

Article

Second-Moment/Order Approximations by Kernel Smoothers with Application to Volatility Estimation

León Beleña ^{1,2,*} , Ernesto Curbelo ³ , Luca Martino ¹  and Valero Laparra ⁴ 

¹ Department of Signal Processing, Universidad Rey Juan Carlos, 28942 Fuenlabrada, Spain; luca.martino@urjc.es

² Universidad Francisco de Vitoria, Ctra. Pozuelo-Majadahonda Km 1,800, 28223 Pozuelo de Alarcón, Spain

³ Department of Statistics, Universidad Carlos III de Madrid, 28911 Leganés, Spain; ecurbelo@est-econ.uc3m.es

⁴ Image Processing Lab, Universitat de Valencia, 46980 Paterna, Spain

* Correspondence: leon.belena@ufv.es

Abstract: Volatility estimation and quantile regression are relevant active research areas in statistics, machine learning and econometrics. In this work, we propose two procedures to estimate the local variances in generic regression problems by using kernel smoothers. The proposed schemes can be applied in multidimensional scenarios (not just for time series analysis) and easily in a multi-output framework as well. Moreover, they enable the possibility of providing uncertainty estimation using a generic kernel smoother technique. Several numerical experiments show the benefits of the proposed methods, even compared with the benchmark techniques. One of these experiments involves a real dataset analysis.

Keywords: quantile regression; kernel smoothers; times series; heteroscedasticity; nearest neighbors

MSC: 62G08; 62G15; 62G0; 62F1; 62F25



Citation: Beleña, L.; Curbelo, E.; Martino, L.; Laparra, V. Second-Moment/Order Approximations by Kernel Smoothers with Application to Volatility Estimation. *Mathematics* **2024**, *12*, 1406. <https://doi.org/10.3390/math12091406>

Academic Editor: Ionut Florescu

Received: 16 February 2024

Revised: 30 April 2024

Accepted: 2 May 2024

Published: 4 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Regression analysis can be considered as a set of methodologies for estimating the relationships between a dependent variable y (often called an ‘output’ or ‘response’) and a vector \mathbf{x} of independent variables (often called an ‘input’). Essentially, the main goal in a regression problem is to obtain an approximation of the trend defined as the expected value of y given \mathbf{x} , i.e., $E[y|\mathbf{x}]$; that is, the first (non-central) moment of the conditional density $p(y|\mathbf{x})$. We can assert that the most complete regression problem consists of approximating the whole conditional density $p(y|\mathbf{x})$, whereas the simplest task in a regression problem consists of estimating only the first moment $E[y|\mathbf{x}] = \int y p(y|\mathbf{x}) dy$. Intermediate scenarios appear in different applications, where other moments (higher than one) are of interest and hence are also approximated.

Volatility estimation (intended as local variance or local standard deviation) and quantile regression analysis are currently important tasks in statistics, machine learning and econometrics. The problem of volatility estimation has particular relevance in financial time series analysis, where the volatility represents the degree of variation in a trading price series over time, usually measured by the standard deviation of logarithmic returns. Indeed, it is important to remark that an inherent problem in the volatility estimation is that the conditional variance is latent and hence is not directly observable when real data are analyzed. It is still an important active area of research [1–4].

In the literature, several schemes have been proposed. A well-known family among the proposed approaches is formed by the generalized autoregressive conditional heteroskedasticity (GARCH) models [5,6]. They are considered the benchmark for time series analysis. Other important classes of techniques are the stochastic volatility models [7] ([8] Section 7.4) and the exponentially weighted moving average models [9]. They have

also been extended for multivariate scenarios, but the application is much more complex than the method that we propose here. Moreover, other criticisms have been detected in the literature: as stated by some authors [6,10,11], most of the latent volatility models fail to satisfactorily describe several stylized facts that are observed in financial time series. Several schemes are based on state-space models ([8] Section 7.4). Some important review papers on this topic can be found in the following [12–15]. In these relevant works, the authors have also focused on the theoretical foundations of the estimators that have been proposed recently.

Other approaches have been considered. For instance, the local linear regression methods (which can resemble a kernel smoother approach) have already been proposed for volatility estimation [16–18]. They share the notion of “locality” with the kernel smoother approach and perform several linear regressions (applied to the residual errors for estimating the variance). Generally, for all these schemes, the extensions for the multivariate and/or multi-output cases are not straightforward. Moreover, most of them are parametric methods (here, we consider non-parametric techniques where the complexity of the model is related to the number of observed data). In mathematical finance and financial engineering, another class of methods is formed by the so-called local volatility models, which are generalizations of the Black–Scholes model. Local volatility models are similar and related to stochastic volatility models, where the instantaneous volatility is itself a random variable [19]. Other authors have studied the high-frequency features of the data, which contain market microstructure noise, in order to estimate the instantaneous volatility (e.g., [20]). Quantile regression models study the relationship between an independent input vector \mathbf{x} and some specific quantile or moment of the output variable y [21–23]. Therefore, in quantile regression analysis, the goal is to estimate the higher-order moments of the response/output variable given an input \mathbf{x} [21–26].

On the other hand, it is important to remark that many advanced and benchmark regression methods in the literature (such as Gaussian processes, etc.) consider a *constant variance* as the initial assumption of the output given the input, $\text{var}[y|\mathbf{x}] = \sigma_e^2$; i.e., it does vary with the input variable \mathbf{x} [27,28].

In this work, we consider an extended approach (with respect to the assumptions used in classical regression methods) where $\text{var}[y|\mathbf{x}] = v(\mathbf{x})$; i.e., the local variance varies with the input \mathbf{x} . More precisely, in this work, we propose two procedures for estimating the local variance in a regression problem and using a kernel smoother approach ([28] Section 9) [29]. Note that the kernel smoother schemes contain several well-known techniques as special cases, such as the *fixed radius nearest neighbors approach*, as an example [30]. The resulting solution is the *non-parametric* method; hence, both the complexity and flexibility of the solution increase with the number of data N . This ensures having the adequate flexibility to analyze the data.

The first proposed method is based on the Nadaraya–Watson derivation of a linear kernel smoother [27,28]. The second proposed approach draws inspiration from *divisive normalization*, a function grounded in the activity of brain neurons [31]. This function aims to standardize the neuron activity by dividing it by the activity of the neighboring neurons. It has demonstrated favorable statistical properties [32–34] and has been utilized in various applications [35,36]. Other important and related schemes, containing relevant theoretical results, can be found in the literature [37–39].

The proposed methods can be applied to time series analysis and/or in more general regression problems where the input variables are multidimensional, and with correlated noise. Therefore, our approach can be implemented in spatial statistical modeling and any other inverse problems [23,40]. There are many datasets with multidimensional inputs (e.g., including space and time variables) whose structures involve a substantial change in variance. The decomposition of the seismic data is just an example. Furthermore, another advantage of the proposed scenario is that the generalization for multi-output scenarios can be easily designed. More generally, the proposed approach is easy to apply and also easy to extend and generalize in terms of flexibility: different regression techniques or different

kernel functions can be considered (possibly different for trend and variance estimation), and the number of hyper-parameters can be decided by the user.

From another point of view, this work provides another relevant contribution. The proposed schemes enable performing uncertainty estimation with a generic kernel smoother technique. Indeed, a generic kernel smoother method is not generally supported by a generative probabilistic derivation that also yields an uncertainty estimation. The Gaussian processes (GPs) and relevance vector machines (RVMs) regression methods are relevant well-known and virtually unique exceptions [28,41].

We have tested the proposed methods in five numerical experiments. Four of them involve the application of the different schemes to time series analysis and comparison with GARCH models, which are considered benchmark techniques for volatility estimation in time series [42]. The last numerical experiment addresses a more general regression problem, with a multidimensional input variable \mathbf{x} of dimension 122, considering a real (emo-soundscapes) database [43,44].

The paper is structured as follows. The main methodology employed here for regression, based on kernel smoothers, is described in Section 2. Several features and relevant characteristics are also discussed. The two proposed variance estimation procedures are introduced in Section 3. The different extensions, generalizations, and variants are provided in Section 4. Numerous numerical experiments covering different scenarios (and with several comparisons) are provided in Section 5. Finally, Section 6 is devoted to the conclusions and some possible ideas for future work.

2. Approximating the Trend

Let us consider a set of N data pairs, such as $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$, with $D \geq 1$ and $y_i \in \mathbb{R}$. First of all, we are interested in obtaining a regression function (a.k.a., “local mean” trend), i.e., removing the noise in the signal obtaining an estimator $\hat{f}(\mathbf{x})$ for all possible \mathbf{x} in the input space. One possibility is to employ a *linear kernel smoother*. More specifically, we consider the Nadaraya–Watson estimator [27,28] that has the following form,

$$E[y|\mathbf{x}] \approx \hat{f}(\mathbf{x}) = \sum_{n=1}^N \frac{h(\mathbf{x}, \mathbf{x}_n)}{\sum_{j=1}^N h(\mathbf{x}, \mathbf{x}_j)} y_n = \sum_{n=1}^N \varphi(\mathbf{x}, \mathbf{x}_n) y_n, \quad (1)$$

where $\varphi(\mathbf{x}, \mathbf{x}_n) = \frac{h(\mathbf{x}, \mathbf{x}_n)}{\sum_{j=1}^N h(\mathbf{x}, \mathbf{x}_j)}$ and $h(\mathbf{x}, \mathbf{z})$ is a function (which is often called ‘kernel’) decided by the user and defined in $\mathbb{R}^D \times \mathbb{R}^D$. Note that, by this definition, the nonlinear weights $\varphi(\mathbf{x}, \mathbf{x}_n)$ are normalized, i.e.,

$$\sum_{n=1}^N \varphi(\mathbf{x}, \mathbf{x}_n) = 1, \quad \forall \mathbf{x}. \quad (2)$$

As an example, we could consider

$$h(\mathbf{x}, \mathbf{z}) = h(\mathbf{x}, \mathbf{z}|\lambda) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{\lambda}\right),$$

where λ is a hyper-parameter that should be tuned. Clearly, we also have

$$\varphi(\mathbf{x}, \mathbf{z}) = \varphi(\mathbf{x}, \mathbf{z}|\lambda).$$

The form of this estimator above is quite general and contains other different well-known methods as special cases. For instance, it contains the k-nearest neighbors algorithm (kNN) for regression as a specific case (to be more specific, the fixed-radius near neighbors algorithm). Indeed, with a specific choice of $h(\mathbf{x}, \mathbf{x}_n|\lambda)$ (as a rectangular function), the expression above can represent the fixed-radius near neighbors estimator [28,30].

Remark 1. The resulting regression function is a flexible non-parametric method. Both complexity and flexibility of the solution grow with the number of data N since, at each new data point \mathbf{x}_{N+1} , it corresponds to the use of an additional kernel function $\varphi(\mathbf{x}, \mathbf{x}_{N+1})$ located at \mathbf{x}_{N+1} , and the final solution \hat{f} will be a linear combination of $N + 1$ values.

Remark 2. The characteristics of the solution are determined by the choice of the kernel, which is decided by the user. For instance, the choice of a Gaussian kernel function, $h(\mathbf{x}, \mathbf{x}_n)$, generates a smooth infinitely differentiable solution. The choice of a Laplacian kernel function yields a solution that is not differentiable at all the inputs' \mathbf{x}_n 's in the dataset.

Remark 3. The input variables $\mathbf{x}_n \in \mathbb{R}^D$ are vectors ($D \geq 1$) in general. Therefore, the described methods have much wider applicability than the techniques that can be employed only for time series (where the time index is a scalar number, $x = t \in \mathbb{R}$). Clearly, the methodologies described here can also be employed for analyzing time series. Moreover, even in a time series framework, we can obtain a prediction between two consecutive time instants. For instance, if we have a time series with daily data, with a kernel smoother, we can obtain a prediction at each hour (or minute) within a specific day.

Remark 4. It is important to remark that the application to multidimensional input case, $\mathbf{x}_n \in \mathbb{R}^D$, is in general fast and straightforward, unlike for other local piecewise polynomial solutions (e.g., [17,18]), which would require the construction of multidimensional and expensive grid (to build and evaluate the piecewise regressor properly). Moreover, the proposed framework contains these piecewise solutions as special cases, choosing a kernel function that is non-zero just in a subset of the support (e.g., a rectangular kernel).

Learning λ . One possibility for tuning the hyper-parameters of the kernel functions is to use a cross-validation (CV) procedure. In this work, we have employed a *leave-one-out* cross-validation (LOO-CV) [27].

3. Variance Estimation Procedures

Let us assume that we have already computed the trend (a.k.a., “local mean”), i.e., the regression function $\hat{f}(\mathbf{x})$. Here, we present two methods to obtain an estimation of the local variance (or volatility) at each point \mathbf{x} , which is theoretically defined as

$$\begin{aligned} v(\mathbf{x}) = \text{var}[y|\mathbf{x}] &= \int_{\mathbf{y}} (y - E[y|\mathbf{x}])^2 p(y|\mathbf{x}) dy, \\ &= E[y^2|\mathbf{x}] - (E[y|\mathbf{x}])^2. \end{aligned} \quad (3)$$

METHOD-1 (M1). If the weights $\varphi(\mathbf{x}, \mathbf{x}_n)$ are adequate for linearly combining the outputs y_i and obtaining an proper approximation of $E[y|\mathbf{x}]$, one can extend this idea for approximating the second non-central moment $E[y^2|\mathbf{x}]$ as

$$E[y^2|\mathbf{x}] \approx \hat{q}(\mathbf{x}) = \sum_{n=1}^N \varphi(\mathbf{x}, \mathbf{x}_n) y_n^2. \quad (4)$$

hence

$$\hat{v}(\mathbf{x}) = \hat{q}(\mathbf{x}) - (\hat{f}(\mathbf{x}))^2 \approx \text{var}[y|\mathbf{x}], \quad (5)$$

which is an estimator of the *instant variance*. Note that $\hat{q}(\mathbf{x})$ and $\hat{f}(\mathbf{x})$ are obtained with the same weights

$$\varphi(\mathbf{x}, \mathbf{x}_n) = \varphi(\mathbf{x}, \mathbf{x}_n|\lambda),$$

with the same value of λ (obtained using a LOO-CV procedure). Thus, a variant of this procedure consists of learning another value of λ_2 , i.e., obtaining other coefficients $\varphi_2(\mathbf{x}, \mathbf{z}) = \varphi_2(\mathbf{x}, \mathbf{z}|\lambda_2)$, in the Equation (4) considering the signal y_n^2 (instead of y_n).

METHOD-2 (M2). Let us define the signal obtained as the estimated square errors

$$v_n = (y_n - \hat{f}(\mathbf{x}_n))^2, \quad n = 1, \dots, N.$$

If $\hat{f}(\mathbf{x}) \approx E[y|\mathbf{x}]$ as assumed, v_n is a one-sample estimate of the variance at \mathbf{x}_n . Then, the goal is to approximate the trend of this new signal (i.e., new output) v_n ,

$$\hat{v}(\mathbf{x}) = \sum_{n=1}^N \varphi_2(\mathbf{x}, \mathbf{x}_n) v_n, \quad (6)$$

where we consider another parameter λ_2 , i.e.,

$$\varphi_2(\mathbf{x}, \mathbf{z}) = \varphi_2(\mathbf{x}, \mathbf{z}|\lambda_2),$$

that is tuned again with LOO-CV but considering the new signal v_n . Note that $\hat{v}(\mathbf{x})$ can be interpreted as estimation of the *instant variance* of the underlying signal. As an alternative, completely different kernel functions (as φ_2) can also be applied (that differ in the functional form with respect to φ instead of just for the choice regarding λ).

Remark 5. Again, as for estimating the trend, note that the two procedures above can be applied for multivariate inputs $\mathbf{x}_n \in \mathbb{R}^D$ and not just for scalar inputs (as in the time series).

Remark 6. If one divides \mathbf{x} by $\hat{v}(\mathbf{x})$, we obtain a signal with uniform local variance; i.e., we have removed the (possible) heteroscedasticity. In [34], this procedure was used to define the relation kernels in the divisive normalization and thus equalize the energy of neuron responses locally.

4. Extensions and Variants

This section is devoted to describing several generalizations and variants of the ideas provided in the previous sections.

4.1. Use of Generic Regression Methods

The use of linear kernel smoothers as regression methods is not mandatory. Indeed, the ideas previously described can be employed even applying different regression methods. Below, we provide some general steps in the same fashion of a pseudo-code. In order to clarify the application of possible different regression techniques,

1. Choose a regression method. Obtain a trend function $\hat{f}(\mathbf{x}) \approx E[y|\mathbf{x}]$ given the dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$.
2. Choose one method for estimating the instance variance (between the two below):
 - **M1.** Choose a regression method (the same as in the previous step or a different one). Consider the dataset $\{\mathbf{x}_i, y_i^2\}_{i=1}^N$ and obtain $\hat{q}(\mathbf{x}) \approx E[y^2|\mathbf{x}]$. Then, compute $\hat{v}(\mathbf{x}) = \hat{q}(\mathbf{x}) - (\hat{f}(\mathbf{x}))^2$.
 - **M2.** Choose a regression method (the same as in the previous step, i.e., for the trend, or a different one). Consider the dataset $\{\mathbf{x}_i, v_i\}_{i=1}^N$ where $v_i = (y_i - \hat{f}(\mathbf{x}_i))^2$, and obtain the function $\hat{v}(\mathbf{x})$.

Note that the regression techniques in step 1 and step 2 above can be different.

4.2. Estimation of the Covariance between Two Generic Inputs

In this section, we describe how to approximate the covariance between two generic inputs \mathbf{x} and \mathbf{z} , extending the previous proposed procedures. For the sake of simplicity, we just adapt method M2 in order to estimate the covariance $\hat{c}(\mathbf{x}, \mathbf{z})$ between two generic inputs as

$$\begin{aligned}\hat{c}(\mathbf{x}, \mathbf{z}) &= \sum_{n=1}^N \sum_{i=1}^N \frac{h(\mathbf{x}, \mathbf{x}_n)g(\mathbf{z}, \mathbf{x}_i)}{\sum_{j=1}^N \sum_{k=1}^N h(\mathbf{x}, \mathbf{x}_j)g(\mathbf{z}, \mathbf{x}_k)} (y_n - \hat{f}(\mathbf{x}_n))(y_i - \hat{f}(\mathbf{x}_i)), \\ &= \sum_{n=1}^N \sum_{i=1}^N \varphi(\mathbf{x}, \mathbf{z}, \mathbf{x}_n, \mathbf{x}_i) (y_n - \hat{f}(\mathbf{x}_n))(y_i - \hat{f}(\mathbf{x}_i)),\end{aligned}\quad (7)$$

where

$$\varphi(\mathbf{x}, \mathbf{z}, \mathbf{x}_n, \mathbf{x}_i) = \frac{h(\mathbf{x}, \mathbf{x}_n)g(\mathbf{z}, \mathbf{x}_i)}{\sum_{j=1}^N \sum_{k=1}^N h(\mathbf{x}, \mathbf{x}_j)g(\mathbf{z}, \mathbf{x}_k)},$$

is a bivariate normalized kernel (and h and g two generic kernel functions). Note that $(y_n - \hat{f}(\mathbf{x}_n))(y_i - \hat{f}(\mathbf{x}_i))$ is a one-sample estimator of the covariance between \mathbf{x}_n and \mathbf{x}_i . Similar ideas have been introduced in [45]. Hence, we can also estimate the correlation within the noise in the dataset. This is of interest in different applications [23,29,46].

Remark 7. This section perhaps clarifies how the kernel estimation approach works in general, for possibly estimating different moments and quantities. The idea is to use one-sample estimators involving the data points $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ as ‘anchors’ and, for extending the results at generic inputs, then use linear combinations of these values, weighting them according to the distance from the generic inputs to these anchor points.

4.3. Multi-Output Scenario and Other Extensions

In a multi-output framework, we have N data pairs, such as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$, but, in this case, the outputs are also vectors $\mathbf{y}_i \in \mathbb{R}^{D_Y}$. Hence, for the local trend, we also have a vectorial hidden function $\hat{\mathbf{f}}(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^{D_Y}$ for all possible \mathbf{x} in the input space. Then, we could easily write

$$\hat{\mathbf{f}}(\mathbf{x}) = \sum_{n=1}^N \varphi(\mathbf{x}, \mathbf{x}_n) \mathbf{y}_n. \quad (8)$$

Regarding the estimations of local variances, follow the same procedures, defining the following vectorial quantities: $\hat{\mathbf{q}}(\mathbf{x}) \approx E[\mathbf{y}^2|\mathbf{x}]$ for M1, $\hat{\mathbf{v}}(\mathbf{x}) = \hat{\mathbf{q}}(\mathbf{x}) - (\hat{\mathbf{f}}(\mathbf{x}))^2$ and $\mathbf{v}_n = (\mathbf{y}_n - \hat{\mathbf{f}}(\mathbf{x}_n))^2$ for M2.

One λ for each data point. Furthermore, we could also consider different local hyperparameters λ , for instance λ_n with $n = 1, \dots, N$, or, more generally, with $\lambda = \lambda(\mathbf{x})$. In this case, we would have different coefficient functions $\varphi_n(\mathbf{x}, \mathbf{x}_n) = \varphi_n(\mathbf{x}, \mathbf{x}_n|\lambda_n)$ (one for each input \mathbf{x}_n), so that trend could be easily expressed as

$$\hat{\mathbf{f}}(\mathbf{x}) = \sum_{n=1}^N \varphi_n(\mathbf{x}, \mathbf{x}_n) \mathbf{y}_n. \quad (9)$$

4.4. Adding More Flexibility to the Starting Model

The starting model in Equation (1) is already a flexible non-parametric method. Moreover, if we consider again Equation (9) but in a single output framework, we have

$$\hat{f}(\mathbf{x}) = \sum_{n=1}^N \varphi_n(\mathbf{x}, \mathbf{x}_n) y_n, \quad \text{with} \quad \varphi_n(\mathbf{x}, \mathbf{x}_n) = \varphi_n(\mathbf{x}, \mathbf{x}_n|\lambda_n).$$

Clearly, this is even a much more flexible model than Equation (1), with N hyper-parameters to tune (where N is the number of data; namely, the number of hyper-parameters grows with the number of data). Clearly, intermediate scenarios can be designed where we have a unique kernel function depending on two hyper-parameters,

$$\varphi(\mathbf{x}, \mathbf{x}_n) = \varphi(\mathbf{x}, \mathbf{x}_n | \lambda_1, \lambda_2),$$

or, more generally, on P hyper-parameters, $\varphi(\mathbf{x}, \mathbf{x}_n) = \varphi(\mathbf{x}, \mathbf{x}_n | \lambda_1, \lambda_2, \dots, \lambda_P)$. It is important to remark that, clearly, all these ideas can also be employed in the estimation of the variance (not just for the trend).

4.5. Joint Use of a Rectangular Kernel and a Least Squares Solution for Time Series Analysis

Let us consider a time series framework; i.e., the input is a scalar time instance, $\mathbf{x} = t \in \mathbb{R}$, and then the dataset is formed by the following pairs $\{t_i, y_i\}_{i=1}^N$. Let us also consider that the intervals, $t_i - t_{i-1}$, are constant; in this case, we can skip the t index and consider the dataset $\{t_i, y_i\}_{i=1}^N$. In this scenario, as an alternative, one can use an autoregressive (AR) model,

$$y_i = a_1 y_{i-1} + a_2 y_{i-2} + \dots + a_W y_{i-W} + \epsilon_y, \quad (10)$$

where ϵ_y is a noise perturbation and the coefficients a_i , for $i = 1, \dots, W$, are obtained by a least squares (LSs) minimization and the length of the temporal window W (i.e., the order of the AR filter) is obtained using a spectral information criterion (SIC) [47,48]. Note that temporal window W corresponds to the use of a rectangular kernel in a kernel smoother approach (*but* this scenario only considers the past samples; see the first ‘Remark’ below). Hence, so far, this variant could be considered as a special case of the generic technique in Section 2. However, here, coefficients that mix the past samples are obtained by an LS procedure (not all equals to the same value). Then, considering the estimated coefficients \hat{a}_i by LS, and \hat{W} by SIC, we have

$$\hat{f}(t_i) = \hat{f}_i = \hat{a}_1 y_{i-1} + \hat{a}_2 y_{i-2} + \dots + \hat{a}_{\hat{W}} y_{i-\hat{W}}. \quad (11)$$

Just as an example, in order to apply M2 for estimating the instant variance, we can set $v_i = (y_i - \hat{f}_i)^2$ and assume another AR model over v_i with window length now denoted as H ,

$$v_i = b_1 v_{i-1} + b_2 v_{i-2} + \dots + b_H v_{i-H} + \epsilon_v, \quad (12)$$

where again the estimations \hat{b}_i can be obtained by LS, and \hat{H} by SIC. The resulting instant variance function is

$$v(t_i) = v_i = \hat{b}_1 v_{i-1} + \hat{b}_2 v_{i-2} + \dots + \hat{b}_{\hat{H}} v_{i-\hat{H}}. \quad (13)$$

Clearly, the application of M1 could be performed in a similar way.

Remark 8. The resulting estimators in this section are still a linear combination of (a portion) of the outputs. Therefore, we still have linear smoothers (or better linear filters since only considering combinations of past samples). However, note that in Equations (11)–(13) the coefficients of the linear combinations are obtained by least squares (LSs) method. Hence, we have a window of length W (and then H), but this differs from the use of a rectangular kernel function for two reasons: (a) the window only considers past samples; (b) the coefficients are not all equal (to the ratio 1 over the number of samples included in the window), but they are tuned by LS.

5. Numerical Experiments

5.1. Applications to Time Series

In this section, we consider 5 different numerical experiments with different (true) generating models. Here, we focus on time series ($x = t \in \mathbb{R}$) for two main reasons: (a) first of all, for the sake of simplicity (we can easily show a realization of the data in one plot), (b) and, last but not least, we have relevant benchmark competitors, such as the GARCH models [42,49,50], that we can also test. Note that GARCH models have been specifically designed for handling volatility in time series.

More specifically, in this section, we test our methodologies and different GARCH models in four easy reproducible examples. These four examples differ for the data generating model. Indeed, the data are generated as the sum of a mean function, $f(t)$, and an error term, $s(t)\varepsilon(t)$, i.e.,

$$y(t) = f(t) + s(t)\varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(0, 1). \quad (14)$$

where we have used the fact that we are analyzing time series, i.e., $x = t \in \mathbb{R}$. The functions $f(t)$ and $s(t)$ are deterministic and represent the trend and the standard deviation of the observations $y(t)$, whereas $\varepsilon(t) \sim \mathcal{N}(0, 1)$ represent a standard Gaussian random noise. Note also that $v(t) = s(t)^2$. In order to keep the notation of the rest of the paper, note that the dataset $\{t_n, y_n\}_{n=1}^N$ will be formed by the N pairs,

$$x_n = t_n, \quad y_n = y(t_n),$$

where y_n will be generated according to the model above. The four experiments consider the following:

1. $f(t) = \sin(0.5t)$, $s(t) = 0.2$.
2. $f(t) = t$, $s(t) = 2|t|$.
3. $f(t) = 0.5t^2$, $s(t) = 5\sin(0.3t)$.
4. $f(t) = 0$ and $s(t)$ generated by a GARCH model.

In all cases, the generated data $y(t)$ and the underlying standard deviation $s(t)$ (or variance $v(t) = s(t)^2$) are known; hence, we can compute errors and evaluate the performance of the used algorithms. For the application of the proposed methods, we consider a nonlinear kernel function as

$$h(t, t_n) = \exp\left(-\frac{(t - t_n)^\alpha}{\lambda}\right), \quad (15)$$

with $\alpha \in \{1, 2, 10\}$. The value $\alpha = 1$ is the minimum possible value if we desire to have a suitable L_p distance (with $p = \alpha$), satisfying all the properties of a distance. The choice of $\alpha = 2$ is just in order to consider the typical Euclidean distance. For larger values of α , we set $\alpha = 10$, as well as for avoiding possible numerical problems. The parameter λ is tuned by LOO-CV. We test 7 different combinations, each one mixing a specific trend approximation method with a specific variance estimation technique:

- **Based on kernel smoothers:** we consider different regression methods based on Equations (1) and (15). Furthermore, for the variance estimation, we consider both procedures M1 and M2 described in Section 3. More specifically,
 - **KS-1-1:** $\alpha = 1$ and M1;
 - **KS-1-2:** $\alpha = 1$ and M2;
 - **KS-2-1:** $\alpha = 2$ and M1;
 - **KS-2-2:** $\alpha = 2$ and M2;
 - **KS-10-1:** $\alpha = 10$ and M1;
 - **KS-10-2:** $\alpha = 10$ and M2.

- **Based on AR models:** we apply the method described in Section 4.5, which employs two auto-regressive models, one for the trend and one for the variance, jointly with procedure M2.

Table 1 summarizes all seven considered specific implementations. Moreover, we test several GARCH(p,q) models: GARCH(1,1), GARCH(2,2), GARCH(5,5) and GARCH(10,10) (using a Matlab implementation of GARCH models).

Remark 9. The GARCH models work directly with the signal without trend, i.e., $|y_n - \hat{f}_n|$. In order to have a fair comparison, we use a kernel smoother with $\alpha = 2$, i.e., applying Equations (1)–(15) with $\alpha = 2$ (as in the KS-2 methods).

We recall that GARCH models have been specifically designed for making inferences in heteroscedastic time series.

Table 1. Proposed methods (specific implementations) tested in the numerical experiments.

Method	KS-1-1	KS-1-2	KS-2-1	KS-2-2	KS-10-1	KS-10-2	AR-Based
For trend estimation	Equations (1)–(12), $\alpha = 1$		Equations (1)–(12), $\alpha = 2$		Equations (1)–(12), $\alpha = 10$		AR— Section 4.5
For variance estimation	M1	M2	M1	M2	M1	M2	AR and M2— Section 4.5

5.1.1. Experiment 1

As previously stated, in this first example, we consider

$$y(t) = \sin(0.5t) + 0.2 \varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(0,1),$$

i.e., where $f(t) = \sin(0.5t)$ and $s(t) = 0.2$ (in this case, we have homoscedastic series). We generate $N = 10^4$ data with $t \in [-10, 10]$. We apply all the algorithms, compute the mean absolute error (MAE) and repeat and average the results over 10^4 independent runs. Figure 1a depicts one realization of the generated data with the model above.

Results. Figure 1b,c provide the results in this example. Figure 1b shows the MAE obtained in estimating the trend, and Figure 1c depicts the MAE in estimating the standard deviation. In this case, regarding the trend, we see that the AR-based has an MAE higher than the rest when estimating the trend, but the other MAEs are quite similar. Similar outcomes are also obtained for approximating with the standard deviation, although the smaller MAE values are provided by the GARCH models. Regarding the estimating of the standard deviation, the M1 seems to provide better results than M2 in this example.

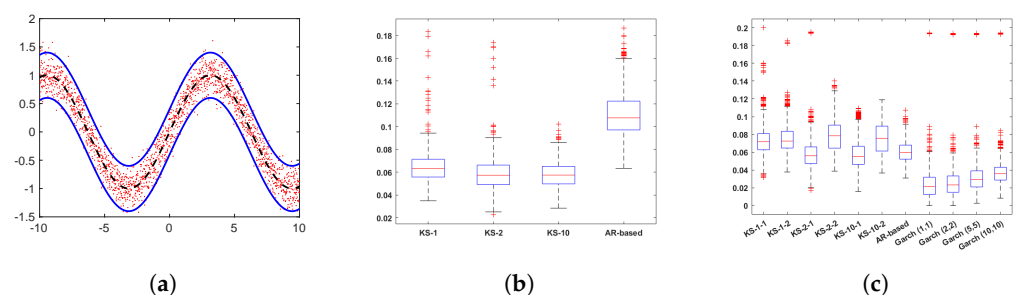


Figure 1. (a) One realization of the data (red dots) generated by the model in Section 5.1.1. The function $f(t)$ is shown dashed black line and the two continue blue lines depict $f(t) \pm 2s(t)$ (containing approx. 95% of the probability mass). (b) MAE in trend estimation. See Table 1. (c) MAE in standard deviation estimation. (b,c) Boxplots of the MAEs of the different algorithms of the experiment in Section 5.1.1.

5.1.2. Experiment 2

In this second example, we consider

$$y(t) = t + 2|t| \varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(0,1),$$

where $f(t) = t$ and $s(t) = 2|t|$ (i.e., in this case, the heteroscedastic scenario). As previously, we generate $N = 10^4$ data with $t \in [-10, 10]$. We apply all the algorithms, compute the mean absolute error (MAE) and again we repeat and average the results over 10^4 independent runs. Figure 2a depicts one realization of the generated data according to the model above.

Results. The results are presented in Figure 2b,c. Again, the MAE for the AR-based method seems to be higher than for the rest of the algorithms when estimating the trend and standard deviation of the time series. Regarding the estimation of the standard deviation, the best results are provided by the kernel methods, which use the M2 (the MAE is particularly small with $\alpha = 10$). Thus, it is remarkable that, even in this *heteroscedastic scenario*, the proposed methods provide similar, and, in some cases, better, results than GARCH models.

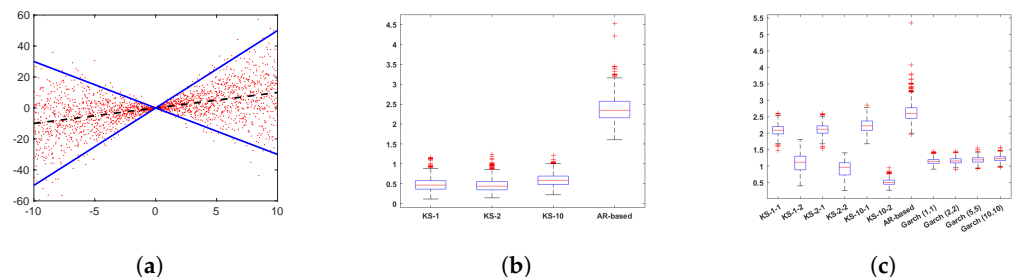


Figure 2. (a) One realization of the data (red dots) generated by the model in Section 5.1.2. The function $f(t)$ is shown dashed black line and the two continuous blue lines depict $f(t) \pm 2s(t)$ (containing approximately 95% of the probability mass). (b) MAE in trend estimation. See Table 1. (c) MAE in trend standard deviation estimation. (b,c) Boxplots of the MAEs of the different algorithms of the experiment in Section 5.1.2.

5.1.3. Experiment 3

In this third example, we consider a standard deviation that varies periodically, and a second-order polynomial as a trend function, i.e.,

$$y(t) = 0.5t^2 + 5\sin(0.3t) \varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(0,1),$$

where $f(t) = 0.5t^2$ and $s(x) = 5\sin(0.3t)$. Again, we have heteroscedastic scenario. As previously, we generate $N = 10^4$ data with $t \in [-10, 10]$. We compute the mean absolute error (MAE), and again we repeat and average the results over 10^4 independent runs. Figure 3a depicts one realization of the generated data according to the described model.

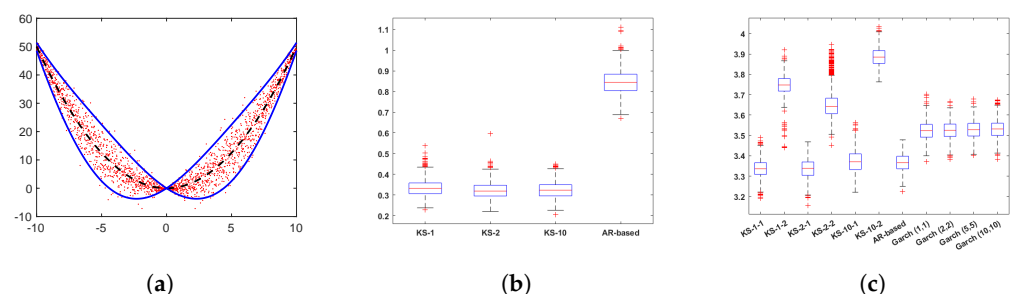


Figure 3. (a) One realization of the data (red dots) generated by the model in Section 5.1.3. The function $f(t)$ is shown as a dashed black line and the two continuous blue lines depict $f(t) \pm 2s(t)$ (containing

approximately 95% of the probability mass). (b) MAE in trend estimation. See Table 1. (c) MAE in standard deviation estimation. (b,c) Boxplots of the MAEs of the different algorithms of the experiment in Section 5.1.3.

Results. Figure 3b,c depict the boxplots of the MAE in estimating the trend and the standard deviation. As with the rest of the examples, the AR-based provides the worst results when calculating the trend of the time series, while the rest of the algorithms have very similar error values (recall that the GARCH models employ the KS-2 method for the trend estimation). However, the AR-based method provides good performance in the approximation of the standard deviation. The AR-based technique uses M2 for the variance. Moreover, for the standard deviation estimation, the methods KS-1-1, KS-2-1 and KS-10-1 provide smaller errors. Hence, it seems that in this case M1 works better even than GARCH models.

5.1.4. Experiment 4

Let us consider

$$y(t) = s(t) \varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(0, 1),$$

$$y_t = s_t \varepsilon_t = z_t,$$

i.e., $f(t) = f_t = 0$, and we have denoted $z_t = s_t \varepsilon_t$. The process for the standard deviation $s_t = \sqrt{v_t}$ will be a GARCH(P, Q), i.e.,

$$v_t = \kappa + \gamma_1 v_{t-1} + \dots + \gamma_P v_{t-P} +$$

$$+ \alpha_1 z_{t-1}^2 + \dots + \alpha_Q z_{t-Q}^2.$$

In this example, we generate the variance from a GARCH(2, 3)

$$v_t = 2 + 0.2 \cdot v_{t-1} + 0.1 \cdot v_{t-2} + 0.2 \cdot z_{t-1}^2 +$$

$$+ 0.2 \cdot z_{t-2}^2 + 0.2 \cdot z_{t-3}^2 \quad (16)$$

Clearly, again, it is a heterostochastic scenario. We generate $N = 10^4$ data and apply the different techniques. We compute the mean absolute error (MAE), and we repeat and average the results over 10^4 independent runs. One realization of the generated data is provided in Figure 4a. Note that the trend is linear, but the volatility presents very fast changes since it generated by a GARCH(2,3) model (see blue line in Figure 4a).

Results. The results for this example are presented in Figure 4b,c. The best performance in the estimation of $s(t)$ is obtained by KS-1-2 and all the GARCH models (as expected in this case). The AR-based and KS-2-2 also provided close values of MAE. In this example, clearly, M2 outperforms M1. We remark that, even in this scenario that is particularly favorable for the GARCH models, all the proposed methods provide competitive results. The highest (hence worst) MAEs values are provided by the KS with $\alpha = 10$, which is a quite extreme tuning/choice of this parameter, especially when the volatility follows an autoregressive rule. Indeed, a Laplacian kernel with $\alpha = 1$ is more adequate in this framework. For further considerations, see ([28] Section 10).

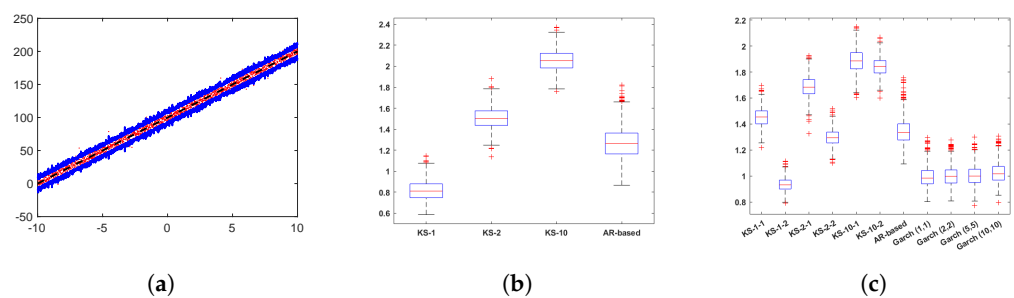


Figure 4. (a) One realization of the data (red dots) generated by the model in Section 5.1.4. The linear trend $f(t) = t$ is shown as a dashed black line, whereas the two continuous blue lines depict $f(t) \pm 2s(t)$, where $s(t)$ is generated by a GARCH model. (b) MAE in trend estimation. See Table 1. (c) MAE in standard deviation estimation. (b,c) Boxplots of the MAEs of the different algorithms of the experiment in Section 5.1.4.

5.1.5. Final Comments about the Applications to Time Series

All the proposed methods based on kernel smoothers (KS) and M1 and M2 always provide competitive results (with closer or smaller MAE values) with respect to the benchmark algorithms, such as GARCH methods, which have been specifically designed for the estimation of the standard deviation in time series.

These numerical experiments show that the proposed M1 and M2 can even outperform GARCH models regarding the estimation of the standard deviation in time series. The GARCH models seem to provide more robust results when the signal has been generated truly by a GARCH model in Section 5.1.4 and, surprisingly, in the homoscedastic scenario in Section 5.1.1. The methods based on kernel smoothers virtually outperform the AR-based approach due to the fact that they incorporate the “future samples” in their estimators. A rectangular window, including both past and future samples, should improve the performance of the AR-based method.

5.2. Application in Higher Input Dimension: Real Data on Soundscape Emotions

GARCH models have been specifically designed for estimating the volatility in time series; hence, $x = t \in \mathbb{R}$. However, the methods proposed in this work can be applied to problems with multidimensional inputs $\mathbf{x} \in \mathbb{R}^D$ with $D \geq 1$.

More specifically, here, we focus on analyzing a soundscape emotion database. In the last decade, soundscapes have become one of the most active topics in acoustics. In fact, the number of related research projects and scientific articles grows exponentially [51–53]. In urban planning and environmental acoustics, the general scheme consists of (a) soundscapes recording, (b) computing of acoustic and psychoacoustic indicators of the signals, (c) including other context indicators (e.g., visual information [54]) and (d) ranking of soundscapes’ audio signals employing emotional descriptors. Finally, the model can be developed [40].

Here, we consider the emo-soundscapes database (EMO) [43,44], where $N = 1200$ and $D = 122$, i.e., $\mathbf{x} \in \mathbb{R}^{122}$, which is the largest available soundscape database with annotations of emotion labels, and also the most recent up to now [43,44]. Among the 122 audio features that form the inputs \mathbf{x} , we can distinguish three main groups of features of the audio signals: psychoacoustic features, time-domain features and frequency-domain features. The output y that we consider is called *arousal*. For further information, the complete database can be found at <https://metacreation.net/emo-soundscapes/> (accessed on 1 January 2024). We again employ a kernel function of type

$$h(\mathbf{x}, \mathbf{z}) = h(\mathbf{x}, \mathbf{z} | \lambda) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^\alpha}{\lambda}\right),$$

with $\alpha = 2$. Therefore, considering the EMO database $\{\mathbf{x}_i, y_i\}_{i=1}^{1200}$, we apply M1 and M2 and CV-LOO for tuning λ . The MAE in estimating the trend is 0.0388, obtained with an

optimal parameter $\lambda^* = 0.12$. Moreover, in both cases, we find a sensible increase in the volatility of the output in the last half of the samples, for i from 621 to 1213. This could be an interesting result, which should be discussed with experts in the field and regarding the EMO database.

6. Conclusions

In this work, we have proposed two methods for estimating the instant/local variance in regression problems based on kernel smoothers. From another point of view, this work can also be viewed as a way to perform quantile regression (at least a second-order quantile regression) using kernel smoothers.

With respect to the other procedures noted in the literature, the proposed methods have much wider applicability than the other techniques since they can be employed only for a time series, such as the well-known GARCH models (where the time index is a scalar number, $x = t \in \mathbb{R}$). Indeed, the proposed schemes can be employed with multidimensional input variables $\mathbf{x}_t \in \mathbb{R}^D$. Moreover, the proposed techniques can also be applied to time series ($D = 1$) and can also obtain a prediction between two consecutive time instants (i.e., with a higher resolution than the received data). More generally, the estimation of the local variance can occur regarding any of the training data \mathbf{x}_i but also regarding any generic test input \mathbf{x} .

Furthermore, analyzing the time series, the numerical simulations have shown that the proposed schemes provide competitive results even with respect to GARCH models (which are specifically designed for analyzing time series). Applications regarding real data and multidimensional inputs (with dimension 122) have also been provided. As a future research direction, we plan to design kernel smoother functions with local parameters λ , or more generally with $\lambda = \lambda(\mathbf{x})$, to better handle the heteroscedasticity. Moreover, applications regarding multi-output scenarios can be designed and considered.

Author Contributions: All authors have contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Young Researchers R&D Project, ref. num. F861 (AUTO-BA-GRAPH), funded by Community of Madrid and Rey Juan Carlos University, the Agencia Estatal de Investigación AEI (project SP-GRAPH, ref. num. PID2019-105032GB-I00), and by MICIIN/FEDER/UE, Spain, under Grant PID2020-118071GB-I00.

Data Availability Statement: Data available on request from the authors.

Acknowledgments: We thank the Reviewers for their effort.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Engle, R. Risk and volatility: Econometric models and financial practice. *Am. Econ. Rev.* **2004**, *94*, 405–420. [\[CrossRef\]](#)
2. Chang, C.; McAleer, M.; Tansuchat, R. Modelling long memory volatility in agricultural commodity futures returns. *Ann. Financ. Econ.* **2012**, *7*, 1250010. [\[CrossRef\]](#)
3. Dedi, L.; Yavas, B. Return and volatility spillovers in equity markets: An investigation using various GARCH methodologies. *Cogent Econ. Financ.* **2016**, *4*, 1266788. [\[CrossRef\]](#)
4. Ibrahim, B.; Elamer, A.; Alasker, T.; Mohamed, M.; Abdou, H. Volatility contagion between cryptocurrencies, gold and stock markets pre-and-during COVID-19: Evidence using DCC-GARCH and cascade-correlation network. *Financ. Innov.* **2024**, *10*, 104. [\[CrossRef\]](#)
5. Engle, R.F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econom. J. Econom. Soc.* **1982**, *50*, 987–1007. [\[CrossRef\]](#)
6. Bollerslev, T. A conditionally heteroskedastic time series model for speculative prices and rates of return. *Rev. Econ. Stat.* **1987**, *6*, 542–547. [\[CrossRef\]](#)
7. Taylor, S.J. Modeling stochastic volatility: A review and comparative study. *Math. Financ.* **1994**, *4*, 183–204. [\[CrossRef\]](#)
8. Martino, L.; Yang, H.; Luengo, D.; Kannianen, J.; Corander, J. A fast universal self-tuned sampler within Gibbs sampling. *Digit. Signal Process.* **2015**, *47*, 68–83. [\[CrossRef\]](#)

9. Morgan, J.; Co. Incorporated. *Creditmetrics-Technical Document*; JP Morgan: New York, NY, USA, 1997; Volume 1, pp. 102–127. Available online: http://www.angelvila.eu/documents/CREDITMETRICSDOCUMENTOTECNICO_000.pdf (accessed on 1 January 2024).
10. Malmsten, H. Evaluating Exponential GARCH Models. Technical Report, SSE/EFI Working Paper Series in Economics and Finance. 2004. Available online: <https://www.econstor.eu/bitstream/10419/56143/1/394140834.pdf> (accessed on 1 January 2024).
11. Carnero, M.A.; Peña, D.; Ruiz, E. Persistence and kurtosis in GARCH and stochastic volatility models. *J. Financ. Econom.* **2004**, *2*, 319–342. [[CrossRef](#)]
12. Asai, M.; McAleer, M. Dynamic asymmetric leverage in stochastic volatility models. *Econom. Rev.* **2005**, *24*, 317–332. [[CrossRef](#)]
13. Asai, M.; McAleer, M.; Yu, J. Multivariate stochastic volatility: A review. *Econom. Rev.* **2006**, *25*, 145–175. [[CrossRef](#)]
14. Hansen, P.R.; Lunde, A. Realized variance and market microstructure noise. *J. Bus. Econ. Stat.* **2006**, *24*, 127–161. [[CrossRef](#)]
15. Barndorff-Nielsen, O.E.; Shephard, N. Variation, jumps and high frequency data in financial econometrics. In *Advanced in Economics and Econometrics. Theory and Applications*; Cambridge University Press: Cambridge, UK, 2006.
16. Fan, J.; Yao, Q. Efficient estimation of conditional variance functions in stochastic regression. *Biometrika* **1998**, *85*, 645–660. [[CrossRef](#)]
17. Yu, K.; Jones, M.C. Likelihood-Based Local Linear Estimation of the Conditional Variance Function. *J. Am. Stat. Assoc.* **2004**, *99*, 139–144. [[CrossRef](#)]
18. Ruppert, D.; Wand, M.P.; Holst, U.; Hössjer, O. Local Polynomial Variance-Function Estimation. *Technometrics* **1997**, *39*, 262–273. [[CrossRef](#)]
19. Derman, E.; Kani, I.; Zou, J.Z. The Local Volatility Surface: Unlocking the Information in Index Option Prices. *Financ. Anal. J.* **1996**, *52*, 25–36. [[CrossRef](#)]
20. Zu, Y.; Boswijk, H.P. Estimating spot volatility with high-frequency financial data. *J. Econom.* **2014**, *181*, 117–135. [[CrossRef](#)]
21. Cheng, H. Second Order Model with Composite Quantile Regression. *J. Phys. Conf. Ser.* **2023**, *2437*, 012070. [[CrossRef](#)]
22. Huang, A.Y.; Peng, S.P.; Li, F.; Ke, C.J. Volatility forecasting of exchange rate by quantile regression. *Int. Rev. Econ. Financ.* **2011**, *20*, 591–606. [[CrossRef](#)]
23. Martino, L.; Llorente, F.; Curbelo, E.; López-Santiago, J.; Míguez, J. Automatic Tempered Posterior Distributions for Bayesian Inversion Problems. *Mathematics* **2021**, *9*, 784. [[CrossRef](#)]
24. Baur, D.G.; Dimpfl, T. A quantile regression approach to estimate the variance of financial returns. *J. Financ. Econom.* **2019**, *17*, 616–644. [[CrossRef](#)]
25. Chronopoulos, I.C.; Raftapostolos, A.; Kapetanios, G. Forecasting Value-at-Risk using deep neural network quantile regression. *J. Financ. Econom.* **2023**. [[CrossRef](#)]
26. Huang, Q.; Zhang, H.; Chen, J.; He, M. Quantile regression models and their applications: A review. *J. Biom. Biostat.* **2017**, *8*, 1–6. [[CrossRef](#)]
27. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: Berlin/Heidelberg, Germany, 2006.
28. Martino, L.; Read, J. Joint introduction to Gaussian Processes and Relevance Vector Machines with Connections to Kalman filtering and other Kernel Smoothers. *Inf. Fusion* **2021**, *74*, 17–38. [[CrossRef](#)]
29. Altman, N.S. Kernel Smoothing of Data with Correlated Errors. *J. Am. Stat. Assoc.* **1990**, *85*, 749–759. [[CrossRef](#)]
30. Bentley, J.L.; Stanat, D.F.; Williams, E.H. The complexity of finding fixed-radius near neighbors. *Inf. Process. Lett.* **1977**, *6*, 209–212. [[CrossRef](#)]
31. Carandini, M.; Heeger, D. Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* **2012**, *13*, 51–62. [[CrossRef](#)] [[PubMed](#)]
32. Malo, J.; Laparra, V. Psychophysically tuned divisive normalization approximately factorizes the PDF of natural images. *Neural Comput.* **2010**, *22*, 3179–3206. [[CrossRef](#)] [[PubMed](#)]
33. Ballé, J.; Laparra, V.; Simoncelli, E. Density modeling of images using a generalized normalization transformation. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, NA, USA, 2–4 May 2016.
34. Laparra, V.; Ballé, J.; Berardino, A.; Simoncelli, E. Perceptual image quality assessment using a normalized Laplacian pyramid. *Electron. Imaging* **2016**, *28*, 1–6. [[CrossRef](#)]
35. Laparra, V.; Berardino, A.; Ballé, J.; Simoncelli, E. Perceptually optimized image rendering. *J. Opt. Soc. Am. A* **2017**, *34*, 1511. [[CrossRef](#)]
36. Hernandez-Camara, P.; Vila-Tomas, J.; Laparra, V.; Malo, J. Neural networks with divisive normalization for image segmentation. *Pattern Recognit. Lett.* **2023**, *173*, 64–71. [[CrossRef](#)]
37. Hall, P.; Carroll, R.J. Variance function estimation in regression: The effect of estimating the mean. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1989**, *51*, 3–14. [[CrossRef](#)]
38. Newey, W.K. Kernel estimation of partial means and a general variance estimator. *Econom. Theory* **1994**, *10*, 1–21. [[CrossRef](#)]
39. Chib, S.; Greenberg, E. On conditional variance estimation in nonparametric regression. *Stat. Comput.* **2013**, *23*, 261–270. [[CrossRef](#)]

40. Millan-Castillo, R.S.; Martino, L.; Morgado, E.; Llorente, F. An Exhaustive Variable Selection Study for Linear Models of Soundscape Emotions: Rankings and Gibbs Analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2460–2474. [\[CrossRef\]](#)
41. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006; pp. 1–248.
42. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [\[CrossRef\]](#)
43. Fan, J.; Thorogood, M.; Pasquier, P. Emo-soundscapes: A dataset for soundscape emotion recognition. In Proceedings of the 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, TX, USA, 23–26 October 2017; pp. 196–201.
44. Fonseca, E.; Pons Puig, J.; Favory, X.; Font Corbera, F.; Bogdanov, D.; Ferraro, A.; Oramas, S.; Porter, A.; Serra, X. Freesound datasets: A platform for the creation of open audio datasets. In Proceedings of the 18th ISMIR Conference, Suzhou, China, 23–27 October 2017; Hu, X., Cunningham, S.J., Turnbull, D., Duan, Z., Eds.; International Society for Music Information Retrieval (ISMIR); pp. 486–493.
45. Pallini, A. Kernel Methods For Estimating Covariance Functions From Curves. In Proceedings of the Classification and Data Analysis, Pescara, Italy, 3–4 July 1999; pp. 319–326.
46. Curbelo, E.; Martino, L.; Llorente, F.; Delgado-Gomez, D. Adaptive Posterior Distributions for Uncertainty Analysis of Covariance Matrices in Bayesian Inversion Problems for Multioutput Signals. 2023. Available online: <https://vixra.org/pdf/2310.0032v2.pdf> (accessed on 1 January 2024).
47. Martino, L.; San Millan-Castillo, R.; Morgado, E. Spectral information criterion for automatic elbow detection. *Expert Syst. Appl.* **2023**, *231*, 120705. [\[CrossRef\]](#)
48. Morgado, E.; Martino, L.; Millan-Castillo, R.S. Universal and automatic elbow detection for learning the effective number of components in model selection problems. *Digit. Signal Process.* **2023**, *140*, 104103. [\[CrossRef\]](#)
49. Hansen, P.R.; Lunde, A. A forecast comparison of volatility models: Does anything beat a GARCH (1, 1)? *J. Appl. Econom.* **2005**, *20*, 873–889. [\[CrossRef\]](#)
50. Trapero, J.R.; Cardos, M.; Kourentzes, N. Empirical safety stock estimation based on kernel and GARCH models. *Omega* **2019**, *84*, 199–211. [\[CrossRef\]](#)
51. Aletta, F.; Xiao, J. What are the current priorities and challenges for (urban) soundscape research? *Challenges* **2018**, *9*, 16. [\[CrossRef\]](#)
52. Lundén, P.; Hurtig, M. On urban soundscape mapping: A computer can predict the outcome of soundscape assessments. In Proceedings of the INTER-NOISE and NOISE-CON Congress and Conference Proceedings, Hamburg, Germany, 21–24 August 2016; Institute of Noise Control Engineering: Washington, DC, USA, 2016; Volume 253, pp. 2017–2024. Available online: <https://www.diva-portal.org/smash/get/diva2:1059673/FULLTEXT01.pdf> (accessed on 1 January 2024).
53. Lionello, M.; Aletta, F.; Kang, J. A systematic review of prediction models for the experience of urban soundscapes. *Appl. Acoust.* **2020**, *170*, 107479. [\[CrossRef\]](#)
54. Axelsson, O.; Nilsson, M.E.; Berglund, B. A principal components model of soundscape perception. *J. Acoust. Soc. Am.* **2010**, *128*, 2836–2846. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.