider a grid for $\log_{10}(LB)$ defined on the interval $[l_1, l_2]$. For every instance of the series, we can pick a *training sample* of k points (for example for $k = 8$) and then use each of the points l from the $\log_{10}(LB)$ grid to fit the training sample using $Lipfit$. From every such fit, we approximate the data we have set aside, the *validation sample*, from which $MPWE$ can be calculated. Figure 21 depicts the $(25\%, 50\%, 75\%)$ quantiles of the error for the $\log_{10}(LB)$ grid. We observe in that figure that the quantile curves show a general convex pattern with a global minimum (after some smoothing to reduce the noise). We also observe, for all three curves the minimum has turned out to be close to $1 = \log_{10}(10)$, which is the LB from which the curves where simulated. The minimum is generally less than 1 which may be partially due to the fact that not every generated curve will obtain the maximum possible LB (since we sample the slope magnitudes from $[m/2, m]$ uniformly). Once we have an estimate of LB found in this way, we can use the LB-BD curve to also find the corresponding BD. The choice of k is important and we discuss that in the cross-validation method later where the same issue arises. We observed that in this case for various size of the training sample size the LB turned out to be close to the LB from which we simulated the data. This does

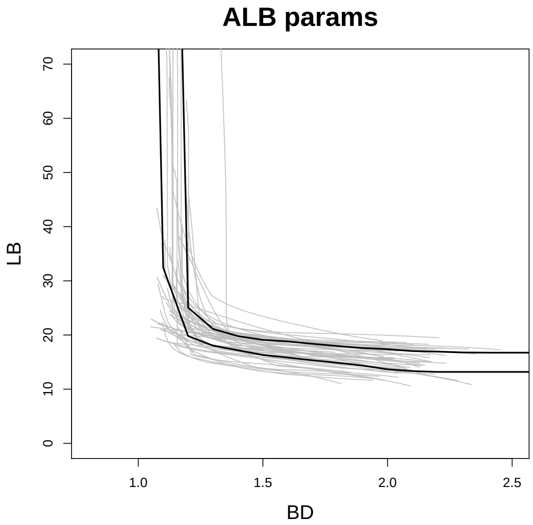

Fig. 20: LB-BD curves (grey) from 50 instances of series with: 8 break points; LB equal to 20; BD equal to 1. Black curves are 95% quantiles of the curves. We observe despite some variation the LB-BD curve for various simulations are close.

not have to be the case in general and the reason that this is the case here is the BD was sampled independently over time. If the BD has some pattern – for example if it is autoregressive – then a larger m (and smaller BD) than the simulated m may be better for prediction when sample size is large. A full discussion of this needs a careful study and we leave that for future research.

The above example can be modified to include many other practical situations. While it may not be the case that multiple instances of the same process are available, we may have access to the data from processes with very similar behavior. For example we may want to approximate the temporal air pollution data in a location with only three data points during the year, while we have access to the complete data for other locations with different seasonal patterns but similar physical/chemical properties so that the rate of change in the process for the two locations are similar. We provide more details about a specific case in Example 3 of the Application Section.

8.3 Cross validation

This subsection discusses a cross-validation method in order to pick the LB value when only data from one instance of the series is available in contrast to Example 1. First we discuss in Example 2 a case for which this method is applicable, while standard techniques of fitting may not be applicable due to data sparsity in some subintervals of the data.

Example 2 (Cross-validation Method):

Suppose $\{Y(t)\},\ t \in [a, b]$ is a slow-moving process for which some data are available with possible sparsity in some intervals and no clear seasonal patterns such that the standard smoothing methods or time series techniques cannot be applied. Despite this, the rate of the change of the

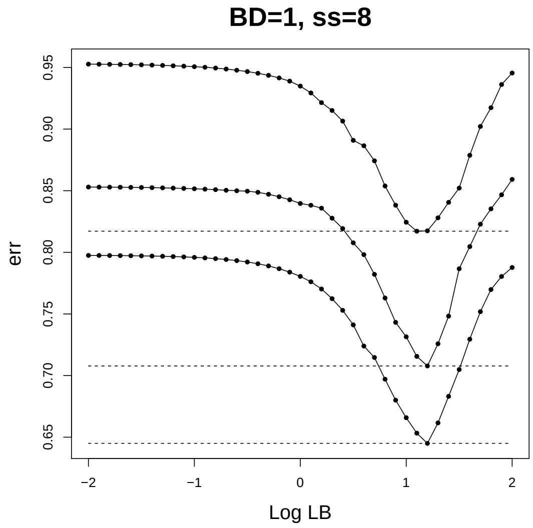

Fig. 21: The MPWE error (y-axis) quantile curves created using complete data from many instances of a series with $\log_{10}(LB) = 1$ and $BD = 1$. For each instance of the series, we pick 8 training points and fit many curves using various $\log_{10}(LB)$ values (x-axis) and calculate the error in predicting the validation sample for each $\log_{10}(LB)$ value.

process over time as measured by LB might be relatively small up to a reasonable BD and we can apply the $Lipfit$ method. We need to develop a method for finding an appropriate LB-BD pair to apply the $Lipfit$ method and calculate the errors associated with the prediction. Again we can apply the Prediction Error Minimization Method (PEM), we discussed before to the estimated LB-BD curve, calculated from data. Below we discuss a cross-validation method in order to pick an appropriate LB-BD pair in such cases.

In order to build the cross-validation algorithm, we perform a simulation study. We simulate a curve with 5 break points on [0,1] with the condition that the distance of each pair of break points is at least $1/7$. We let $m = 10$. At each break point a slope magnitude is drawn at random from $[m/2, m]$ and a random sign is sampled for the slope. Then we add independent deviations sampled from $[-0.5, 0.5]$ to each point to get a function in $\mathcal{ALB}(m = 10, \sigma = 0.5)$. Then we create a data set by sampling $n = 200$ random points in $[0, 1]$ and use this data set for developing the cross-validation algorithm. In order to perform the cross-validation, we choose the training sample size for example $k = 5$. Then for $N = 1000$ times: we choose sub samples of our data of size k to fit the data using $Lipfit$ with various $l = \log(m)$ values taken from an interval $L = [l_1, l_2]$; predict the rest of the points and calculate the $MPWE$. For each $l \in L$, we calculate the mean of all N errors and plot the CV error against the l as shown in Figure 22 for $k = 2, 5, 10, 100$. For each k, we can view the cross-validation function as a function:

$$CV : [l_1, l_2] \to \mathbb{R}^{\geq 0},$$

which maps a $l = \log(m)$ value to a non-negative real number. In Figure 22 and for various k – except for some wiggly behavior – we observe two general patterns: (i) The function CV is convex with a global minimum. (ii) The function CV is almost constant at first and rapidly takes off at some point (the point with maximum 2nd derivative). The point for which the CV function attains a minimum or the point for which the CV function attains the largest 2nd derivative are candidates for the chosen $l = \log(m)$. But first we need to omit the wiggly behavior and then find the point for which minimum is obtained (when it exist) and the point for which the 2nd derivative is maximized. We use the following algorithm.

Smoothing for optimal point detection:

- Receive as input a gridded function CV defined on $[l_1, l_2]$
- Calculate the number of changes in sign of the derivative of the function calculated from the gridded data.
- While the number of the derivative changes is larger than 1, perform a filtering to smooth the function.
- Consider the final function for which the number of the derivative sign changes is either 0 or 1.
- Calculate the 2nd derivative on the grid for the function obtained.
- If the number of the sign changes is 1 then find the point for which the minimum is obtained and return that otherwise return "NA" to denote not available.
- Calculate the point for which the 1st derivative is non-negative and the 2nd derivative is maximized (for both cases) and also return that.

Figure 23 shows the result of the above smoothing to the cross-validation functions for a data set of size 200 and Table 5 presents the calculated values. We observe in the figure that for $k = 2, 5, 10$ a smooth curve with a global minimum is obtained (which correspond to the 1 sign change in the derivative) while for $k = 100$ an increasing curve is obtained (which corresponds to 0 sign change). In the table we observe that both argmin(CV) and armax($CV^{''}$) are close to the $\log_{10}(LB) = 1$. Note that 5 slope magnitude for the curve were sampled from $[LB/2, LB]$ and therefore the LB slope is not necessarily obtained by the curve. The 2nd derivative is maximized at the global minimum when it is not "NA" (Not Available). Therefore argmin(CV) and armax($CV^{''}$) point at the same value. When the minimum does not exist the armax($CV^{''}$) for points with non-negative derivative provide a reasonable value as well.

Figure 24 and Table 6 provide the result of the above algorithm to a data set of size 32 and with training sizes $k = 2, 5, 8, 16$. We observe similar results despite a much smaller data size. Also in this case all the curves do attain a global minimum. In general it seems that smaller training sample size attain a global minimum while very large training sizes may end up in no change in the derivative sign. Our general recommendation is to perform the cross-validation for a dataset of size n for various values for example $k = 2, 5, n/8, n/4, n/2$ and inspect the CV curves and smoothed CV curves. For an algorithmic solution to pick the optimal LB, here we restrict ourselves to the curves for which a global minimum is obtained and the 2nd derivative is maximized at the minimum. More research and comparison is needed in picking the optimal LB which is out of the scope of this paper.

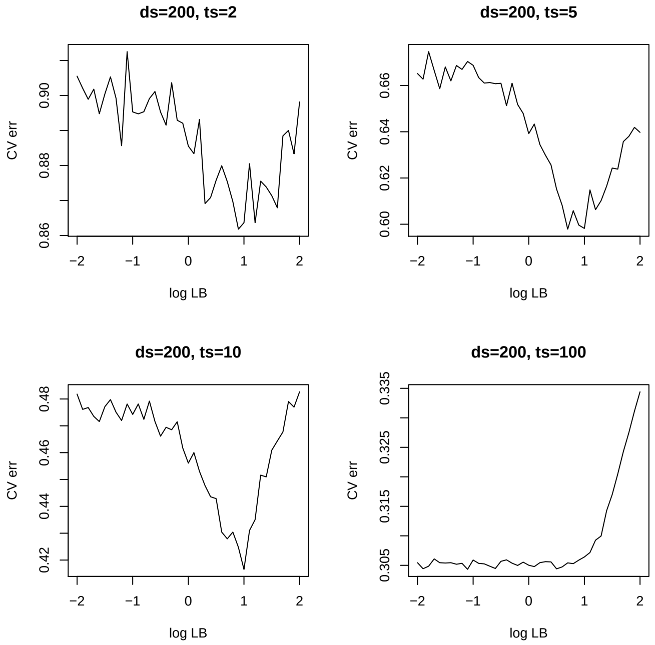

Fig. 22: The cross validation error (y-axis) is plotted versus $\log(LB)$ (x-axis) for a data set of size 200 for various size of training sample size $k = 2, 5, 10, 100$.

Once we have found an appropriate LB, the LB-BD function can be used to find BD. For the examples above, if we choose the LB from a curve with a global minimum and with the largest 2nd derivative, we find BD=0.56 for data size 200 and 0.61 for data set of size 32.

Tab. 5: Picking a LB-BD pair by cross-validation when data size is 200.

training size	smoothing num	argmin(CV)	min(CV)	armax($CV^{''}$)	max($CV^{''}$)
2	49	1	1	1.00	0.16
5	25	0.80	0.55	0.80	0.36
10	16	0.80	0.37	0.80	0.27
100	17	NA	NA	0.700	0.014

Tab. 6: Picking a LB-BD pair by cross-validation when data size is 32.

training size	smoothing num	argmin(CV)	min(CV)	armax($CV^{''}$)	max($CV^{''}$)
2	15	0.6	1.0	0.600	0.094
5	44	0.90	0.61	0.90	0.11
8	4	0.80	0.42	0.80	0.44
16	25	0.70	0.31	0.70	0.09

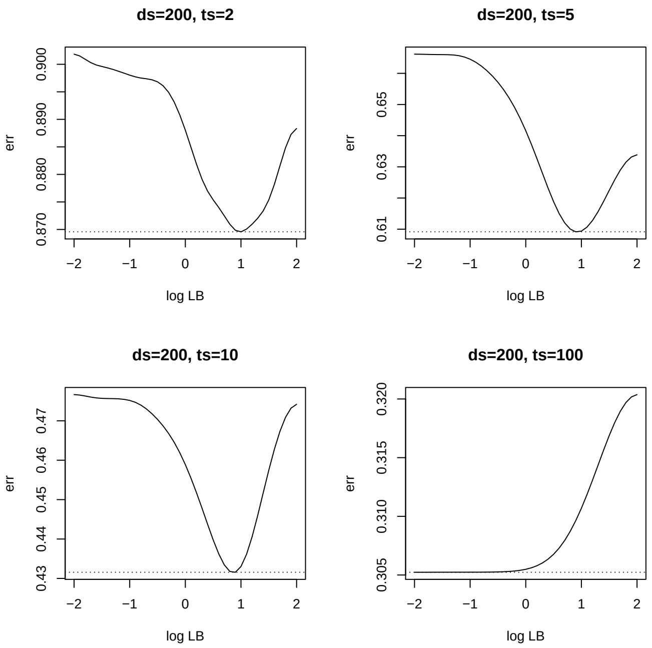

Fig. 23: The smoothed versions of CV plots for optimal point detection for a data set of size 200 and various training sample size $k = 2, 5, 10, 100$.

9 Application to air pollution data

This section applies the methods developed in this work to air pollution data.

We are interested in approximating the biweekly averaged air pollution (Ozone) process in homes and schools in Southern California during 2005, using three biweekly measurements in the spring, summer and winter. The moving average process refers to a process for which the value of the process on each day is the average of the process in 15 days centered around that day. (For data collection in the study, measurement filters are placed in the school for two-week periods to collect aggregated air pollution levels.) We also have access to 11 central sites for which complete data are available during 2004–2007. To be more concrete denote the biweekly averaged pollution process by $Y(s,t)$ at the location s for which three times during 2005 are available: $Y(s,t_1), Y(s,t_2), Y(s,t_3)$. We denote the 11 central site locations by $s_1, s_2, \cdots, s_{11}$. Figure 26, bottom right panel, shows the data for one of the central stations, Upland (UPL). The other panels show the application of the methods we developed in this paper including estimating the LB-BD curve and the cross-validation method for picking one LB-BD pair. The curve corresponding to this LB-BD pair for training size (ts) equal to 3 is plotted over the data in the bottom right panel. This is just for illustration of cross-validation method as we use PEM for our data analysis.

Figure 27 depicts the calculated LB-BD functions for the 11 communities in grey for both $\mathcal{ALB}$ and $\mathcal{PALB}$ families as well as the 25% and 75% quantile curves. There is good consistency among the curves across the central sites, except for Santa Maria for which the curve is visibly

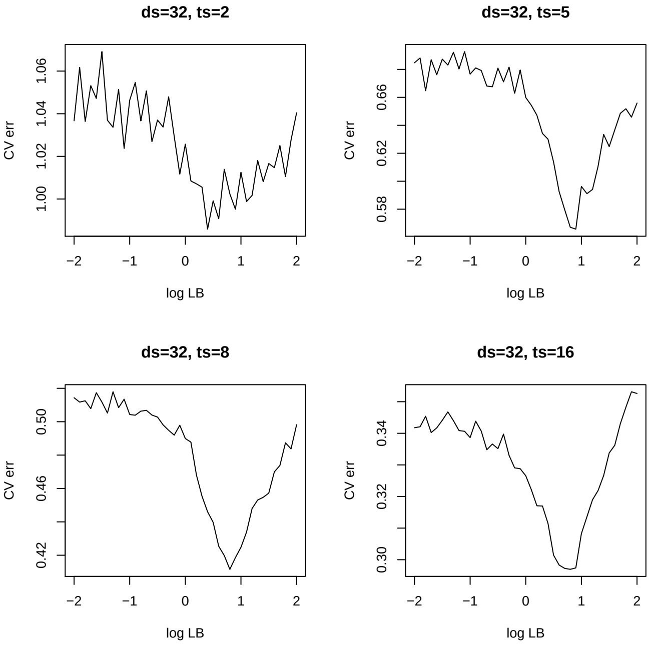

Fig. 24: The cross validation error (y-axis) is plotted versus $\log(LB)$ (x-axis) for a data set of size 32 for various size of training sample size $k = 2, 5, 8, 16$.

placed below all the other curves for both families. This is because Santa Maria is a much cleaner community with lower levels of Ozone and its variation across the year. This figure suggests that if we use the LB-BD curve from one location for a location which is not too far or too different from the location of interest then the results will be reliable.

Here we describe two scenarios one may be interested in for prediction: (1) We assume that the new location s does not belong to any of the communities and therefore does not have a very similar weather pattern to any central site with complete data. (2) We assume that the station belongs to one of the communities and therefore there is a rather close central site but there is no guarantee that the seasonal patterns completely match. Also assume in (1) there are no observations in nearby locations with similar weather patterns to borrow strength across space to build a complex statistical model; in (2) there is a nearby station with complete data across the year, but there is no guarantee the seasonal patterns of the air pollution process in the two locations are exactly the same. In situation (1), as discussed above, we can choose nearby locations with complete data and then apply the methods described above to find appropriate LB-BD for those locations in order to use for the location with incomplete data. Note that all we need is a bound on m and σ and therefore we can use slightly larger m, σ from what we have found, if there is more doubt about the similarity of the other locations in terms LB-BD.

In situation (2), suppose the closest central site location is s_i. Then we can calculate the difference process: $D(t_j) = Y(s, t_j) - Y(s_i, t_j)$. If the seasonal patterns are the same, we must have $D(t_1) \approx D(t_2) \approx D(t_3)$. Note that this does not guarantee that the seasonal patterns are the

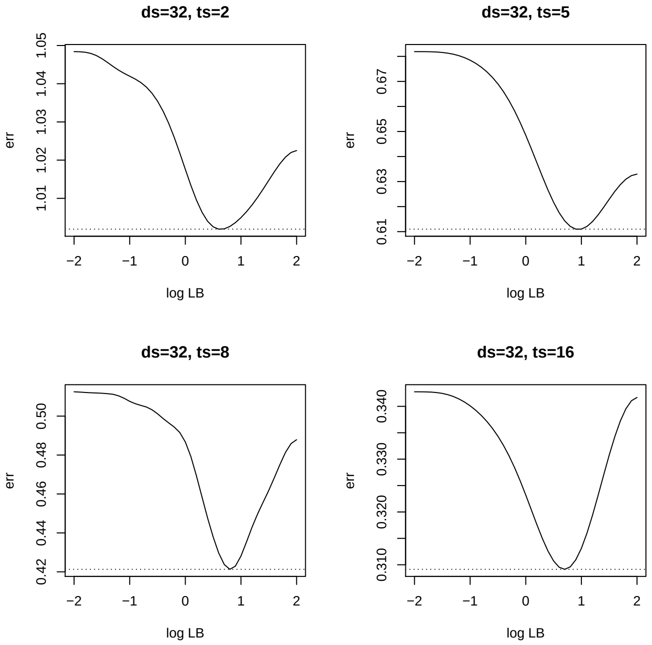

Fig. 25: The smoothed versions of CV plots for optimal point detection for a data set of size 32 and various training sample size $k = 2, 5, 8, 16$.

same and we are obliged to rely also on some expert knowledge. However if such a knowledge is available one may suggest to take the average of the differences $D = \sum_{j=1}^{3} D(t_j)/3$ and then return $Y(s_i, t) - D$ as an approximation of $Y(s, t)$ for all t during 2005. Instead of averaging $D(t_j)$'s: we approximate the difference process $D(t)$ using $Lipfit$ to get a complete series $\hat{D}(t)$ and then return $Y(s_i, t) - \hat{D}(t)$ as our approximation of $Y(s, t)$. Thus we accommodate the possibility that the difference between the two processes $Y(s, t)$ and $Y(s_i, t)$ varies over time and $Lipfit$ tries to capture that variation.

To give example for scenarios (1) and (2), we choose three central sites in Upland (UPL), Long Beach (LGB) and Anaheim (ANA). We focus on approximating the curves for UPL and LGB. For scenario (1) we calculate the LB-BD curve for UPL, LGB and for scenario (2) we calculate the LB-BD curve for the difference of the two processes with ANA. Then for each scenario and for both $\mathcal{ALB}$ and $\mathcal{PALB}$, we use the Prediction Error Minimization method (PEM) to pick a LB-BD pair.

Table 7 presents the results of calculating the minimal error, Υ, to the Ozone process during 2005 and in the locations Long Beach (LGB) and Upland (UPL) and the difference processes of these two locations with Anaheim (ANA). For each case the LB-BD curve is calculated using the data and then the minimal error for each of (IE, DIE); $(SPWE, DSPWE[LI], DSPWE[Lipfit])$ when three data points are available are reported in the table. The pair (BD, LB) for which the error is minimized is also reported. We have done this for both $\mathcal{ALB}$ and $\mathcal{PALB}$ families. The results observed in the table are as follows: $\mathcal{PALB}$ method generally works better for these data due to the approximate periodicity of Ozone process; the data-informed errors $(DIE, DSPWE)$

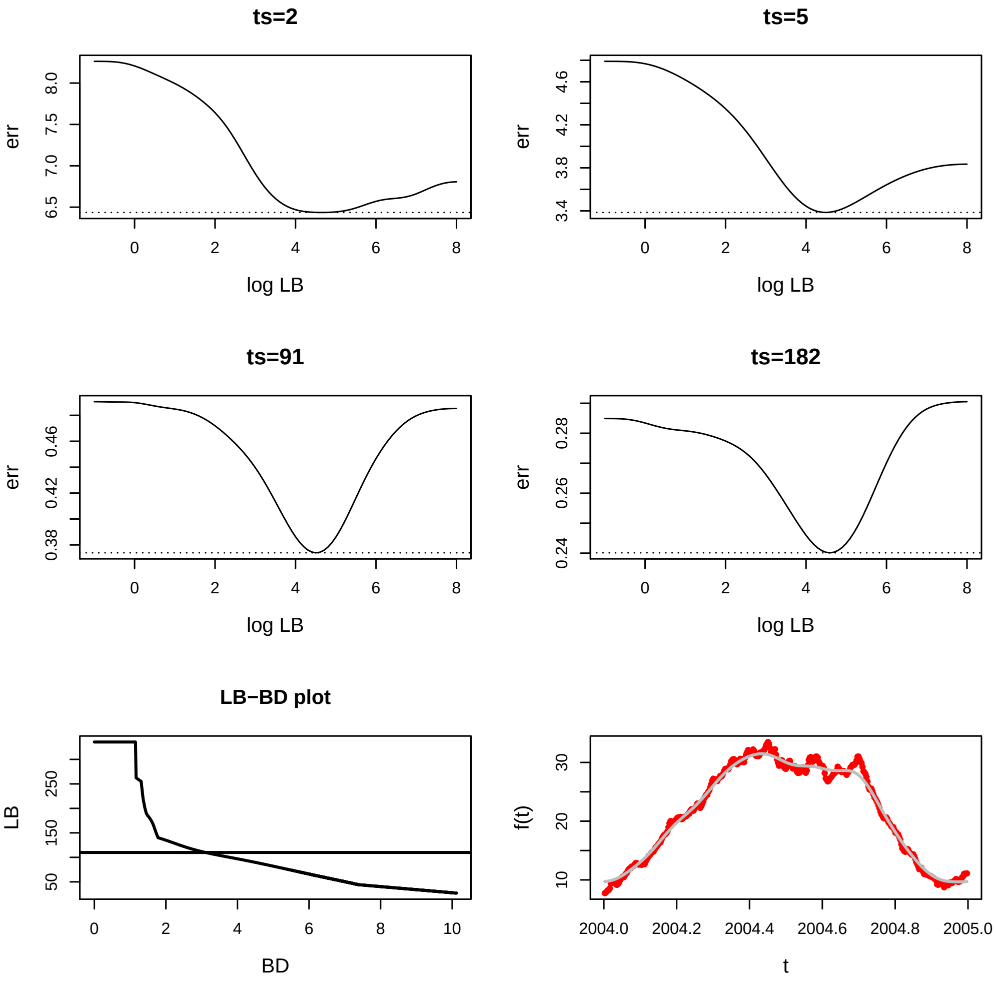

Fig. 26: The application of the cross-validation method along with LB-BD plot to find LB and BD for UPL.

are considerably smaller than their non-informed versions $(IE, SPWE)$ in some cases, showing that developing and using the data-informed errors is worthwhile; the $Lipfit$ method has outperformed LI in some cases and is never inferior to LI.

Figure 26 shows the application of the cross-validation method to finding LB and BD for UPL.

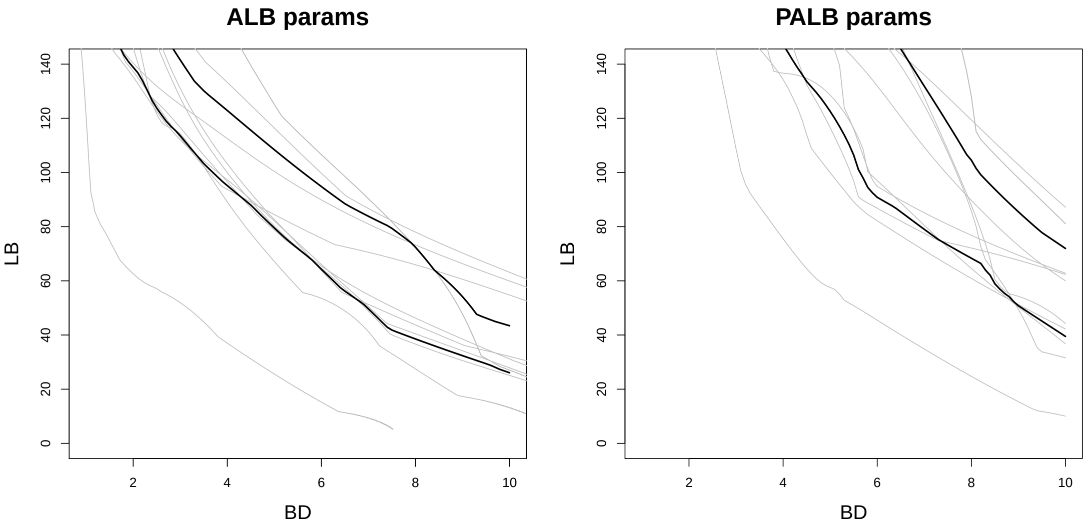

Fig. 27: LB-BD functions for the temporal biweekly Ozone process for 11 central stations in Southern California.

Tab. 7: The errors $(IE, DIE); (SPWE, DSPWE[LI], DSPWE[Lipfit])$, when three data points are available.

	$\mathcal{ALB}$		$\mathcal{PALB}$	
Process	errors	(LB, BD)	errors	(LB, BD)
LGB	(12, 10); (15, 15, 15)	(36, 9)	(13, 8.1); (15, 15, 13)	(29, 11)
UPL	(13, 9.9); (15, 15, 15)	(17, 12)	(13, 7.1); (14, 14, 14)	(17,12)
ANA-LGB	(6.6, 5.2); (8.3, 8.3, 8.3)	(20, 5)	(7.1, 4.2); (8.4, 7.1, 6.5)	(16, 5.7)
ANA-UPL	(6.4, 4.6); (6.9, 6.9, 6.9)	(5.5, 6.0)	(6.6, 3.9); (7, 6.1, 5.1)	(5.4, 6.1)

10 Discussion and future directions

This work developed a framework for fitting functions with sparse data. At first we considered a framework based on measuring the variation of the functions by Lipschitz Bound, also considered by Sukharev (1978) and Beliakov (2006). The limitation in using such a framework is due to the fact that many processes in practice, despite revealing a slow global variation, have some smaller scale variations which cause the Lipschitz Bound to be too large to be useful in fitting or calculating the prediction errors. Thus we extend this framework by measuring the Lipschitz Bound by allowing a Bound Deviation in the sense that we accept a number m as Lipschitz Bound up to a deviation σ if the function of interest can be approximated by another which accepts m as LB and does not deviate from the original function more than σ, in terms of sup norm. Using this framework, we can find reasonable fits and prediction errors for functions which do not admit a small enough

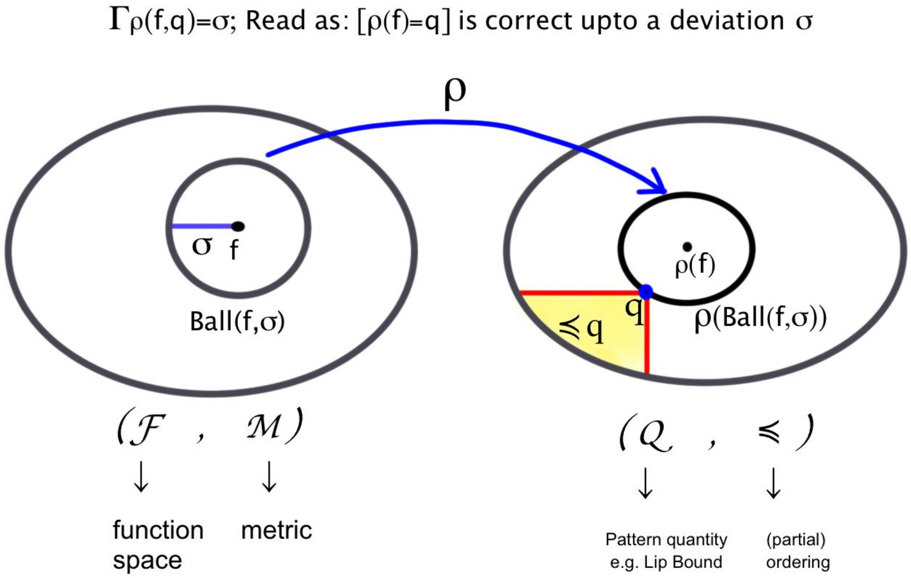
Γρ(f,q)=σ; Read as: [ρ(f)=q] is correct upto a deviation σ
ρ
σ
f
Ball(f,σ)
ρ(f)
q
≼q
ρ(Ball(f,σ))
(F , M)
function space
metric
(Q , ≼)
Pattern quantity e.g. Lip Bound
(partial) ordering

Fig. 28: The variation-deviation function in a generalized framework.

Lipschitz Bound. We also provided validation and cross-validation methods to pick appropriate LB-BD curve from data to fit the data using the $Lipfit$ method or calculate the prediction errors.

Another key idea we introduced is the formalization of the trade-off between the variation measure (LB here) and the deviation measure (BD as measured by sup norm here) which is summarized in a non-increasing convex curve – LB-BD function (curve). We provided convex optimization methods to calculate the LB-BD curve using data or gridded versions of the functions under study and provided the connection of the LB-BD curve of a gridded function to its more fine-resolution version. Given the LB-BD curve for a function, we develop a method to find an appropriate LB-BD pair to apply $Lipfit$ – a pair which depended on the data. Given the LB-BD curve, we also calculated the minimal prediction error, e.g. $DSPWE$, by minimizing it across the LB-BD curve. In the background section, we made some connection between this work and some smoothing methods such as smoothing splines. In fact the smoothing spline method merely used a different variation measure $\int_D ||f''(x)||dx$ and deviation measure: the sum of the square of the difference between the observed data and fitted. This immediately leads to the idea of generalizing this framework by choosing various variation and deviation measures and define a variation-deviation curve (an extension of LB-BD curve) for each case. To that end suppose:

- $\mathcal{F}$ is a function family equipped with a metric $\mathcal{M}$ which is used to define deviation of one function to another.
- $\rho : \mathcal{F} \to \mathcal{Q}$ is a variation measure from $\mathcal{F}$ to $\mathcal{Q}$. In this work we used LB as the variation measure and $\mathcal{Q} = \mathbb{R}^{\geq 0}$. We no longer assume $\mathcal{Q}$ to be one dimensional, but we still assume it is partially ordered, for example $\mathcal{Q} = \mathbb{R}^2$.

The diagram in Figure 28 depicts the idea to generalize this framework. We denote the variation-deviation curve in general by $\gamma_f(q) = \Gamma_\rho(f,q) = \sigma$, to emphasize the dependence on ρ. We define $\Gamma_\rho(f,q)$ to be the smallest σ so that the image of the unit ball of radius σ, $Ball(f,\sigma)$, under ρ includes an element in $\mathcal{Q}$ less than q. A multidimensional $\mathcal{Q}$ is useful for example in: (1) a spatial problem where we need to allow different variation bounds in different directions. Note that in this case the variation-deviation function $\Gamma_\rho(f,.) : \mathcal{Q} \to \mathbb{R}^2$, has multidimensional domain. (2) When we like to control various measures of variation for example both the first and second derivative. We leave a through investigation of this general framework to future research.

Some other important extensions and open problems are: (1) Given a curve decide if the curve can be LB-BD curve for a function; such a curve must be non-increasing and convex as we show in this work, but is that enough? (2) If we consider a random process $\{Y(t)\}, t \in T$ for some space T (time, space, spatial-temporal), for each instance of this process we can calculate an LB-BD curve, thus LB-BD curve is a random quantity. In this work we assumed that this random quantity does not vary much from one instance to another; or it is applicable from one time interval to another; or if we use the LB-BD curve of a comparable data set (for example a nearby station) the results are reliable. In fact using some simulations and real air pollution data we showed that this can be the case. However, it is interesting to investigate the LB-BD variability for random processes and it may be even useful to develop parametric and non-parametric models for LB-BD curve as a random quantity. We also leave these problems for future research. (3) In this we work, we developed the Prediction Error Minimization (PEM) to pick a LB-BD curve to minimize the error of interest for example $DSPWE$ over the domain of interest, given data. Then we used that pair for applying $Lipfit$ and calculating the prediction errors. The restriction of this method lies in the fact that we used the same LB-BD pair to approximate all points. Alternatively we can allow picking a different LB-BD pair for each given data point x, a method which in general can improve the pef at any given point at a cost of more computations.

11 Appendix: Proofs

Proof (Lemma 3.1)
(i) We give a proof for the 1-d case and a similar proof can be given for the multidimensional case. Suppose $a \leq x < y \leq y$, then by *Mean Value Theorem* from calculus, there exist z such that $x < z < y$ and

$$\frac{f(x) - f(y)}{x - y} = f'(z) \Rightarrow f(x) - f(y) = (x - y)f'(z) \Rightarrow |f(x) - f(y)| \leq m|x - y|.$$

(ii) Suppose $x, y \in D$. Consider the line segment joining x and y which is in D by the convexity assumption. Then the line segment will intersect with a number of D_j. We can define a sequence $x = b_1, \cdots, y = b_k$ where (b_i, b_{i+1}) is inside a D_{i_j} for some i_j. Then

$$|f(x) - f(y)| \leq |f(b_2) - f(x)| + |f(b_3) - f(b_2)| + \cdots + |f(y) - f(b_{k-1})|$$

$$\leq m[||b_2 - x|| + ||b_3 - b_2|| + \cdots + ||y - b_{k-1}||)] = m||x - y||.$$

Note that we have used the continuity assumption of f in the last step to insure that even if b_i or b_{i+1} are on the boundary of D_{i_j} the Lipschitz property holds.

(iii) Convexity:
Suppose $f_1, f_2 \in \mathcal{LB}(m)$ and $0 \leq \theta \leq 1$. Then if $f = \theta f_1 + (1-\theta) f_2$:

$$\begin{aligned} \forall x, y \in D,\ |f(x) - f(y)| &= |\theta(f_1(x) - f_1(y)) + (1-\theta)(f_2(x) - f_2(y))| \\ &\leq \theta |f_1(x) - f_1(y)| + (1-\theta)|f_2(x) - f_2(y)| \\ &\leq \theta m||x-y|| + (1-\theta) m ||x-y|| = m||x-y||. \end{aligned}$$

Closed:
Suppose $f_n,\ n = 1, 2, \cdots$ is a sequence in $\mathcal{LB}(D, m)$ and $f_n \to f$ when $n \to +\infty$ with respect to the sup norm. Then we need to prove $f \in \mathcal{LB}(D, m)$ as well. Fix $x_0, y_0 \in D$. Then for any $\epsilon > 0$ there exist N such that $n > N$, $\|f - f_n\| \leq \epsilon$ and we conclude:

$$|f(x_0) - f(y_0)| \leq |f(x_0) - f_n(x_0)| + |f_n(x_0) - f_n(y_0)| + |f(y_0) - f_n(y_0)| \leq 2\epsilon + m||x_0 - y_0||.$$

Since above holds for any $\epsilon > 0$ we have shown that $|f(x_0) - f(y_0)| \leq m||x_0 - y_0||$ and the proof is completed.
(iv) The convexity is straight-forward from the definition. To see that it is not closed, let $c = (a+b)/2$ be the middle point and consider a function h_n :

$$h_n(x) = \begin{cases} mx^2/2 + (m\delta - m\delta^2/2), & x \in [a,b] \cap [c - 1/n, c + 1/n] \\ m|x|, & x \in [a,b] - [c - 1/n, c + 1/n], \end{cases}$$

which is in $\mathcal{DIF}[(a,b), m]$ because the derivative and value of h matches in the two cases at the boundary and the derivative is bounded by m. However $h_n \to h$ where $h(x) = m|x - c|$, which is not differentiable.
(v) Suppose that is not true. Then $\exists x, y \in D$ such that $|f(x) - f(y)| > Lip(f)|x - y|$. By properties of real numbers there exists $m < Lip(f)$ such that $|f(x) - f(y)| > m|x - y|$ which is a contradiction to the definition of $Lip(f)$.
(vi) Suppose $f = \theta f_1 + (1-\theta) f_2$ for $\theta \in (0,1)$. Then for any $x, y \in D$

$$|f(x) - f(y)| \leq \theta |f_1(x) - f_1(y)| + (1-\theta)|f_2(x) - f_2(y)| \leq (\theta L(f_1) + (1-\theta) L(f_2))|x - y|.$$

Therefore $f \in \mathcal{LB}(m)$, where $m = (\theta L(f_1) + (1-\theta) L(f_2))$ and the proof is complete since $Lip(f)$ is the infimum m for which $f \in \mathcal{LB}(m)$.
(vii) For the general multi-dimensional $D \subset \mathbb{R}^d$ (even more generally for separable Riemannian manifolds), the result holds, Azagra et al. (2007). Here we present a simple proof for the 1-dimensional case as well.

We show that $\mathcal{DIF}([a,b], m)$ is dense in $\mathcal{LB}([a,b], m)$ and the periodic case is similar. Since $\mathcal{PL}([a,b], m)$ is dense in $\mathcal{LB}([a,b], m)$, it is enough to show that any element of $\mathcal{PL}$ is approximated by an element of $\mathcal{DIF}([a,b], m)$ with any given precision $\epsilon > 0$ with respect to the sup norm. Since any function $f \in \mathcal{PL}[(a,b), m]$ is differentiable everywhere except for the break points, we will construct a function that agrees with f except for small neighborhoods of a finite number of break points. For simplicity assume that f has only one break point in the middle point $c = (a+b)/2,\ f(x) = m|x - c|$. Then we can define h_n :

$$h_n(x) = \begin{cases} mx^2/2 + (m\delta - m\delta^2/2) & x \in [a,b] \cap [c - 1/n, c + 1/n] \\ m|x| & x \in [a,b] - [c - 1/n, c + 1/n] \end{cases}$$

which is in $\mathcal{DIF}[(a,b),m]$ because the derivative and value of h matches in the two cases at the boundary of the cases and the derivative is bounded by m.

■

Proof (Lemma 3.2)
Consider Figure 4 and suppose the target function is observed on $C = (x_1, y_1)$. From C draw two lines with slopes $m, -m$ and extend them until x_A, x_B. Now define three functions f_0, f_1, f_2 as follows: let f_1 be the function with trajectory consisting of line segments A_1C, CB_1; let f_2 be the function with trajectory A_2C, CB_2; and finally let $f_0(x) = y_1, \forall x \in [a,b]$. Then note that $f_1, f_2, f_0 \in \mathcal{LB}(m)$ and any other function $f \in \mathcal{LB}(m)$ satisfies the property that $f_1(x) \leq f(x) \leq f_2(x)$. Moreover for any $x \in [a,b]$ and $f_1(x) \leq y \leq f_2(x)$, there exist a function $f \in LB(m)$ such that $f(x) = y$. We construct one such function by considering the line that goes through (x_C, y_C) to (x,y). Therefore for each x the largest possible value is attained by $f_1(x)$ and the smallest possible value is attained by $f_2(x)$. Then the value y that minimizes this maximum is optimal:

$$\max\{|y - f_1(x)|, |y - f_2(x)|\}.$$

The above is minimized uniquely by letting $y = \frac{f_1(x)+f_2(x)}{2} = f_0(x) = y_1$. In other words we have shown that the constant function $f_0(x) = y_1$ minimizes pef at each point and therefore it is optimal in terms of $DSPWE$ and $DMPWE$. These errors can also be obtained easily using Figure 4. Also similar argument can be used to show that the constant function minimizes DIE as its integral, I, is the unique value that minimizes

$$\max\{|I - \int_a^b f_1(x)dx|, |I - \int_a^b f_2(x)dx|\}.$$

■

Proof (Lemma 3.3)
Suppose the function $f_1 : [x_A, x_B] \to \mathbb{R}$ has trace along AC, CB and $f_2 : [x_A, x_B] \to \mathbb{R}$ has trace along AD, DB. Then $f_1, f_2 \in \mathcal{LB}(m)$. Note that BC and AD are parallel. With some elementary algebra we can find the coordinates of C, D, F, G. In particular

$$x_C = \frac{a_1+b_1}{2} + \frac{b_2-a_2}{2m},\ y_C = \frac{b_2+a_2}{2} + m\frac{b_1-a_1}{2}$$

$$x_D = \frac{a_1+b_1}{2} - \frac{b_2-a_2}{2m},\ y_C = \frac{b_2+a_2}{2} - m\frac{b_1-a_1}{2}$$

Hence the length of $|CG| = |FD|$ is given by

$$(y_F - y_D) = y_A - y_D = (x_F - x_A)m - y_D = \frac{m}{2}(b_1 - a_1) - \frac{b_2 - a_2}{2}.$$

We also know that $f_2 - f_1 = 2|CG| = 2|FD|$ on (x_F, x_G). Hence any $approx[f]$ on (x_F, x_G) is off from either one of f_1 or f_2 by a distance at least of FD. Now note that $Lipfit[f]$ is exactly in the middle of f_1, f_2 and minimizes the max distance to either of f_1, f_2, a distance bounded by $e =: |FD| = \frac{m}{2}(b_1 - a_1) - \frac{b_2-a_2}{2}$. Moreover any other function g in $\mathcal{LB}(m)$ that goes through A, B

must be within $ACBD$ and hence is off from $Lipfit[f]$ by at most of a distance e. ■

Proof (Theorem 3.3)
Figure 6 shows the situation with two observed points at $A = (x_A, y_A)$ and $B = (x_B, y_B)$. From each of A and B, we draw two lines with slopes $m, -m$ and call the intersection points C, D. Now consider three functions on $[x_A, x_B]$ f_1, f_2, f_0 by defining f_0 to be the function with trajectory A, B; f_1 the function with trajectory AC, CB; and f_2 the function with trajectory AD, DB. By Lipschitz assumption any function $f \in \mathcal{LB}(m)$ which goes through A and B must fall inside the parallelogram $ACBD$. Therefore the largest integral value is taken by f_1 and the smallest is taken by f_2. Also note that $f_1, f_2 \in \mathcal{LB}(m)$. Suppose f minimizes the integral error and let $\int_{x_A}^{x_B} f(t)dt = I, \int_{x_A}^{x_B} f_1(t)dt = I_1, \int_{x_A}^{x_B} f_2(t)dt = I_2$. Then I minimizes the error if and only if it minimizes

$$\max\{|I - I_1|, |I - I_2|\}.$$

The solution to the above problem is $I = \frac{I_1+I_2}{2}$. From Figure 6 we observe that $I_0 = \int_{x_A}^{x_A} f_0(t)dt$ satisfies this property by symmetry. Therefore the optimal integral approximation is equal to $\int_{x_A}^{x_A} f_0(t)dt = \frac{y_A+y_B}{2}(x_B - x_A)$. ■

Proof (Theorem 3.4)
Figure 6 shows the situation with two observed points at $A = (x_A, y_A)$ and $B = (x_B, y_B)$. Denote the slope of the line AB by $m^\star$. Now consider three functions on $[x_A, x_B]$ f_1, f_2, f_0 by defining f_0 to be the function with trajectory A, B; f_1 the function with trajectory AC, CB; and f_2 the function with trajectory AD, DB.
Since f_0 is the optimal solution to minimizing the integral and due to symmetry, to calculate the error we only need to calculate the area of the triangle ACB.
First note that the length of AB is equal to

$$l := \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = \sqrt{1 + m^{\star 2}}|x_B - x_A|.$$

To find the area of the triangle it remains to find the distance from C to AB – which we denote by h – since then $Area(ABC) = 1/2hl$. First we find C by finding the intersection point of AC and BC:

$$C = (\frac{x_A + x_B}{2} + \frac{y_B - y_A}{2m}, \frac{y_B + y_A}{2} + \frac{m(x_B - x_A)}{2}).$$

Since the distance of a point (x_0, y_0) from a line $ax + by + c = 0$ is equal to $\frac{ax_0+by_0+c}{\sqrt{a^2+b^2}}$, we conclude the distance of C from AB is equal to:

$$\begin{aligned} h &= \frac{1}{1+m^{\star 2}}[\frac{y_B+y_A}{2}+m\frac{x_B-x_A}{2}-|m^\star|(\frac{x_A+x_B}{2}+\frac{y_B-y_A}{2m})]-y_A-|m^\star|x_A] \\ &= \frac{1}{1+m^{\star 2}}[\frac{y_B-y_A}{2}+m\frac{x_B-x_A}{2}-|m^\star|\frac{x_B-x_A}{2}-|m^\star|\frac{y_B-y_A}{2m}] \\ &= \frac{1}{2(1+m^{\star 2})}[(m-m^\star)\{\frac{y_B-y_A}{m}+(x_B-x_A)\}] \\ &= \frac{1}{2(1+m^{\star 2})}[(x_B-x_A)\frac{|m^\star|}{m}+(x_B-x_A)] \\ &= \frac{1}{2(1+m^{\star 2})}(1+\frac{|m^\star|}{m})(x_B-x_A). \end{aligned}$$

Therefore the area is equal to

$$\frac{hl}{2}=\frac{1}{2(1+m^{\star 2})}(1+\frac{|m^\star|}{m})(x_B-x_A)\sqrt{1+m^{\star 2}}(x_B-x_A)=$$

$$\frac{(m^2-m^{\star 2})}{4m}(x_B-x_A)^2.$$

■

Proof (Theorem 3.5)
Before proving (a) to (c), we calculate the length of AF which we denote by Δ. For that using Figure 5, we write down the length of (x_B-x_A) as follows

$$(x_B-x_A)=|AF|+|FG||\cos(\alpha)|+|GB|=2|AF|+|FG||\cos(\alpha)|,$$

where $\alpha=\arctan(m)$. We also have:

$$|y_B-y_A|=|FG||\sin(\alpha)|.$$

Replacing $|FG|$ in the first equation we get:

$$(x_B-x_A)=2|AF|+|\frac{y_B-y_A}{m}|,$$

which gives

$$|AF|=1/2[(x_B-x_A)-|\frac{y_B-y_A}{m}|]$$

(a) Using Figure 5 the distance is maximized with the dashed lines in the middle of the interval $[x_A,x_B]$ and by defining new coordinates with A as the new origin, this is between the point $(\frac{x_B-x_A}{2},0)$ and $(\frac{x_B-x_A}{2},m\frac{x_B-x_A}{2})$. Therefore the bound is $|m\frac{x_B-x_A}{2}|$.

(b) Using Figure 5 the distance is maximized at x_C, between the point C and point H with the same x-value on the line AB. The line AC and AH have slopes m and $m^\star$ respectively and diverge. Therefore the distance between C and H is $|AF|(|m|+|m^\star|)$.

(c) Using Figure 5 the maximum distance is equal to CF which is equal to $|AF||m|$.

■

Proof (Theorem 3.6)
Let

$$\begin{aligned} E_1 &= [a, x_1 + e_1), \\ E_2 &= (x_2 - e_1, x_2 + e_2), \cdots, \\ E_{n-1} &= (x_{n-1} - e_{n-2}, x_{n-1} + e_{n-1}), \\ E_n &= (x_n - e_{n-1}, b]. \end{aligned}$$

Then interpolate f on each interval E_i using *approx* as prescribed by the definition.
(a) This is immediate by the assumption.
(b) We only need to show this for two points problem and the solution extends to the general case. Suppose $A = (x_A, y_A = f(x_A))$ and $B = (x_B, y_B = f(y_B))$ are given and we want to interpolate the interval (x_A, x_B). Without loss of generality also assume $x_A < x_B$ and $y_A \leq y_B$. Then let $e = \frac{x_B - x_A}{2}$ and consider the midpoint $x_C = x_A + e$. Define $y_C = y_A + me$. $SPWL$ is bounded by me and we want to show this bound is sharp. Consider a function $f_1 : [x_A, x_B] \to \mathbb{R}$ to be the function with trace consisting of the line segments AC, CB. Obviously f_1 goes through A, B. We claim $f_1 \in \mathcal{LB}(m)$. First note that by definition, the slope of AB is m. It remains to show the slope of CB is also bounded by m in magnitude.

$$m_{CB} = \frac{y_B - y_C}{x_B - x_C}.$$

Now note that y_B is at most $y_A + 2me$ hence

$$m_{CB} = \frac{y_B - y_C}{x_B - x_C} = \frac{2me + y_A - y_C}{e} = \frac{me}{e} = m.$$

Also from our assumption that $y_A \leq y_B$ we have

$$m_{CB} \geq \frac{y_A - y_C}{x_B - x_C} = \frac{-me}{e} = -m.$$

We have shown $m_{CB} \leq m$.

The proof is complete by noting at x_C we have

$$|f_1(x_C) - \mathit{approx}[f](x_C)| = |y_A + me - y_A| = me.$$

■

Proof (Theorem 3.7)
Define

$$\begin{aligned} E_1 &= [x_1 - e_{n,0}, x_1 + e_1), \\ E_2 &= (x_2 - e_1, x_2 + e_2), \cdots, \\ E_{n-1} &= (x_{n-1} - e_{n-2}, x_{n-1} + e_{n-1}), \\ E_n &= (x_n - e_{n-1}, x_n + e_{n,0}]. \end{aligned}$$

Then interpolate f on each interval E_i using *approx* as prescribed by the definition. Now the result can be deduced from Theorem 3.6. ■

Proof (Theorem 4.3)
Case 1:
In this case $|m^\star| < m$ as shown in Figure 11 (Top Left Panel) and the proof is similar to the case with $BD = 0$.
Case 2:
Consider Figure 11 (Top Right Panel) for the proof.
Define

$$A_1 = (x_A, y_A - \sigma),\ A_2 = (x_A, y_A + \sigma),\ B_1 = (x_B, x_B - \sigma),\ B_2 = (x_B, x_B + \sigma),$$

so that A_1, A_2 have the same x-value as A and off by σ in the y-axis. Similarly B_1, B_2 have the same x-value as B and off by σ in the y-axis. From each of $A_1,\ A_2,\ B_1,\ B_2$ draw a line with slope equal to m then one of the lines starting from $A_1,\ A_2$ (in Figure 11, the line starting from A_2) will intersect the line segment B_1B_2 at a point we denote by C and one of the lines starting from $B_1,\ B_2$, (in Figure 11, the line starting from B_1), will intersect the line segment A_1A_2 and we call that point D. Then the parallelogram A_2CB_1D and call the midpoint of A_2C, F and the midpoint of B_1C, G. The $Lipfit$ approximation is given by FG.

For LI it is clear that the error is equal to σ since the supremum error is equal to $|AA_2| = |BB_2|$.
For finding the error for $Lipfit$ let $\delta = |AD| = |BC|$. The supremum error made by $Lipfit$ is equal to $DSPWE = |FD| = |A_2D|/2 = (\sigma + \delta)/2$. Therefore it suffice to find δ in order to find the error. For that we measure $|B_3C|$ in two ways:

$$(1)\ \ |B_3C| = |B_3B_1| + |B_1B| + |BC| = m\Delta_x + \sigma + \delta;$$

$$(2)\ \ |B_3C| = |AD| + m^\star\Delta_x + |CB| = 2]\delta + m^\star\Delta_x.$$

By letting (1) and (2) equal, we find $\delta = \sigma - \Delta_x(|m^\star - m|)$ and therefore

$$SPWE = (\sigma + \delta)/2 = \sigma - \frac{\Delta_x}{2}(|m^\star - m|).$$

Case 3:
Consider Figure 11 (Bottom Left Panel) for the proof.
For LI it is clear that the error is equal to σ since the supremum error is equal to $|AA_2| = |BB_2|$.
For finding the error for $Lipfit$ let $\delta = |AD| = |BC|$. The supremum error made by $Lipfit$ is equal to $DSPWE = |FD| = |A_2D|/2 = (\sigma - \delta)/2$. Therefore it suffice to find δ in order to find the error. For that we measure $|B_3B|$ in two ways:

$$(1),\ \ |B_3B| = |AA_2| + m\Delta_x + |CB| = \sigma + m\Delta_x + \delta;$$

$$(2),\ \ |B_3B| = m^\star\Delta_x.$$

By letting (1) and (2) equal, we find $\delta = \Delta_x(|m^\star - m|) - \sigma$ and therefore

$$SPWE = (\sigma + \delta)/2 = \sigma - \frac{\Delta_x}{2}(|m^\star - m|).$$

■

Proof Lemma 6.2.

(a) This is straightforward by from the properties of infimum.

(b) For $m = +\infty$, $f \in \mathcal{LB}(m)$ and therefore $\gamma_f(m) = 0$. Also only constant functions satisfy $m = 0$ and therefore $\gamma_f(m) = d/2$ as the constant function $g(x) = (\sup_{z\in[a,b]} f(z) - \inf_{z\in[a,b]} f(z))/2$ minimizes $SPWL(f, g)$.

(c) For $\sigma = +\infty$ any bounded function, g, satisfies $SPWL(f, g) \leq \sigma$ including any constant function $g = c$ for which we have $Lip(g) = 0$. The only function, g, which satisfies $SPWL(f, g) = 0$ is f and therefore $\gamma^{-1}(0) = Lip(f)$.

(d) Obvious from the definition.

(e) Suppose $\gamma_f(m) = \sigma$ which means $\sigma = \inf_{g\in\mathcal{LB}(m)} SPWL(f, g)$. Now let us calculate the quantity of interest $\gamma_{f_1}(m)$:

$$\begin{aligned}\gamma_{f_1}(m) =& \inf_{g_1\in\mathcal{LB}(m)} SPWL(f_1, g_1)\\ =& \inf_{g_1\in\mathcal{LB}(m)} \sup_{x\in[a/k,b/k]} |f_1(x) - g_1(x)|\\ =& \inf_{g_1\in\mathcal{LB}(m)} \sup_{x\in[a/k,b/k]} |f(kx) - g_1(kx/k)|\\ =& \inf_{g_1\in\mathcal{LB}(m)} \sup_{y\in[a,b]} |f(y) - g_1(y/k)|\\ =& \inf_{g_2(y)=g_1(y/k);\, g_1\in\mathcal{LB}(m)} \sup_{y\in[a,b]} |f(y) - g_2(y)|\\ =& \inf_{g_2\in\mathcal{LB}(m/k)} \sup_{y\in[a,b]} |f(y) - g_2(y)|\\ =&\gamma_f(m/k).\end{aligned}$$

(f) Define $f_1(x) = kf(x)$ on the same domain. Then we have

$$\begin{aligned}\gamma_{f_1}(m) =& \inf_{g_1\in\mathcal{LB}(m)} \sup_{x\in[a,b]} |f_1(x) - g_1(x)|\\ =& \inf_{g_1\in\mathcal{LB}(m)} \sup_{x\in[a,b]} |kf(x) - g_1(x)|\\ =& \inf_{g_2=g_1/k;\, g_1\in\mathcal{LB}(m)} \sup_{x\in[a,b]} |kf(x) - kg_2(x)|\\ =& \inf_{g_2\in\mathcal{LB}(m/k)} \sup_{x\in[a,b]} |k||f(x) - g_2(x)|\\ =&|k|\gamma_f(m/k).\end{aligned}$$

■

Proof Theorem 6.1.

(a) Suppose $\gamma_{f_i}(m_i) = \sigma_i,\ i = 1,2$. Then for any (small) $\epsilon > 0$, there exist functions $g_i \in \mathcal{LB}(m_i)$ such that $SPWL(f_i, g_i) \leq \sigma_i - \epsilon,\ i = 1,2$. Then clearly $g = g_1 + g_2 \in \mathcal{LB}(m)$ and we have

$$\begin{aligned}\gamma_f(m) \leq & SPWL(f,g) \leq SPWL(f_1,g_1) + SPWL(f_2,g_2)\\ \leq & \sigma_1 + \sigma_2 - 2\epsilon = \gamma_{f_1}(m_1) + \gamma_{f_2}(m_2) - 2\epsilon.\end{aligned}$$

Since above holds for any $\epsilon > 0$, we conclude $\gamma_f(m) \leq \gamma_{f_1}(m_1) + \gamma_{f_2}(m_2)$.

(b) Suppose $\gamma_{f_i}^{-1}(\sigma_i) = m_i,\ i = 1,2$ and fix bounded f_1, f_2 so that $f = f_1 + f_2$ and let $d_i = diam(f_1), i = 1,2,\ d = \max\{d_1, d_2\}$. Then for any (small) $\epsilon > 0$, there exist functions $g_i \in \mathcal{LB}(m_i + \epsilon)$ such that $SPWL(f_i, g_i) \leq \sigma_i,\ i = 1,2$. Clearly $g = g_1 + g_2 \in \mathcal{LB}(m + 2\epsilon)$ and define

$$c = \frac{m}{m + 2\epsilon},\ \tilde{g} = cg.$$

Then we have $\tilde{g} \in \mathcal{LB}(m)$ and

$$\begin{aligned}\gamma_f^{-1}(m) \leq SPWL(f,\tilde{g}) \leq SPWL(f_1, cg_1) + SPWL(f_2, cg_2)\\ \leq SPWL(f_1,g_1) + SPWL(g_1,cg_1) + SPWL(f_2,cg_2) + SPWL(g_2,cg_2)\\ \leq \sigma_1 + \sigma_2 + (1-c)diam(g_1) + (1-c)diam(g_2)\\ \leq \sigma_1 + \sigma_2 + (1-c)(d+\sigma_1) + (1-c)(d+\sigma_2)\\ \leq \sigma_1 + \sigma_2 + (1-c)(d+\sigma_1+\sigma_2)\\ = \gamma_{f_1}^{-1}(m_1) + \gamma_{f_2}^{-1}(m_2) + (1-c)(d+\sigma_1+\sigma_2)\end{aligned}$$

Since the above holds for any $\epsilon > 0$, $(1-c) = 2\epsilon/(m+2\epsilon)$ can become arbitrarily small. Now since $(d+\sigma_1+\sigma_2)$ is fixed, we can omit the last term and conclude $\gamma_f^{-1}(m) \leq \gamma_{f_1}^{-1}(m_1) + \gamma_{f_2}^{-1}(m_2)$.

■

Proof Theorem 6.3.
First note that, clearly $\gamma_f(m) - \gamma_g(m) \geq 0$ as f is defined on a domain which includes the domain of g. Now suppose $\gamma_g(m) = \sigma_1$. Then we claim that $\gamma_{LI(g)}(m) = \sigma_1$ also. $\gamma_{LI(g)}(m) \geq \sigma_1$ is obvious because the domain of $LI(g)$ includes that of g. To show that $\gamma_{LI(g)}(m) \leq \sigma_1$, for any $\epsilon > 0$ we will show $\gamma_{LI(g)}(m) \leq \sigma_1 + \epsilon$. Since $\gamma_g(m) = \sigma_1$, there is a grid function h, defined on $\mathbf{x}$, such that $SPWL(g,h) \leq \sigma_1 + \epsilon$ and $h \in \mathcal{LB}(m, \mathbf{x})$. Now consider the linear interpolation of h on the interval $[a,b]$ and denote it by $LI(h)$. Then we have $LI(h) \in \mathcal{LB}(m, [a,b])$. But we also have $SPWL(LI(g), LI(h)) \leq \sigma_1 + \epsilon$ because the supremum distance is obtained at the break points for piece-wise linear functions. Therefore

$$SPWL(f, LI(h)) \leq SPWL(f, LI(g)) + SPWL(LI(g), LI(h)) \leq \sigma + \sigma_1 + \epsilon,\ \forall\epsilon \geq 0.$$

We conclude $\gamma_f(m) \leq \sigma + \sigma_1 = \sigma + \gamma_g(m)$, which completes the proof. ■

Acknowledgements: The authors gratefully acknowledge useful discussions with Drs Duncan Thomas, Jim Gauderman, and Meredith Franklin. This work was partially supported by National Institute of Environmental Health Sciences (5P30ES007048, 5P01ES011627, 5P01ES009581); United States Environmental Protection Agency (R826708-01, RD831861-01); National Heart Lung and Blood Institute (5R01HL061768, 5R01HL076647); California Air Resources Board contract (94-331); and the Hastings Foundation. The first author was also supported by research grants from the Japanese Society for Promotion of Science (JSPS).

References

Azagra, D., Ferrera, J., López-Mesas, F., Rangel, Y. (2007) Smooth approximation of Lipschitz functions on Riemannian manifolds. *Journal of Mathematical Analysis Applications*, 326:1370–1378

Beliakov, G. (2006) Interpolation of Lipschitz functions. *Journal of Computational and Applied Mathematics*, 196(1):20–44

Beliakov, G. (2007) Smoothing Lipschitz functions. *Optimization Methods and Software*, 22:6, 901–916

Cheney, W. and Kincaid, D. (2008) *Numerical Mathematics and Computing, 6th Edition.* Thomson Brooks/Cole

Cleveland, W. S., Grosse, E. and Shyu, W. M. (1992) Local regression models. *Chapter 8 of Statistical Models in S, eds: Chambers, J.M. and Hastie, T.J.*, Wadsworth & Brooks/Cole

Franklin, M., Vora, H., Avol, E., McConnell, R., Lurmann, F., Liu, F., Penfold, B., Berhane, K., Gilliland, F., and Gauderman, W. J. (2012) Predictors of intracommunity variation in air quality. *Journal of Exposure Science and Environmental Epidemiology*, 22:135–147

Gaffney, P. W., and Powell M. J. D. (1976) Optimal interpolation. *in: G.A. Watson (Ed.), On Numerical Analysis, Lecture Notes in Mathematics, vol. 506*, Springer, Heidelberg, 90–99

Gauderman, W. J., Avol, E., Gilliland, F., Vora, H., Thomas, D., Berhane K. et al. (2004) The effect of air pollution on lung development from 10 to 18 years of age. *New England Journal of Medicine*, 351(11):1057–1067

Gauderman, W. J., Vora, H., McConnell, R., Berhane, K., Gilliland, F., Thomas D. et al. (2007) Effect of exposure to traffic on lung development from 10 to 18 years of age: a cohort study. *Lancet*, 369(9561):571–577

Grant, M. and Boyd, S. (2008) Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), editors: Blondel, V., Boyd, S. and Kimura, H., pages 95–110. *Lecture Notes in Control and Information Sciences, Springer*, `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, M. and Boyd, S. (2013) CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`

Hansen, P., Jaumard, B., and Lu, S. H. (1992) Global optimization of univariate Lipschitz functions: I. Survey and properties. *Mathematical Programming*, 55(1–3):251–272

Hansen, P., Jaumard, B., and Lu, S. H. (1992) Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison. *Mathematical Programming*, 55(1–3):273–292

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition*, Springer

Lehmann, T. M., Gönner, C. and Spitzer, K. (1999) Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075

Meijering, E. (2002) A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342

Sergeyev, Y. D. and Kvasov, D. E. (2010) Lipschitz Global Optimization. *In Wiley Encyclopedia of Operations Research and Management Science, Editor-in-Chief: James J. Cochran*, John Wiley & Sons, Inc.

Silverman, B. W. (1985) Some aspects of the spline smoothing approach to non-parametric curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52

Sukharev, A. G. (1978) Optimal method of constructing best uniform approximation for functions of a certain class. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 18(1):21–31

Thévenaz, P., Blu, Th. and Unser, M. (2000) Interpolation Revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758