This is a draft version of work in progress, content will be revisited in subsequent versions.

Robust Local Polynomial Regression with Similarity Kernels

Yaniv Shulman

yaniv@shulman.info

Abstract

Local Polynomial Regression (LPR) is a widely used nonparametric method for modeling complex relationships due to its flexibility and simplicity. It estimates a regression function by fitting low-degree polynomials to localized subsets of the data, weighted by proximity. However, traditional LPR is sensitive to outliers and high-leverage points, which can significantly affect estimation accuracy. This paper revisits the kernel function used to compute regression weights and proposes a novel framework that incorporates both predictor and response variables in the weighting mechanism. By introducing two positive definite kernels, the proposed method robustly estimates weights, mitigating the influence of outliers through localized density estimation. The method is implemented in Python and is publicly available at `https://github.com/yaniv-shulman/rsklpr`, demonstrating competitive performance in synthetic benchmark experiments. Compared to standard LPR, the proposed approach consistently improves robustness and accuracy, especially in heteroscedastic and noisy environments, without requiring multiple iterations. This advancement provides a promising extension to traditional LPR, opening new possibilities for robust regression applications.

1. Introduction

Local polynomial regression (LPR) is a powerful and flexible statistical technique that has gained increasing popularity in recent years due to its ability to model complex relationships between variables. Local polynomial regression generalizes the polynomial regression and moving average methods by fitting a low-degree polynomial to a nearest neighbors subset of the data at the location. The polynomial is fitted using weighted ordinary least-squares, giving more weight to nearby points and less weight to points farther away. The value of the regression function for the point is then obtained by evaluating the fitted local polynomial using the predictor variable value for that data point. LPR has good accuracy near the boundary and performs better than all other linear smoothers in a minimax sense [2]. The biggest advantage of this class of methods is not requiring a prior specification of a function i.e. a parameterized model. Instead, only a small number of hyperparameters need to be specified such as the type of kernel, a smoothing parameter and the degree of the local polynomial. The method is therefore suitable for modeling complex processes such as non-linear relationships, or complex dependencies for which no theoretical models exist. These two advantages, combined with the simplicity of the method, makes it one of the most attractive of the modern regression methods for applications that fit the general framework of least-squares regression but have a complex deterministic structure.

Local polynomial regression incorporates the notion of proximity in two ways. The first is that a smooth function can be reasonably approximated in a local neighborhood by a simple function such as a linear or low order polynomial. The second is the assumption that nearby points carry more importance in the calculation of a simple local approximation or alternatively, that closer points are more likely to interact in simpler ways than far away points. This is achieved by a kernel which produces values that diminish as the distance between the explanatory variables increase to model stronger relationship between closer points.

Methods in the LPR family include the Nadaraya-Watson estimator [10, 18] and the estimator proposed by Gasser and Müller [7] which both perform kernel-based local constant fit. These were improved on in terms of asymptotic bias by the proposal of the local linear and more general local polynomial estimators [16, 3, 9, 4, 5]. For a review of LPR methods the interested reader is referred to [2].

LPR is however susceptible to outliers, high leverage points and functions with discontinuities in their derivative which often cause an adverse impact on the regression due to its use of least-squares based optimization [17]. The use of unbounded loss functions may result in anomalous observations severely affecting the local estimate. Substantial work has been done to develop algorithms to apply LPR to difficult data. To alleviate the issue [15] employs variable bandwidth to exclude observations for which residuals from the resulting estimator are large. In [3] an iterated weighted fitting procedure is proposed that assigns in each consecutive iteration smaller weights to points that are farther then the fitted values at the previous iteration. The process repeats for a number of iterations and the final values are considered the robust parameters and fitted values. An alternative common approach is to replace the squared prediction loss by one that is more robust to the presence of large residuals by increasing more slowly or a loss that has an upper bound such as the Tukey or Huber loss. These methods however require specifying a threshold parameter for the loss to indicate atypical observations or standardizing the errors using robust estimators of scale [8]. For a recent review of robust LPR and other nonparametric methods see [17, 11]

The main contribution of this paper is to revisit the kernel used to produce regression weights. The simple yet effective idea is to generalize the kernel such that both the predictor and the response are used to calculate weights. Within this framework, two positive definite kernels are proposed that assign robust weights to mitigate the adverse effect of outliers in the local neighborhood by estimating the density of the response at the local locations. Note the proposed framework does not preclude the use of robust loss functions, robust bandwidth selectors and standardization techniques. In addition the method is implemented in the Python programming language and is made publicly available. Experimental results on synthetic benchmarks demonstrate that the proposed method achieves competitive results and generally performs better than LOWESS using only a single training iteration.

The remainder of the paper is organized as follows: In Section 2, a brief overview of the mathematical formulation of local polynomial regression is provided. In Section 3, a framework for robust weights as well as specific robust positive definite kernels are proposed. Section 4 provides an analysis of the estimator and a discussion of its properties. In Section 5, implementation notes and experimental results are provided. Finally, in Section 6, the paper concludes with directions for future research.

2. Local Polynomial Regression

This section provides a brief overview of local polynomial regression and establishes the notation subsequently used. Let (X, Y) be a random pair and $\mathcal{D}_T = \{(X_i, Y_i)\}_{i=1}^{T} \subseteq \mathcal{D}$ be a training set comprising a sample of T data pairs. Suppose that $(X, Y) \sim f_{XY}$ a continuous density and $X \sim f_X$ the marginal distribution of X. Let $Y \in \mathbb{R}$ be a continuous response and assume a model of the form $Y_i = m(X_i) + \epsilon_i,\ i \in 1, \ldots, T$ where $m(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is an unknown function and ϵ_i are independently distributed error terms having zero mean representing random variability not included in X_i such that $\mathbb{E}[Y \mid X = x] = m(x)$. There are no global assumptions about the function $m(\cdot)$ other than that it is smooth and that locally it can be well approximated by a low degree polynomial as per Taylor's theorem. Local polynomial regression is a class of nonparametric regression methods that estimate the unknown regression function $m(\cdot)$ by combining the classical least-squares method with the versatility of non-linear regression. The local p-th order Taylor expansion for $x \in \mathbb{R}$ near a point X_i yields:

$$m(X_i) \approx \sum_{j=0}^{p} \frac{m^{(j)}(x)}{j!}(x - X_i)^j := \sum_{j=0}^{p} \gamma_j(x)(x - X_i)^j \tag{1}$$

To find an estimate $\hat{m}(x)$ of $m(x)$ the low-degree polynomial (1) is fitted to the N nearest neighbors using weighted least-squares such to minimize the empirical loss $\mathcal{L}_{lpr}(\cdot\,; \mathcal{D}_N, h)$:

$$\mathcal{L}_{lpr}(x; \mathcal{D}_N, h) := \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \gamma_j(x)(x - X_i)^j \right)^2 K_h(x - X_i) \tag{2}$$

$$\hat{\gamma}(x) := \min_{\gamma(x)} \mathcal{L}_{lpr}(x; \mathcal{D}_N, h) \tag{3}$$

Where $\gamma, \hat{\gamma} \in \mathbb{R}^{p+1}$; $K_h(\cdot)$ is a scaled kernel, $h \in \mathbb{R}_{>0}$ is the bandwidth parameter and $\mathcal{D}_N \subseteq \mathcal{D}_T$ is the subset of N nearest neighbors of x in the training set where the distance is measured on the predictors only. Having computed $\hat{\gamma}(x)$ the estimate of $\hat{m}(x)$ is taken as $\hat{\gamma}(x)_1$. Note the term kernel carries here the meaning typically used in the context of nonparametric regression i.e. a non-negative real-valued weighting function that is typically symmetric, unimodal at zero, integrable with a unit integral and whose value is non-increasing for the increasing distance between the X_i and x. Higher degree polynomials and smaller N generally increase the variance and decrease the bias of the estimator and vice versa [2]. For derivation of the local constant and local linear estimators for the multidimensional case see [6].

Remark on Nearest Neighbors and Bandwidth.. In the following, the local neighborhood is defined by taking the N nearest neighbors to x. Thus, $\mathcal{D}_N \subseteq \mathcal{D}_T$ contains exactly N points. Then, a distance-based kernel K_h (with bandwidth h) is used to weight those neighbors, such that nearer points receive larger weights. In the experiments, h is chosen or scaled in accordance with the distribution of the distances within $\mathcal{D}_N$. This approach combines a fixed-sized local subset (via N) with a variable kernel scaling (via h), ensuring stable local fits even in heterogeneous data scenarios.

3. Robust Weights with Similarity Kernels

The main idea presented is to generalize the kernel function used in equation (2) to produce robust weights. This is achieved by using a similarity kernel function defined on the data domain $K_{\mathcal{D}} : \mathcal{D}^2 \to \mathbb{R}_+$ that enables weighting each point and incorporating information on the data in the local neighborhood in relation to the local regression target.

$$\mathcal{L}_{rsk}(x, y; \mathcal{D}_N, H) := \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x, y)(x - X_i)^j \right)^2 K_{\mathcal{D}}\left((x, y), (X_i, Y_i); H\right) \tag{4}$$

$$\hat{\beta}(x, y; \mathcal{D}_N, H) := \min_{\beta(x,y)} \mathcal{L}_{rsk}(x, y; \mathcal{D}_N, H) \tag{5}$$

Where H is the set of bandwidth parameters. There are many possible choices for such a similarity kernel to be defined within this general framework. However, used as a local weighting function, such a kernel should have the following attributes:

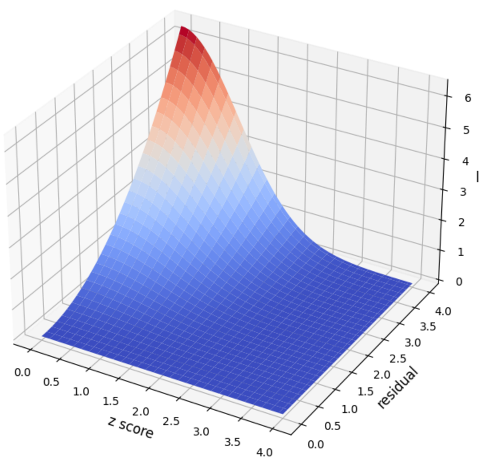
1. Non-negative, $K_{\mathcal{D}}((x, y), (x', y') \geq 0$.
2. Symmetry in the inputs, $K_{\mathcal{D}}((x, y), (x', y')) = K_{\mathcal{D}}((x', y'), (x, y))$.
3. Tending toward decreasing as the distance in the predictors increases. That is, given a similarity function on the response $s(\cdot, \cdot) : \mathbb{R}^2 \to \mathbb{R}_+$, if $s(y, y')$ indicates high similarity the weight should decrease as the distance between the predictors grows, $s(y, y') > \alpha \implies K_{\mathcal{D}}((x, y), (x + u, y')) \geq K_{\mathcal{D}}((x, y), (x + v, y'))\ \ \forall\ \|u\| \leq \|v\|$ and some $\alpha \in \mathbb{R}_+$.

In this work two such useful positive definite kernels are proposed. Similarly to the usual kernels used in (2), these tend to diminish as the distance between the explanatory variables increases to model stronger relationship between closer points. In addition, the weights produced by the kernels also model the "importance" of the pair (x, y). This is useful for example to down-weight outliers to mitigate their adverse effect on the ordinary least square based regression. Formally let $K_{\mathcal{D}}$ be defined as:

$$K_{\mathcal{D}}\Big((x, y), (x', y'); H_1, H_2\Big) = K_1(x, x'; H_1) K_2\Big((x, y), (x', y'); H_2\Big) \tag{6}$$

Where $K_1 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ and $K_2 : \mathcal{D}^2 \to \mathbb{R}_+$ are positive definite kernels and H_1, H_2 are the sets of bandwidth parameters. The purpose of K_1 is to account for the distance between a neighbor to the local regression target and therefore may be chosen as any of the kernel functions that are typically used in equation (2). The role of K_2 is described now in more detail as this is the main idea proposed in this work. Using K_2, the method performs robust regression by detecting local outliers in an unsupervised manner and assigns them with lower weights. There are many methods that could be employed to estimate the extent to which a data point is a local outlier however in this work it is estimated in one of the following two ways.

Figure 1: Loss function, assuming a standard quadratic function of the residual, a standard normal density for K_2 and excluding the K_1 distance kernel scaling.



Conditional Density

The first proposed method for K_2 is proportional to the estimated localized conditional marginal distribution of the response variable at the location:

$$K_2\Big((x, y), (x', y'); H_2\Big) = \hat{f}(y \mid x; H_2)\hat{f}(y' \mid x'; H_2) \tag{7}$$

The nonparametric conditional density estimation is performed using the Parzen–Rosenblatt window (kernel density estimator):

$$\hat{f}(y \mid x; H_2) = \hat{f}(x, y; H_2)/\hat{f}(x; H_2) \tag{8}$$

$$= \hat{f}(v; \mathbf{H}_v)/\hat{f}(x; \mathbf{H}_x) \tag{9}$$

$$= \frac{|\mathbf{H}_x|^{1/2} \sum_{i=1}^{N} K_v\left(\mathbf{H}_v^{-1/2}(v - V_i)\right)}{|\mathbf{H}_v|^{1/2} \sum_{i=1}^{N} K_x\left(\mathbf{H}_x^{-1/2}(x - X_i)\right)} \tag{10}$$

Where $v = [x, y] \in \mathbb{R}^{d+1}$ is the concatenated vector of the predictors and the response; and $\mathbf{H}_v, \mathbf{H}_x$ are bandwidth matrices.

Joint Density

The second proposed kernel is proportional to the joint distribution of the random pair, this could be useful for example to also down-weight high leverage points:

$$K_2\left((x, y), (x', y'); H_2\right) = \hat{f}(x, y; H_2)\hat{f}(x', y'; H_2) \tag{11}$$

Where the joint density can be estimated using the same aforementioned approach.

Regardless of the choice of kernel, the hyperparameters of this model are similar in essence to the standard local polynomial regression and comprise the span of included points, the kernels and their associated bandwidths. Note that this estimator can be replaced with other robust density estimators and better results are anticipated by doing so however exploring this option is left for future work.

4. Properties

This section discusses some properties of the estimator. Note the notation in this section is simplified by excluding explicit mentions of D_N and H, however the analysis is conditional on the nearest neighbors in the sample, D_N.

4.1. Invariance to y at the Regression Location and Simplification of the Objective

The objective (5) is invariant to the value of y at the location (x, y) for the proposed similarity kernels.

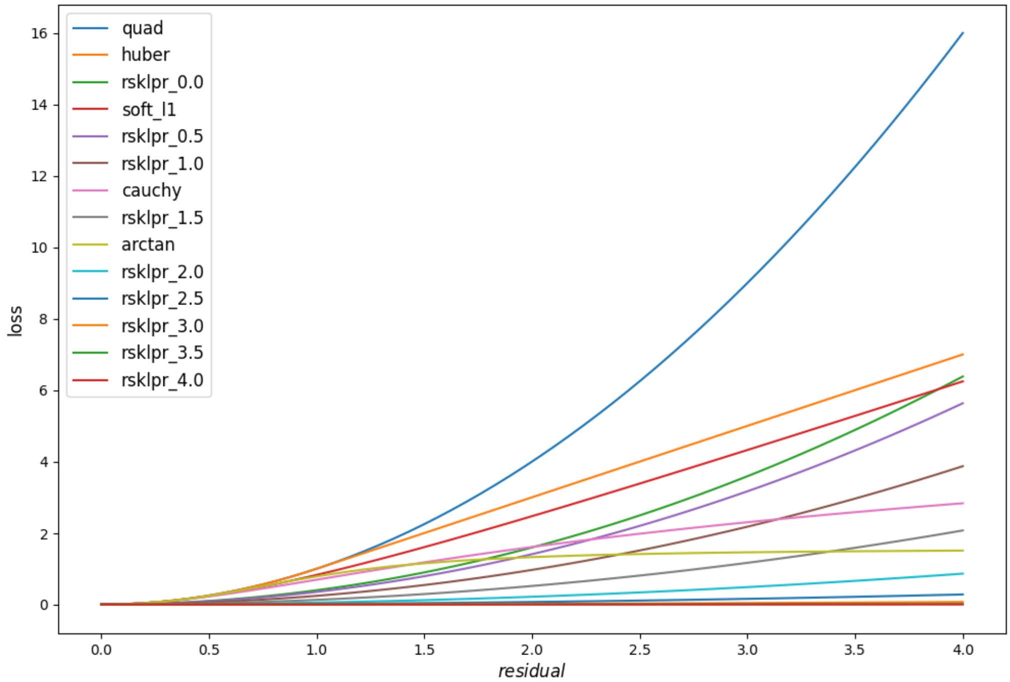
Proof: The optimization is invariant to the scale of the objective function. Therefore:

$$\hat{\beta}(x, y) := \min_{\beta(x,y)} \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x, y)(x - X_i)^j \right)^2 K_{H_1}(x - X_i)\hat{f}(x, y)\hat{f}(X_i, Y_i) \tag{12}$$

$$= \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x, y)(x - X_i)^j \right)^2 K_{H_1}(x - X_i)\hat{f}(X_i, Y_i) \tag{13}$$

The equality holds because $\hat{f}(x, y)$ is a constant scalar that uniformly scales the weights. Since the objective is now independent of y, it follows that:

Figure 2: This figure compares the proposed loss function (rsklpr) at various standard deviation levels with common robust losses (e.g., Huber, Cauchy) and the standard quadratic loss. The attenuation of loss in areas with low-density data demonstrates the enhanced robustness of the proposed method. It is assumed that K_2 is equivalent to the standard Gaussian density and the K_1 distance kernel scaling is excluded. The numbers appended to "rsklpr" indicate the number of standard deviations away from the mean.



1

$$\hat{\beta}(x, y) \coloneqq \min_{\beta(x)} \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i)\hat{f}(X_i, Y_i) \tag{14}$$

$$\coloneqq \hat{\beta}(x) \quad \forall y \tag{15}$$

This simplification enables more efficient calculations of the estimator because the dependence on y is removed from the objective function. Note that $\hat{f}(X_i, Y_i)$ can also be replaced with $\hat{f}(Y_i \mid X_i)$ with similar results.

4.2. *Weighted Arithmetic Mean of the Standard LPR*

The proposed estimator is equivalent to the weighted arithmetic mean of the terms in the standard LPR loss (2), with weights $w_i = \hat{f}(X_i, Y_i)$.

Proof: Since the optimization is invariant to scaling:

$$\hat{\beta}(x) := \min_{\beta(x)} \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i)\hat{f}(X_i, Y_i) \tag{16}$$

$$= \min_{\beta(x)} \left(\sum_{i=1}^{N} \hat{f}(X_i, Y_i) \right)^{-1} \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i)\hat{f}(X_i, Y_i) \tag{17}$$

$$= \min_{\beta(x)} \left(\sum_{i=1}^{N} w_i \right)^{-1} \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i) w_i \tag{18}$$

The normalization by $\sum_{i=1}^{N} w_i$ shows the equivalence to the weighted arithmetic mean, with the weights $w_i = \hat{f}(X_i, Y_i)$. Note the weights can be equivalently replaced with $w_i = \hat{f}(Y_i \mid X_i)$.

4.3. Asymptotic degeneration of the estimator to the standard LPR

Asymptotically, the proposed estimator degenerates to the standard LPR when the weights w_i are uncorrelated with the standard LPR terms. Formally, as $N \to \infty$, $\hat{\beta}(x) \to \hat{\gamma}(x)$, where $\hat{\gamma}(x)$ is the standard LPR estimator, and the condition that $\left(Y - \sum_{j=0}^{p} \beta_j(x)(x - X)^j\right)^2 K_{H_1}(x - X)$ and $\hat{f}(X, Y)$ are uncorrelated holds. It is assumed that (X_i, Y_i) are independent and identically distributed (i.i.d.) random variables and that $\hat{f}(X, Y) > 0$ almost everywhere.

Proof: Define

$$g(X, Y) := \left(Y - \sum_{j=0}^{p} \beta_j(x)(x - X)^j \right)^2 K_{H_1}(x - X),$$

it follows that:

$$\hat{\beta}(x) := \min_{\beta(x)} \left(\sum_{i=1}^{N} \hat{f}(X_i, Y_i) \right)^{-1} \sum_{i=1}^{N} g(X_i, Y_i)\hat{f}(X_i, Y_i) \tag{19}$$

$$= \min_{\beta(x)} \left(\frac{1}{N} \sum_{i=1}^{N} \hat{f}(X_i, Y_i) \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} g(X_i, Y_i)\hat{f}(X_i, Y_i) \right) \tag{20}$$

As $N \to \infty$, by the law of large numbers:

$$\left(\frac{1}{N} \sum_{i=1}^{N} \hat{f}(X_i, Y_i) \right)^{-1} \xrightarrow{a.s.} \frac{1}{\mathbb{E}\left[\hat{f}(X, Y)\right]} \tag{21}$$

$$\frac{1}{N} \sum_{i=1}^{N} g(X_i, Y_i)\hat{f}(X_i, Y_i) \xrightarrow{a.s.} \mathbb{E}\left[g(X, Y)\hat{f}(X, Y)\right] \tag{22}$$

Assuming $\mathbb{E}[\hat{f}(X, Y)] \neq 0$, it follows that:

$$\hat{\beta}(x) \xrightarrow{a.s.} \min_{\beta(x)} \frac{\mathbb{E}\left[g(X, Y)\hat{f}(X, Y)\right]}{\mathbb{E}\left[\hat{f}(X, Y)\right]} \tag{23}$$

If $g(X, Y)$ and $\hat{f}(X, Y)$ are uncorrelated, then:

$$\mathbb{E}\left[g(X, Y)\hat{f}(X, Y)\right] = \mathbb{E}\left[g(X, Y)\right]\mathbb{E}\left[\hat{f}(X, Y)\right] \tag{24}$$

$$\hat{\beta}(x) \xrightarrow{a.s.} \min_{\beta(x)} \mathbb{E}\left[g(X, Y)\right] \tag{25}$$

Therefore, as $N \to \infty$:

$$\hat{\beta}(x) \xrightarrow{a.s.} \min_{\beta(x)} \mathbb{E}\left[\left(Y - \sum_{j=0}^{p} \beta_j(x)(x - X)^j\right)^2 K_{H_1}(x - X)\right] \tag{26}$$

This is the same objective minimized by the standard LPR estimator in the asymptotic sense. Thus, the proposed estimator degenerates to the standard LPR as $N \to \infty$, provided that $g(X, Y)$ and $\hat{f}(X, Y)$ are uncorrelated. Note that one such special case is when $\hat{f}(Y \mid X)$ follows a uniform distribution.

4.4. Asymptotic Convergence of the Expected Loss Function under the Normality Assumption

In this section, it is established that under the assumption of conditional normality, the expected loss function minimized by the proposed robust estimator converges asymptotically to that of standard local polynomial regression (LPR). As a consequence, both methods target the same underlying regression function $m(x)$ in expectation.

To proceed, consider the data-generating process and the associated assumptions. Let (X_i, Y_i), $i = 1, \ldots, N$, be i.i.d. observations drawn from a joint distribution with density $f(X, Y)$. Suppose that for each fixed x, the conditional density $f(Y \mid X = x)$ is given by:

$$f(Y \mid X = x) = \frac{1}{\sqrt{2\pi\sigma^2(x)}} \exp\left(-\frac{(Y - m(x))^2}{2\sigma^2(x)}\right), \tag{27}$$

where $m(x) = \mathbb{E}[Y \mid X = x]$ and $\sigma^2(x) = \mathbb{E}[(Y - m(x))^2 \mid X = x]$ are both continuous functions in a neighborhood of the point of interest x. This assumption of normality is often reasonable in many settings or can serve as a benchmark for understanding the behavior of the estimator.

Recall that the proposed robust estimator is defined through the minimization of:

$$\mathcal{L}_{rsk}(x) = \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j\right)^2 K_{H_1}(x - X_i)\hat{f}(Y_i \mid X_i), \tag{28}$$

where $\hat{f}(Y_i \mid X_i)$ is a nonparametric estimate of $f(Y_i \mid X_i)$ with bandwidth H_2, and K_{H_1} is a kernel function applied to the predictors with bandwidth H_1. For simplicity, if X is univariate, set $H_1 = h$. The analysis is then conducted subject to the usual nonparametric conditions as $N \to \infty$, with $h \to 0$ and $Nh \to \infty$.

Taking expectations of both sides:

$$\mathbb{E}[\mathcal{L}_{rsk}(x)] = \mathbb{E}\left[\sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x)(x - X_i)^j\right)^2 K_h(x - X_i)f(Y_i \mid X_i)\right], \tag{29}$$

where $\hat{f}(Y_i \mid X_i)$ is replaced with its limiting form $f(Y_i \mid X_i)$ as $N \to \infty$. This step is justified by standard results in nonparametric density estimation, which ensure that a consistent estimator $\hat{f}(Y_i \mid X_i) \xrightarrow{a.s.} f(Y_i \mid X_i)$ under asymptotic behavior.

Recall that the expected loss function is expressed as an integral over the joint density $f(X, Y)$:

$$\mathbb{E}[\mathcal{L}_{rsk}(x)] = N \iint \Big((Y - \beta(X; x))^2 K_h(x - X)[f(Y \mid X)]^2\Big) f(X)\, dY\, dX. \tag{30}$$

Under the normality assumption, we now focus on the integrand $(Y - \beta(X; x))^2 [f(Y \mid X)]^2$. Since $f(Y \mid X)$ is Gaussian, $[f(Y \mid X)]^2$ is also proportional to a Gaussian density, but with the same mean $m(X)$ and halved variance $\sigma^2(X)/2$. More precisely, for each fixed $X = x$,

$$[f(Y \mid X)]^2 \propto \frac{1}{\sqrt{\pi\sigma^2(x)}} \exp\left(-\frac{(Y - m(x))^2}{\sigma^2(x)}\right). \tag{31}$$

Integrating out Y, consider the expectation:

$$\int (Y - \beta(X; x))^2 [f(Y \mid X)]^2 dY. \tag{32}$$

Since this integral is now taken with respect to a Gaussian density centered at $m(X)$ but with half the original variance, it is obtained:

$$\int (Y - \beta(X; x))^2 [f(Y \mid X)]^2 dY = (m(X) - \beta(X; x))^2 + \frac{\sigma^2(X)}{2}. \tag{33}$$

Substituting this result back into the expectation:

$$\mathbb{E}[\mathcal{L}_{rsk}(x)] \propto N \int f(X) K_1\left(\frac{X - x}{h}\right)\left((m(X) - \beta(X; x))^2 + \frac{\sigma^2(X)}{2}\right) dX. \tag{34}$$

Because $\frac{\sigma^2(X)}{2}$ does not depend on $\beta(X; x)$, it does not influence the minimization. Thus, minimizing $\mathbb{E}[\mathcal{L}_{rsk}(x)]$ with respect to $\beta_j(x)$ is equivalent to minimizing:

$$\int f(X) K_1\left(\frac{X - x}{h}\right)(m(X) - \beta(X; x))^2 dX. \tag{35}$$

This matches precisely the objective that standard LPR minimizes in expectation. Hence, under the normality assumption and as $N \to \infty$, the proposed robust estimator and the standard LPR estimator identify the same target function $m(x)$.

In summary, when the conditional distribution is normal, the weighting mechanism introduced by $\hat{f}(Y_i \mid X_i)$ does not alter the asymptotic solution in expectation. While the proposed approach may achieve increased robustness to outliers and noise in finite samples, it retains the desirable asymptotic correctness of standard LPR. This result provides a theoretical anchor: under idealized (normal) conditions, the robust method and standard LPR coincide asymptotically in expectation, ensuring no asymptotic penalty is incurred for adopting the robust weighting scheme.

4.5. Asymptotic Bias under Non-Normal Conditional Distributions

While the proposed robust estimator aligns asymptotically with standard local polynomial regression (LPR) under the assumption of conditional normality, real-world data often deviate from this idealized condition. When the conditional distribution $f(Y \mid X)$ is not normal, particularly if it exhibits asymmetry, the asymptotic behavior of the estimator can be affected, potentially introducing bias.

To explore the implications of non-normal conditional distributions on the asymptotic properties of the proposed estimator, consider the expected loss function:

$$\mathbb{E}\left[\mathcal{L}_{\text{rsk}}(x)\right] \propto N \iint (Y - \beta(X;x))^2 K_1\left(\frac{X-x}{h}\right)[f(Y \mid X)]^2 f(X)\, dY\, dX. \tag{36}$$

When $f(Y \mid X)$ is asymmetric, the squared conditional density $[f(Y \mid X)]^2$ alters the weighting in the loss function in a way that can shift the effective mean and variance. Specifically, the expected value of Y under the squared density $[f(Y \mid X)]^2$ is generally not equal to the mean $m(X)$ of the original conditional distribution.

This shift implies that the minimization of the expected loss function may lead the estimator to converge to a value different from the true regression function $m(X)$, introducing an asymptotic bias. The magnitude and direction of this bias depend on the nature of the asymmetry in $f(Y \mid X)$.

To quantify the asymptotic bias in a general sense, consider that the mean of the squared conditional density $[f(Y \mid X)]^2$ is given by:

$$\mu'(X) = \frac{\int Y[f(Y \mid X)]^2\, dY}{\int [f(Y \mid X)]^2\, dY}. \tag{37}$$

Similarly, the variance under the squared density is:

$$\sigma'^2(X) = \frac{\int (Y - \mu'(X))^2 [f(Y \mid X)]^2\, dY}{\int [f(Y \mid X)]^2\, dY}. \tag{38}$$

The expected loss function then becomes:

$$\mathbb{E}\left[\mathcal{L}_{\text{rsk}}(x)\right] \propto N \int K_1\left(\frac{X-x}{h}\right) f(X)\left(\left(\mu'(X) - \beta(X;x)\right)^2 + \sigma'^2(X)\right) dX. \tag{39}$$

Since $\sigma'^2(X)$ does not depend on $\beta(X;x)$, minimizing $\mathbb{E}\left[\mathcal{L}_{\text{rsk}}(x)\right]$ with respect to $\beta(X;x)$ is equivalent to minimizing:

$$J(\beta(X;x)) = \int K_1\left(\frac{X-x}{h}\right) f(X)\left(\mu'(X) - \beta(X;x)\right)^2 dX. \tag{40}$$

Therefore, the estimator $\beta(X;x)$ converges to $\mu'(X)$ rather than $m(X)$. The asymptotic bias at point x can thus be quantified as:

$$\mathrm{Bias}(x) = \mu'(x) - m(x). \tag{41}$$

This bias arises because the mean under the squared conditional density $\mu'(X)$ differs from the mean $m(X)$ of the original conditional distribution $f(Y \mid X)$. The amount of bias depends on the degree and nature of asymmetry in $f(Y \mid X)$.

A detailed example illustrating this effect, including specific calculations of $\mu'(X)$ and $\sigma'^2(X)$ for a particular asymmetric distribution, is provided in **??**. This example demonstrates how the asymmetry of $f(Y \mid X)$ can lead to a shift in the estimator's asymptotic target due to the squared density weighting.

In practice, the presence of asymmetry in the conditional distribution may introduce some bias into the estimator. However, the robust weighting scheme of the proposed method can still provide advantages in terms of reducing the influence of outliers and improving estimation in the presence of heteroscedasticity or heavy-tailed errors. The trade-off between asymptotic bias and robustness to outliers should be considered in practical applications. Experiments on synthetic benchmarks in Section 5 demonstrate that, if the data is not overly dense, the proposed estimator often achieves comparable or better results in terms of RMSE than the standard LPR and typically substantially outperforms the iterative robust LOWESS estimator.

4.6. Trade-off Between Robustness and Bias via the K_2 Kernel and Bandwidth Selection

The proposed estimator utilizes the K_2 kernel to adjust data point weights based on both predictors and responses, controlling the trade-off between robustness and bias through the negative correlation between weights and residuals. The bandwidth H_2 of the K_2 kernel plays a crucial role in this mechanism.

In the loss function (4), each data point is weighted by:

$$w_i = K_{H_1}(x - X_i)\hat{f}(Y_i \mid X_i; H_2),$$

where K_{H_1} is a kernel based on the predictors, and $K_2 := \hat{f}(Y_i \mid X_i; H_2)$ is the estimated conditional density of the response at Y_i given X_i. The K_2 kernel assigns lower weights to less probable responses, effectively down-weighting outliers and inducing a negative correlation between the weights w_i and residuals $r_i = Y_i - \hat{m}(X_i)$.

The bandwidth H_2 controls the sensitivity of K_2 to variations in the response by adjusting the degree of negative correlation between weights and residuals. For very small H_2 values the density estimator $\hat{f}(Y_i \mid X_i; H_2)$ becomes sharply peaked at each Y_i, resembling delta functions. Since this occurs for all data points, the weights w_i become nearly uniform after normalization, diminishing the influence of residuals on the weights. Conversely, for very large H_2 the density estimator $\hat{f}(Y_i \mid X_i; H_2)$ becomes nearly constant across different Y_i, resulting in weights primarily determined by $K_{H_1}(x - X_i)$. In both extremes, the negative correlation between weights and residuals diminishes due to the weights becoming more uniform across data points.

An intermediate bandwidth H_2 achieves a balance between robustness and bias. It allows K_2 to assign weights that vary appropriately with the residuals, effectively down-weighting outliers while giving sufficient weight to informative points. The optimal H_2 depends on the data distribution and can be selected using methods like cross-validation or adaptive techniques based on local data characteristics.

By adjusting the bandwidth parameters, the estimator can realize a continuum of behaviors, ranging from the standard LPR approach to a more robust estimation regime. At one extreme, a larger bandwidth for K_2 effectively reduces the influence of response variability and approaches standard LPR. At the other extreme, a more restrictive bandwidth amplifies the role of local density and similarity, enhancing robustness but potentially introducing bias. This trade-off allows for nuanced tuning to suit specific applications and data characteristics. In settings with dense data, for example, reducing the bandwidth can dynamically control potential bias in high-density regions, yielding a locally tailored balance between robustness and accuracy. This adaptive capability opens the door for more sophisticated, context-dependent bandwidth selection strategies but is left for future work.

In summary, the K_2 kernel enables control over the robustness-bias trade-off by adjusting the negative correlation between weights and residuals through bandwidth selection. Proper choice of H_2 allows the estimator to mitigate the influence of outliers while maintaining low bias, effectively combining the strengths of robust and standard local polynomial regression.

4.7. Relationship to Kernel Methods and RKHS

In this subsection, the relationship of the proposed method to kernel methods and Reproducing Kernel Hilbert Spaces (RKHS) is explored. The use of positive definite kernels in defining the weights $K_{\mathcal{D}}$ allows the proposed estimator to be interpreted within the RKHS framework, providing deeper insights into its properties and connections to existing kernel-based methods.

Recall that in the proposed method, the weights in the loss function (4) are defined using a compound positive definite kernel $K_{\mathcal{D}}$ on the data domain $\mathcal{D}$:

$$\mathcal{L}_{\text{rsk}}(x, y; \mathcal{D}_N, H) := \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j(x, y)(x - X_i)^j \right)^2 K_{\mathcal{D}}\left((x, y), (X_i, Y_i); H\right). \tag{42}$$

As per equation (6), the kernel $K_{\mathcal{D}}$ is defined as a product of two positive definite kernels:

$$K_{\mathcal{D}}\Big((x, y), (x', y'); H_1, H_2\Big) = K_1(x, x'; H_1) \cdot K_2\Big((x, y), (x', y'); H_2\Big), \tag{43}$$

where K_1 is a kernel function depending only on the predictors x and x', typically chosen as the traditional distance-based kernel used in local polynomial regression, and K_2 is a kernel function that incorporates both predictors and responses.

Since $K_{\mathcal{D}}$ is a product of positive definite kernels, it is itself a positive definite kernel. Therefore, there exists a feature mapping $\phi : \mathcal{D} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, such that:

$$K_{\mathcal{D}}\Big((x, y), (x', y')\Big) = \langle \phi(x, y), \phi(x', y') \rangle_{\mathcal{H}}. \tag{44}$$

Thus, the weights $K_{\mathcal{D}}((x, y), (X_i, Y_i))$ can be interpreted as inner products in the feature space $\mathcal{H}$. Consequently, the loss function (42) can be viewed as a weighted least-squares problem where the weights are determined by the similarity between the feature representations of the data points and the point of interest.

Furthermore, consider the role of the Kernel Density Estimator (KDE) in the proposed method. The KDE at a point (x, y) is given by:

$$\hat{f}(x, y) = \frac{1}{N} \sum_{i=1}^{N} K_2\left((x, y), (X_i, Y_i); H_2\right). \quad (45)$$

Since K_2 is a positive definite kernel, there exists a feature mapping $\psi : \mathcal{D} \to \mathcal{G}$, where $\mathcal{G}$ is another Hilbert space, such that:

$$K_2\left((x, y), (x', y')\right) = \langle \psi(x, y), \psi(x', y') \rangle_{\mathcal{G}}. \quad (46)$$

Therefore, the KDE at (x, y) can be expressed in terms of inner products in the feature space $\mathcal{G}$:

$$\hat{f}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \langle \psi(x, y), \psi(X_i, Y_i) \rangle_{\mathcal{G}}. \quad (47)$$

This expression shows that the KDE at (x, y) is proportional to the inner product between the feature mapping $\psi(x, y)$ and the mean of the feature mappings of the data:

$$\hat{v}_{\psi} = \frac{1}{N} \sum_{i=1}^{N} \psi(X_i, Y_i), \quad (48)$$

so that:

$$\hat{f}(x, y) = \langle \psi(x, y), \hat{v}_{\psi} \rangle_{\mathcal{G}}. \quad (49)$$

This interpretation shows that the KDE measures how closely the feature representation $\psi(x, y)$ of a point (x, y) aligns with the average feature representation $\hat{v}_{\psi}$ of the data in the space induced by K_2. In the proposed method, this alignment influences the weights in the regression, as the density estimates $\hat{f}(x, y)$ or $\hat{f}(Y_i \mid X_i)$ derived from K_2 directly affect the overall weights $K_{\mathcal{D}}((x, y), (X_i, Y_i))$. This interplay underpins the robustness and adaptability of the proposed method.

By leveraging positive definite kernels for defining $K_{\mathcal{D}}$, the method inherently operates within the RKHS framework, where weights represent similarities in feature space. This perspective highlights the connection between the kernel-based weighting and the feature mappings, offering insights into the estimator's flexibility and robustness.

5. Experiments and Implementation Notes

This section presents an evaluation of the proposed method (RSKLPR), implemented in Python and published as an open source package https://github.com/yaniv-shulman/rsklpr. The experiments focus on comparing the performance of RSKLPR against existing local regression techniques under synthetic settings with different noise characteristics.

Implementation Details

The implementation normalizes distances in each neighborhood to the range $[0, 1]$, consistent with the approach in [3]. For the kernel $K_1(x, x')$, a Laplacian kernel $e^{-\|x-x'\|}$ was selected, demonstrating more consistent and efficient performance compared to alternatives like the tricube kernel. For density estimation in K_2, a factorized multidimensional Kernel Density Estimator (KDE) with scaled Gaussian kernels was used. Bandwidth selection for density estimation was explored using five methods: Scott's rule [12], the normal reference rule, global least-squares cross-validation, local least-squares cross-validation, and local maximum-likelihood cross-validation. Additionally, the bandwidth for the predictor kernel K_1 was empirically adjusted as a function of the window size in certain experiments. Scaling constants within neighborhoods, such as those in $\hat{f}(y \mid x)$ and $\hat{f}(x, y)$, were excluded for computational efficiency, as they do not impact the local regression results. The implementation supports local constant and local linear estimators however the experiments were done only with the local linear estimator i.e. $p = 1$ as it is well known to be superior.

Experimental Design

Synthetic datasets were generated with both additive Gaussian noise and asymmetric data distributions to simulate various regression scenarios. The following characteristics were varied: noise types, including homoscedastic and heteroscedastic Gaussian noise as well as asymmetric noise distributions (Exponential, Log-normal, Gamma, and Weibull); data density, encompassing both sparse and dense data regimes; and regression complexity, modeling non-linear curves and surfaces. Performance was evaluated using Root Mean Square Error (RMSE) and sensitivity to neighborhood size. For asymmetric noise settings, RMSE trends were analyzed as a function of data density.

Results and Observations

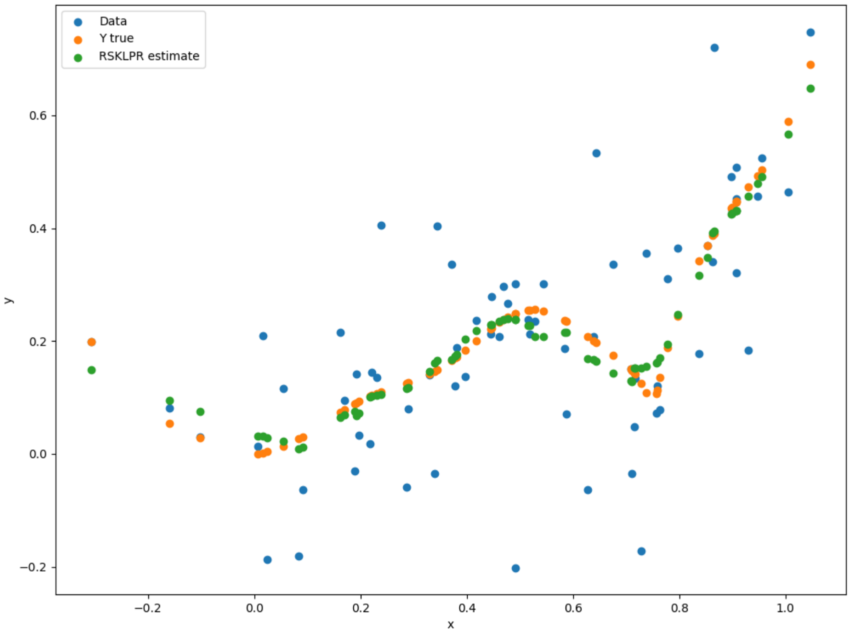
Under Gaussian noise settings, the proposed method performed competitively across a range of synthetic settings. Unlike iterative robust variants, RSKLPR achieved these results with a single iteration. A regression example with heteroscedastic Gaussian noise is shown in Figure 3. The proposed method aligns with the true regression function while effectively mitigating the influence of noise and outliers.

Under asymmetric data distributions, RSKLPR exhibited robust performance in low density settings, often matching or outperforming standard LPR and the iterative robust variant. In high-density settings, the proposed method diverged, thus confirming the theoretical results, however, it consistently outperformed the iterative robust LPR. Figure 4 presents RMSE trends for asymmetric noise distributions for the three methods.

The robustness-bias trade-off in RSKLPR is controlled by the bandwidth H_2 of the kernel K_2. Small bandwidths enhance robustness by down-weighting outliers but may introduce bias, while larger bandwidths reduce bias but diminish robustness. An intermediate bandwidth provides an optimal balance, as demonstrated in experiments.

The method was also significantly less sensitive to the neighborhood size making it an an attractive option for applications where robust regression is critical. Complete experimental results, including multivariate settings and bootstrap-based confidence intervals, are available at https://nbviewer.org/github/yaniv-shulman/rsklpr/tree/main/src/experiments as interactive Jupyter notebooks [1].

Figure 3: Performance of RSKLPR on 1D synthetic data with heteroscedastic Gaussian noise. The proposed method effectively aligns with the true regression function while mitigating the influence of outliers and noise.

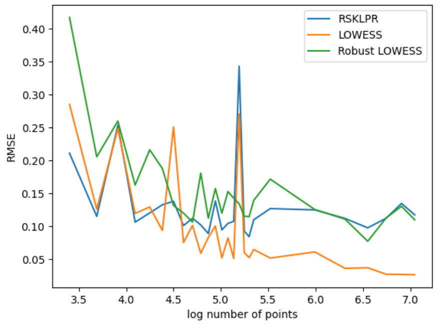


6. Future Work and Research Directions

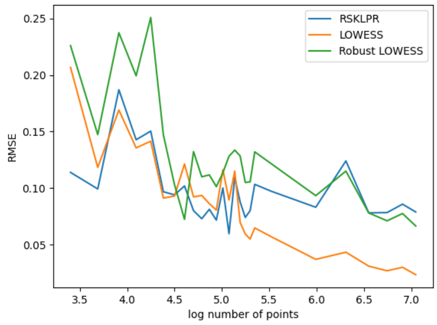
This work introduces a new robust variant of Local Polynomial Regression (LPR), opening several avenues for further exploration and refinement. Since the proposed method generalizes the traditional LPR, there are opportunities to replace certain standard components in equation (4) with more robust alternatives. These could include approaches such as robust methods for bandwidth selection or substituting the conventional quadratic residual function with alternatives better suited for handling outliers.

An important research direction is to explore adaptive bandwidth selection strategies that respond dynamically to local data density. In regions where data are sparse, the bandwidth in K_2 could be fine-tuned to maintain robust down-weighting of potential outliers, ensuring sufficient flexibility while avoiding an overly coarse estimate. Conversely, in denser regions, broader bandwidths may be adopted, causing the estimator to behave more like standard LPR and reduce any bias introduced by the robust weighting. Incorporating such adaptive bandwidths could further enhance the method's overall performance and flexibility, particularly in heterogeneous data scenarios.

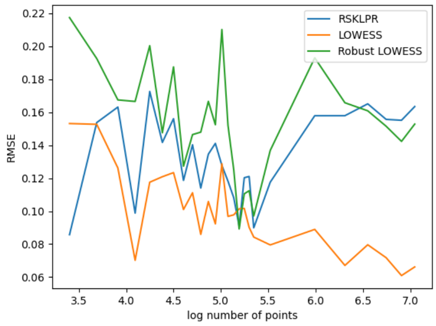
Additionally, further development of this framework may involve exploring different kernel functions K_D and assessing how robust density estimators influence overall performance. Ex-



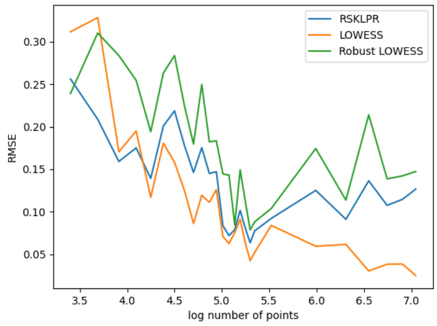
(a) Exponential.



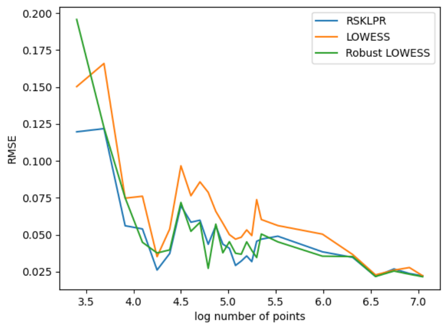
(b) Gamma.



(c) Log-normal.



(d) Weibull.



(e) Gaussian.

Figure 4: These subplots compare RMSE as a function of data density for the proposed method (RSKLPR), standard LOWESS, and Robust LOWESS (5 iterations) across various noise distributions: (a) Exponential, (b) Gamma, (c) Log-normal, (d) Weibull, and (e) Gaussian. The results demonstrate the effectiveness of RSKLPR in low-density data and align well with theoretical expectations for denser data.

tending the method within the RKHS framework presents another valuable direction. This could allow for the introduction of a regularization term in the loss function, enhancing control over estimator smoothness and mitigating the risk of overfitting. Through these future directions, the robustness and adaptability of the proposed method could be substantially advanced.

References

[1] Project jupyter is a non-profit, open-source project, born out of the ipython project in 2014 as it evolved to support interactive data science and scientific computing across all programming languages. `https://jupyter.org/`.

[2] M. Avery. Literature review for local polynomial regression. 2010.

[3] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[4] W. S. Cleveland and S. J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

[5] J. Fan. Local linear regression smoothers and their minimax efficiencies. *The Annals of Statistics*, 21, 03 1993.

[6] E. García-Portugués. *Notes for Nonparametric Statistics*. 2023. Version 6.9.0. ISBN 978-84-09-29537-1.

[7] T. Gasser and H.-G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scandinavian Journal of Statistics*, 11:171–185, 1984.

[8] R. A. Maronna, D. Martin, V. J. Yohai, and Hardcover. Robust statistics: Theory and methods. 2006.

[9] H.-G. Muller. Weighted local regression and kernel methods for nonparametric curve fitting. *Journal of the American Statistical Association*, 82(397):231–238, 1987.

[10] E. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9:141–142, 1964.

[11] M. Salibian-Barrera. Robust nonparametric regression: Review and practical considerations. *Econometrics and Statistics*, 2023.

[12] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 2015.

[13] S. Seabold and J. Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[14] sigvaldm. Localreg is a collection of kernel-based statistical methods. `https://github.com/sigvaldm/localreg`.

[15] V. G. Spokoiny. Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice. *The Annals of Statistics*, 26(4):1356 – 1378, 1998.

[16] C. J. Stone. Consistent nonparametric regression. *Annals of Statistics*, 5:595–620, 1977.

[17] P. Čížek and S. Sadıkoğlu. Robust nonparametric regression: A review. *WIREs Comput. Stat.*, 12(3), apr 2020.

[18] G. S. Watson. Smooth regression analysis. 1964.