Final Term Paper

Topics in Econometrics and Statistics, Summer, 2021

Tim Mensinger*
(Matriculation Number: 2916323)

University of Bonn

Abstract

In this essay I introduce the core insights presented in Poß et al., 2020. I extend their work by following up on a remark in the paper, allowing for a less restrictive assumption on the functional regressors. I add functionality to the existing `R`-package and compare results using a Monte-Carlo study. All materials needed to reproduce this project, and in particular the Monte-Carlo study, can be found online[1]. This essay is *not* designed to be read independently from the original paper.

*tmensinger[at]uni-bonn.de

[1]Online repository: `https://github.com/timmens/topics-metrics-2021`

1 Introduction

In this essay I build on the paper *"Superconsistent estimation of points of impact in non-parametric regression with functional predictors"* by Poß et al., 2020. The paper considers the classical problem of scalar outcome prediction in the case of functional regressors. The main variation is the structural assumption that there are certain time points for which the outcome depends *only* on the functional regressors observed at these time points. These points are called *points-of-impact*. The number and location of the points-of-impact is assumed to be unknown and must be estimated from the data directly. A similar problem has been examined in Kneip et al., 2016, where the model presumes a linear influence of the functional regressors at the points-of-impact plus a common effect of the whole trajectory. In Poß et al., 2020 the effect of the whole trajectory is ignored; however, the relationship between the functional regressors at the points-of-impact and the outcome are modeled in a non-parametric framework. Unexpectedly, identification of the points-of-impact does not require harsh assumptions on the link between regressors and outcome, but rather on the structure of the functional regressors themselves. We will see that a useful abstraction level is to require assumptions on the smoothness of the covariance kernel of the functional regressor on and off the diagonal. These assumptions naturally lead to a criterion function which can then be used to estimate the points-of-impact. Following a remark in Poß et al., 2020, I consider a closely related assumption on the covariance kernel which, in principle, allows for a greater number of regressor processes to be modeled. This new assumption then implies a different criterion function. To see how these different criteria compare I run Monte-Carlo studies. The R-code that implements the new features is hosted on GitHub (`https://github.com/timmens/fdapoi`) and builds on the original package corresponding to the paper; see `https://github.com/lidom/fdapoi`.

The rest of this essay is structured as follows: In Section 2 I present the mathematical model and review the main assumptions and theorems from the paper. Section 3 considers the aforementioned extension from a theoretical and computational viewpoint. And at last, in Section 4 I present the Monte-Carlo study.

2 Review

In this section I present the general setting of Poß et al., 2020, albeit restricted to the assumptions and theorems that are needed to understand the extension in Section 3.

We assume there is an independent and identically distributed random sample (X_i, y_i) for $i = 1, \dots, n$ individuals. The functional regressors $X_i = \{X_i(t) : t \in [a, b]\}$ are understood to be a square-integrable process and y_i is a real-valued random variable.

The relationship between regressors and outcome is modeled as

$$y_i = g\left(X_i(\tau_1), \ldots, X_i(\tau_S)\right) + \epsilon_i\,,$$

with ϵ_i representing an error term satisfying $\mathbb{E}\left[\epsilon_i \mid X_i(t)\right] = 0$ for all $t \in [a, b]$. The points-of-impact are denoted by $\tau_1, \ldots, \tau_S$; where the specific locations $\tau_s \in [a, b]$ and the number of points $S \in \mathbb{N}_0$ are both assumed to be unknown a priori. In the same way the link function g is taken to be unknown as well. The paper considers centered random functions X_i.

Given this general framework, one may reasonably question how it is possible to estimate the locations and number of points-of-impact, while allowing for a fully non-parametric link function g. The upcoming repetitions of the main assumptions and theorems in Poß et al., 2020 illustrate that restrictions on X_i through the covariance kernel suffice for the identification and estimation. Let us therefore first consider the covariance kernel of the functional regressor, which I denote by $\sigma(t, s) = \mathbb{E}\left[X_i(t)X_i(s)\right]$.

Assumption 1. *Given the kernel σ, there exists open $\Omega \subset [0,1]^3$ and twice continuously differentiable function $\omega : \Omega \to \mathbb{R}$, as well as some $\kappa \in (0, 2)$, such that $\forall s, t \in [0, 1]$*

$$\sigma(s, t) = \omega(s, t, |s - t|^\kappa)\,.$$

Moreover, $0 < \inf\{c(t) : t \in [0, 1]\}$, where $c(t) = -\frac{\partial}{\partial z}\omega(t, t, z)|_{z=0}$.

Assumption 1 restricts the degree of smoothness at the diagonal using the parameter κ. This in turn implies certain behavior of the sample paths of the process. Values with $\kappa < 2$ suggest non-smooth trajectories, while processes with smooth sample paths and twice continuously differentiable kernel will satisfy the assumption with $\kappa = 2$. That is, in the current form Assumption 1 implies that the sample paths of the regressors X_i need to be somewhat rough. Many known processes fulfill Assumption 1, for example, Brownian Motion fulfills it with $\kappa = 1$. In Section 3 I present an extension to this assumption with $\kappa > 2$, allowing for a greater number of processes to be modeled.

The ability to identify and estimate the points-of-impact relies heavily on the decomposition presented in Theorem 1.

Theorem 1. *Let X_i be a Gaussian process and $g : \mathbb{R}^S \to \mathbb{R}$ an arbitrary function with continuous partial derivatives almost everywhere. For $s = 1, \ldots, S$ define*

$$\vartheta_s = \mathbb{E}\left[\frac{\partial}{\partial x_s} g(X_i(\tau_1), \ldots, X_i(\tau_S))\right]\,.$$

If $0 < |\vartheta_s| < \infty, \forall s = 1, \ldots, S$, then we may write, $\forall t \in [a, b]$

$$f_{XY}(t) \overset{def}{=} \mathbb{E}\left[X_i(t)y_i\right] = \sum_{s=1}^{S} \vartheta_s \sigma(t, \tau_s)\,.$$

In view of Assumption 1 we know that the kernel σ is *not* two-times differentiable at the diagonal. In that case f_{XY} will not be two-times differentiable at the points-of-impact $\tau_1, \ldots, \tau_S$. As it turns out, this will be enough to ensure identification. To differentiate between any point ($t \in [a, b]$) and a point-of-impact ($t = \tau_s$ for some s) the paper proposes the measure

$$f_{ZY}(t) \stackrel{def}{=} f_{XY}(t) - \frac{1}{2}\left(f_{XY}(t+\delta) + f_{XY}(t-\delta)\right),$$

with hyper-parameter $\delta > 0$. Under our current set of assumptions, the function f_{ZY} will be large in absolute value for t close to a point-of-impact. This then allows for the estimation of the points-of-impact using extremum points of an estimated version of $|f_{ZY}|$. But what is f_{ZY} actually measuring? Notice that there is an intriguing relationship between our criterion f_{ZY} and the second-order central finite difference of f_{XY}:

$$\begin{aligned} FD(f_{XY}, \delta, 2, x) &= (f(x+\delta) + f(x-\delta)) - 2f(x) \\ &= -\frac{1}{2} f_{ZY}(x), \end{aligned}$$

where $FD(h, \delta, k, x)$ denotes the k-th order central finite difference of function h with step length δ, evaluated at x. Henceforth I will just write *finite difference* and drop the *central*. Since the criterion function is in absolute value, using f_{ZY} is proportional to using a second-order finite difference of f_{XY}. This should make sense, as Assumption 1 implies that f_{XY} should not be two-times differentiable at the points-of-impact. If that is the case then the second-order finite difference is expected to be larger at the points-of-impact than at the other time points.

With this new interpretation in mind we may ask whether we can exploit higher-order finite differences to relax the assumption on the covariance kernel? The answer is affirmative and the details will be the concern of Section 3.

At last I state the original version of the estimation algorithm as a comparison to the more general version supplied in the next section. For the estimation stage we suppose that the functional regressors X_i are observed at p equidistant points $t_1, \ldots, t_p$ with $a \leq t_1$ and $t_p \leq b$. Since the functions f_{XY} and f_{ZY} are simple functions of expectations, they can be easily estimated by the standard sample counterpart, which I denote by $\hat{f}_{XY}$ and $\hat{f}_{ZY}$. The original algorithm is listed as Algorithm 1.

The main identification and consistency results are found in Theorem 2 in Poß et al., 2020.

1: compute $\hat{f}_{XY}(t_j) = \sum_i X_i(t_j)Y_i/n$, for all $j = 1, \ldots, p$
2: choose $\delta > 0$ s.t. $\exists\, k_\delta \in \mathbb{N}$ with $1 \leq k_\delta < (p-1)/2$ and $\delta = k_\delta/(p-1)$
3: define $\mathcal{J}_\delta = \{k_\delta + 1, \ldots, p - k_\delta\}$ and set $\ell = 1$
4: compute $\hat{f}_{ZY}(t_j) = \hat{f}_{XY}(t_j) - (\hat{f}_{XY}(t_j + \delta) + \hat{f}_{XY}(t_j - \delta))/2$, for all $j \in \mathcal{J}_\delta$
5: **while** $\mathcal{J}_\delta \neq \varnothing$ **do**
6: estimate $\hat{\tau}_\ell = \text{argmax}\left\{ |\, \hat{f}_{ZY}(t_j)| : \text{for } t_j \text{ with } j \in \mathcal{J}_\delta \right\}$
7: update $\mathcal{J}_\delta \leftarrow \mathcal{J}_\delta \setminus [\hat{\tau}_\ell - \sqrt{\delta}, \hat{\tau}_\ell + \sqrt{\delta}]$
8: update $\ell \leftarrow \ell + 1$
9: **return** $\{\hat{\tau}_\ell\}$

Algorithm 1: *Original algorithm from Poß et al., 2020, adapted for readability.*

3 Extension

In this section I consider a generalization which was touched upon in the previous section. This corresponds to the generalization discussed in subsection 2.4.2 (*Generalizing covariance assumption 1*) of Poß et al., 2020.

One of the main restrictions of Assumption 1 is that it excludes smooth, twice continuously differentiable processes with $\kappa \geq 2$. On first thought, it may be intuitive that a certain degree of *local variation* is necessary for identification; as also discussed in Kneip et al., 2016. However, in light of Theorem 1 this is not necessary. As discussed in the previous section, a sensible criterion function inherits the properties of the covariance kernel at the diagonal to allow for the differentiation between regular time points and points-of-impact. Before, under Assumption 1 we presumed that the kernel is twice continuously differentiable off the diagonal and not twice continuously differentiable on the diagonal. But this can be relaxed to the rather general case where the kernel is less smooth on the diagonal than off the diagonal. For example, we may extend the assumption to the case $\kappa < 4$, which then implies that the kernel may be four-times continuously differentiable off the diagonal but not on the diagonal. Given this modified assumption we would then have to adapt the definition of f_{ZY} and hence the criterion function.

This generalization may be used for any even $d = 2, 4, 6, 8, \ldots$ while using Assumption 1 with $\kappa < d$. For each d the natural extension of the criterion function then involves the d-th order finite difference. Again, let $FD(h, \delta, k, x)$ denote the k-th order finite difference of function h with step size δ at x. Algorithm 1 can then be gener-

alized by updating lines 3 and 4. Since higher-order finite difference computations need multiple steps in the argument, the grid spanned in line 3 has to be adjusted. Most importantly the second-order finite difference formula from line 4 is exchanged for the general case. The updated version is listed as Algorithm 2. This version of the algorithm is used in the Monte-Carlo study (Section 4) and is available online: `https://github.com/timmens/fdapoi`.

1: compute $\hat{f}_{XY}(t_j) = \sum_i X_i(t_j)Y_i/n$, for all $j = 1, \ldots, p$
2: choose $\delta > 0$ s.t. $\exists\, k_\delta \in \mathbb{N}$ with $1 \leq k_\delta < (p-1)/2$ and $\delta = k_\delta/(p-1)$
3: define $\mathcal{J}_\delta = \textsc{ComputeGrid}(p, k_\delta, d)$ and set $\ell = 1$
4: compute $\hat{f}^*(t_j) = FD(\hat{f}_{XY}, \delta, d, t_j)$ for all $j \in \mathcal{J}_\delta$
5: **while** $\mathcal{J}_\delta \neq \varnothing$ **do**
6: estimate $\hat{\tau}_\ell = \text{argmax}\left\{ |\hat{f}^*(t_j)| : \text{for } t_j \text{ with } j \in \mathcal{J}_\delta \right\}$
7: update $\mathcal{J}_\delta \leftarrow \mathcal{J}_\delta \setminus [\hat{\tau}_\ell - \sqrt{\delta}, \hat{\tau}_\ell + \sqrt{\delta}]$
8: update $\ell \leftarrow \ell + 1$
9: **return** $\{\hat{\tau}_\ell\}$

Algorithm 2: *Generalization of Algorithm 1.*

Using arbitrary large values for d does not come without costs, though. There are two main potential problems. One, higher-order finite differences can be numerically unstable, especially in our case where δ is usually chosen much larger than in the standard case of derivative approximation. And two, for higher-order differences fewer points in $\{t_1, \ldots, t_p\}$ can be used, because an evaluation of the finite difference formula requires more and more points to the left and right of the evaluation point. This latter point implies that the algorithm becomes blind to points-of-impact close to the boundaries a and b.

My implementation of general higher-order central differences is taken from Jordan et al., 1965, which further provides a helpful introduction to the topic. The formula is given by

$$FD(f, \delta, k, x) = \sum_{j=0}^{k} (-1)^j \binom{k}{j} f\left(x + \left[\frac{k}{2} - j\right]\delta\right) .$$

4 Monte-Carlo Study

In this last section I test the aforementioned extension using a simulation study.

The Monte-Carlo design compares an application of Algorithm 2 for $d \in \{2, 4\}$. To gain a better understanding on the criticality of Assumption 1, and its relaxation, I consider a parameterized functional regressor which allows me to choose the level of local variation.

Setup. The data generating process is setup as follows. For a given smoothness parameter $\nu \in \{0.5, 1.5, 2.5\}$, the functional regressors X_i are simulated as a mean-zero Gaussian process with a Matern covariance kernel using length scale parameter $\ell = 0.1$ and smoothness parameter ν —a mathematical description of the Matern kernel and its relation to Assumption 1 is provided in the next paragraph. The process is observed for $T = 100$ periods on an equidistant grid. I compare how the method performs for $S \in \{0, 1, 2\}$ points-of-impact. In the case of $S = 1$ the location is given by $\tau_1 = 49$. And in the case of $S = 2$ we have $(\tau_1, \tau_2) = (24, 49)$. Given the number of points-of-impact S, the coefficient vectors are fixed with: $\beta_0 = (1)$, $\beta_1 = (1, 2)$ and $\beta_2 = (1, 2, -1)$. The outcomes are then simulated using

$$y_i = \beta_{S,0} + \sum_{r=1}^{S} \beta_{S,r} \, X_i(\tau_r) + \epsilon_i \,,$$

where ϵ_i is an i.i.d. Gaussian error with $Var(\epsilon_i) = 1/2$. Note that $\beta_{S,r}$ denotes the r-th entry of the $(S+1)$-dimensional coefficient vector, in the case of S number of points-of-impact. The number of observations is fixed to $n = 100$. In principle, it would be interesting to see how the results depend on the sample size; however, for the sake of clarity I refrain from analyzing this dimension.

Figure 4.1 illustrates three simulated sample paths of the functional regressor, for the three different smoothness parameters ν. As is clearly visible, for small ν (top row) the process possesses a lot of local variation, while for larger ν (bottom row) the process is much smoother with a low level of variation.

The Kernel. For this study I choose kernels from the Matern class. Explicitly, the kernel is defined by

$$\sigma(s,t) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|s-t|}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}|s-t|}{\ell} \right) ,$$

with length scale parameter $\ell > 0$ and smoothness parameter $\nu > 0$. Note that Γ denotes the usual gamma function, while K_ν is a modified Bessel function. For a more

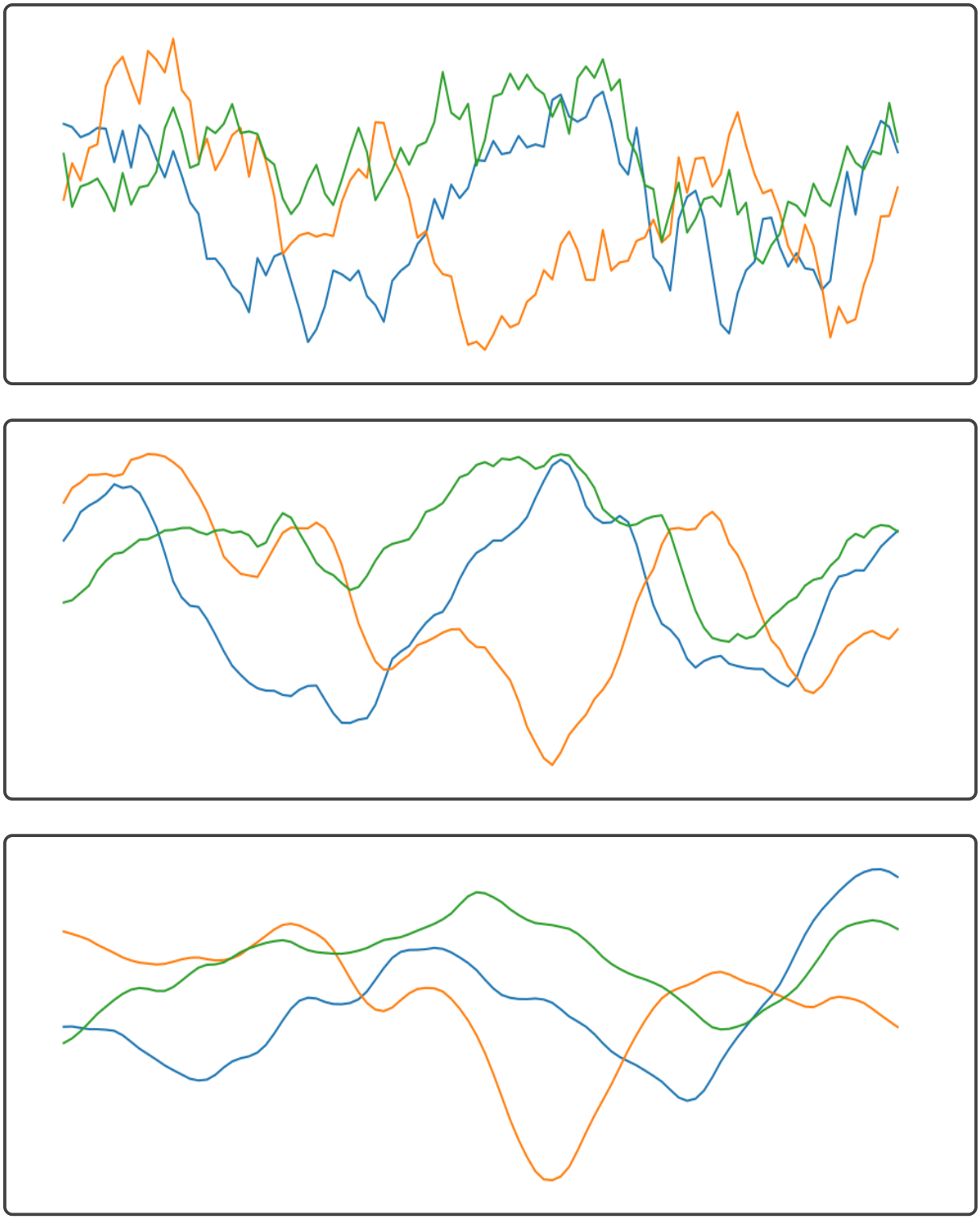

Figure 4.1: *Simulated trajectories of a Gaussian process with Matern kernel and differing smoothness parameter; top:* $\nu = 0.5$*; center:* $\nu = 1.5$*; bottom:* $\nu = 2.5$*.*

ν	$\sigma_{\text{Matern}}(z)$
1/2	$\exp\left(-z/\ell\right)$
3/2	$\left(1+\sqrt{3}z/\ell\right)\exp\left(-\sqrt{3}z/\ell\right)$
5/2	$\left(1+\sqrt{5}z/\ell+5z^2/(3\ell^2)\right)\exp\left(-\sqrt{5}z/\ell\right)$

Table 1: *Simple expressions of the Matern kernel for special cases of the smoothness parameter ν; see Rasmussen and Williams, 2006. Here $z = |s-t|$.*

detailed reference on the components of the Matern kernel, as well as a reference for the following properties, see Rasmussen and Williams, 2006. The above expression is hard to work with. Luckily, for the special cases $\nu \in \{0.5, 1.5, 2.5\}$ it simplifies dramatically, as is shown in Table 1, where I define $z = |s-t|$.

Remark. I must note that, for the case of $\kappa < 2$, neither the Matern kernel with $\nu = 1.5$ nor with $\nu = 2.5$ satisfy Assumption 1. This is because $\inf\{c(t) : t \in [0,1]\} = 0$ in these cases. That means that the covariance at the diagonal does not drop off fast enough. Furthermore, an extension of the assumption to the case $\kappa < 4$ is not of help either, as the partial derivative $\partial\omega/\partial z$ is not defined at $z = 0$ for any $\kappa > 1$. To analyze the extended method properly one would need to use a kernel satisfying an extended version of Assumption 1 with $\kappa \in [2, 4)$. It proved difficult to find a reasonable kernel for this case. As the sample paths induced by the Matern kernel with high ν are fairly standard, the results should, nevertheless, still tell us something of relevance about the underlying method.

Monte-Carlo Design. I perform 500 Monte-Carlo repetitions over the parameter grid spanned by $d \in \{2, 4\}, S \in \{0, 1, 2\}$ and $\nu \in \{0.5, 1.5, 2.5\}$. The results are visualized using frequency plots that summarize the estimated points-of-impact over *all* Monte-Carlo repetitions. A detailed explanation follows in the next paragraph. Alternatively, one could have computed e.g. the Hausdorff-distance between the true and estimated points-of-impact in each simulation run, or reported the average number of estimated points-of-impact. For the sake of brevity I stick to one way of reporting the results. Furthermore, in this study I focus only on the estimation of the points-of-impact and not on the subsequent estimation of the coefficient parameters.

Results. Before we consider the actual results, let us think about what we may expect. The slope parameter corresponding to the second point-of-impact is significantly smaller than the one corresponding to the first point-of-impact. Hence, in the case of $S = 2$ we should expect that the method finds the first point-of-impact at least as often. We also

expect that the precision of the method decreases in ν, i.e. the smoother the functional regressor the less precise the estimates. For the case of $S = 0$ we should see no difference for varying smoothness nor for varying d. What we hope to see is that for the case $d = 4$, i.e. when using the fourth-order finite difference, the performance of the method increases in the smoother $\nu = 2.5$ case.

Figure 4.2 summarizes the results when applying the standard algorithm ($d = 2$). The top row shows the case of no points-of-impact ($S = 0$), the center row depicts the case of one point-of-impact ($S = 1$) and the bottom row exhibits the case of two points-of-impact ($S = 2$). The results are consistent with our expectations. The plot can be understood as follows: In the middle sub-figure we see that for the $\nu = 0.5$ case the true location (49) makes up more than 50% of the estimated locations. In all cases for ν and both cases for d we see no difference in the frequency of false-positives (top row).

Even though Assumption 1 is not satisfied for $\nu \in \{1.5, 2.5\}$, the estimated locations form a cluster around the true points. This should tell us that a mild violation of Assumption 1 does not ruin the analysis completely and that an improved criterion function may result in even more precise estimates. The large variance is also in line with the argumentation of Poß et al., 2020, in that the smooth functional regressors make it harder for the method to distinguish between the influence of neighboring points.

Figure 4.3 depicts the case when employing the fourth-order finite difference ($d = 4$). The image is similar to the above figure. However, a main difference is that the precision is lower than in the $d = 2$ case. Especially in the smooth cases the method performs worse, as there are no sharp peaks and the estimated points are spread over a large area.

The observation from Figure 4.3 may be due to several reasons. First, as stated in the aforementioned remark, the kernel used to simulate the functional regressors does not satisfy the extended Assumption 1. The method remains to be tested with such a kernel. A counter-argument would be that in the coarsely discretized case the regressors will almost certainly look similar to the ones I am using here. Second, the fourth-order finite difference formula may not be precise enough. One could think about modifying this measure slightly to get a more precise criterion function.

References

Jordan, C., Jordán, K., & Carver, H. (1965). *Calculus of finite differences*. Chelsea Publishing Company.

Kneip, A., Poß, D., & Sarda, P. (2016). Functional linear regression with points of impact. *The Annals of Statistics*.

Poß, D., Liebl, D., Kneip, A., Eisenbarth, H., Wager, T. D., & Barrett, L. F. (2020). Superconsistent estimation of points of impact in non-parametric regression with functional predictors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. MIT Press.

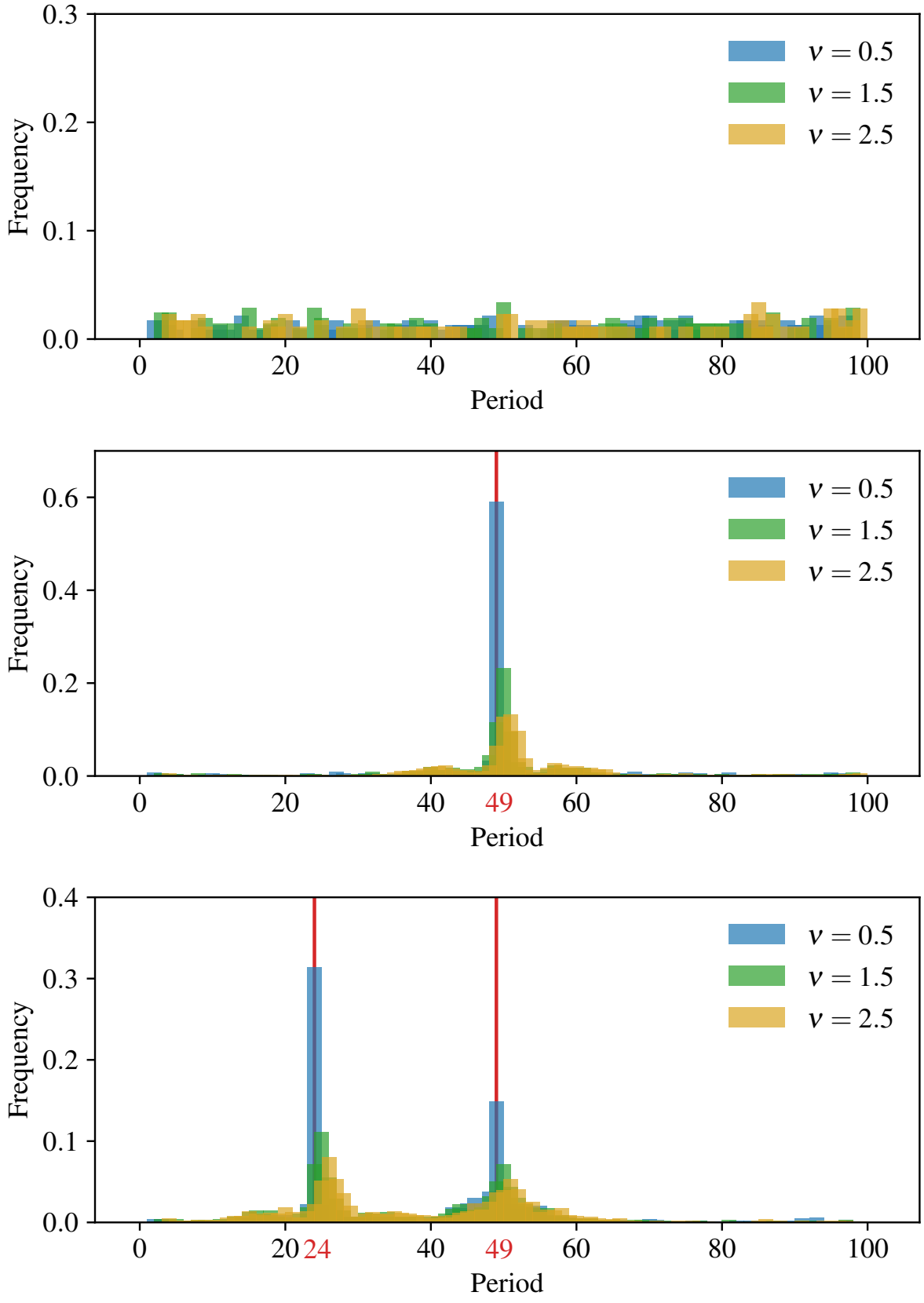

Figure 4.2: *Results from 500 Monte-Carlo repetitions. Length scales are differentiated using color; orange: $\nu = 0.5$; green: $\nu = 1.5$; blue: $\nu = 2.5$. In this Monte-Carlo run Algorithm 2 was used with order $d = 2$. True points-of-impact are depicted by the red vertical line.*

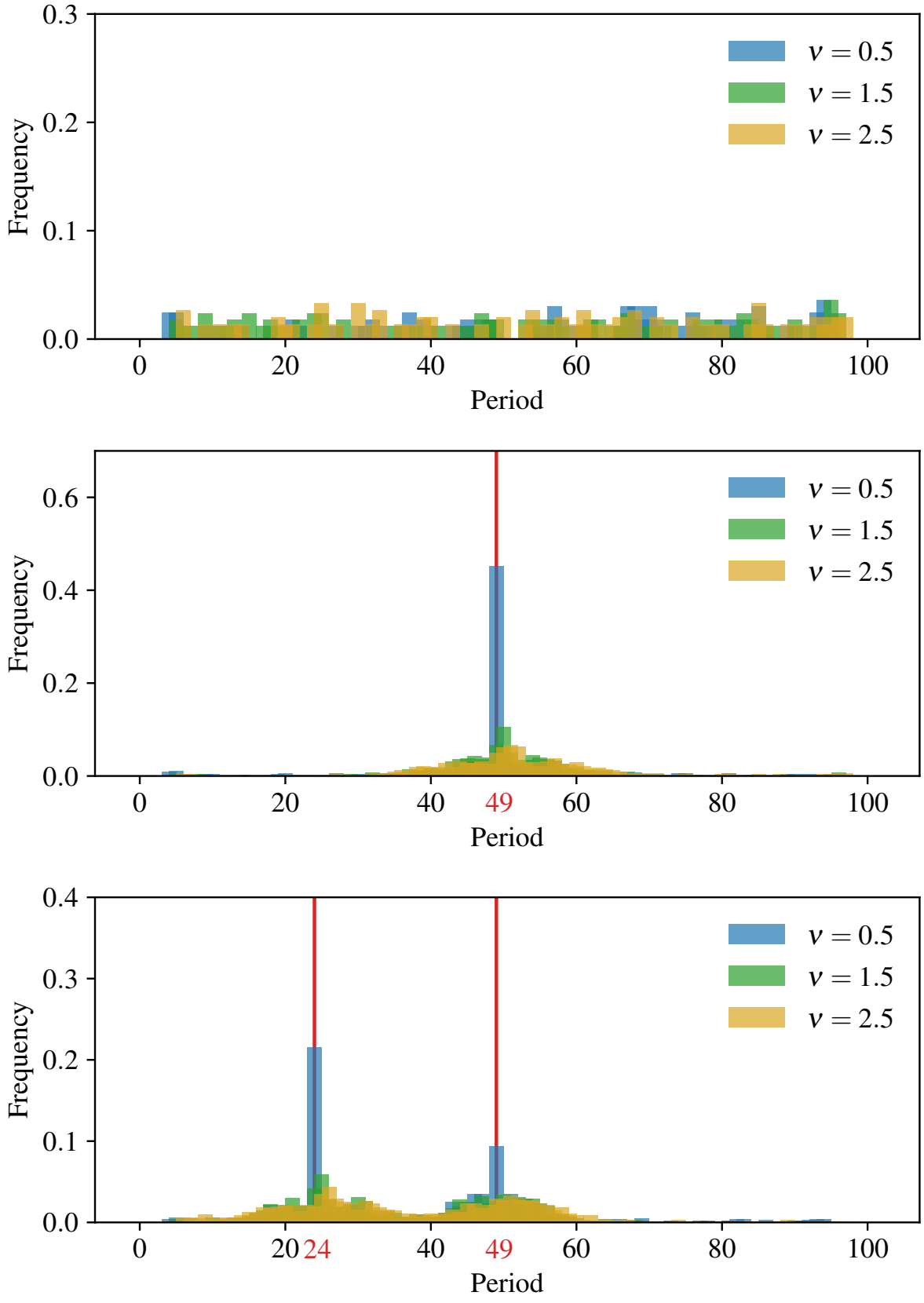

Figure 4.3: *Results from 500 Monte-Carlo repetitions. Length scales are differentiated using color; orange: $\nu = 0.5$; green: $\nu = 1.5$; blue: $\nu = 2.5$. In this Monte-Carlo run Algorithm 2 was used with order $d = 4$. True points-of-impact are depicted by the red vertical line.*