

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

## 2 Robust Local Polynomial Regression with Similarity Kernels

3 Yaniv Shulman

4 *yaniv@aleph-zero.info*

---

### 5 Abstract

Local polynomial regression 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 further away. Local polynomial regression is however susceptible to outliers and high leverage points which may cause an adverse impact on the estimation accuracy. The main contribution of this paper is to revisit the kernel that is used to produce local 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, a positive definite kernel is proposed that assigns robust weights to mitigate the adverse effect of outliers in the local neighborhood by incorporating the conditional density of the response at the local locations. The method is implemented in the Python programming language and is made publicly available at <https://github.com/yaniv-shulman/rsklpr>. Experimental results on synthetic benchmarks across a range of settings demonstrate that the proposed method achieves competitive results.

---

## 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 further 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 parametrized 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 [9, 17] and the estimator proposed by Gasser and Müller [6] 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 [15, 3, 8, 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 [16]. 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 [14] 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 [7]. For a recent review of robust LPR and other nonparametric methods see [16, 10]

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 re-

sponse are used to calculate weights. Within this framework, a positive definite kernel is proposed that assigns robust weights to mitigate the adverse effect of outliers in the local neighborhood by incorporating the conditional 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 LOESS/LOWESS using only a single training iteration. Furthermore in the tested setting the method is less sensitive to the choice of the bandwidth parameter. This last quality makes it an attractive choice in practice since it is more likely for the analyst to select an "appropriate" value when there is no ground truth available.

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 a specific robust positive definite kernel are proposed. In section 4, implementation notes and experimental results are provided. Finally, in section 5, 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  $(X, Y) \sim f_{XY}$  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  $Y_i = m(X_i) + \epsilon_i$ ,  $i \in 1, \dots, T$  where  $m(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$  is an unknown function and  $\epsilon_i$  are independently distributed error terms having zero mean representing random variability not included in  $X_i$  such that  $\mathbb{E}[Y|X = x] = m(x)$ . 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  $x$  near a point  $X_i$  yields:

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

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  $\mathcal{L}_{lpr}(\cdot; \mathcal{D}_N, H)$ :

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

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

Where  $\gamma \in \mathbb{R}^{p+1}$ ;  $K_H(\cdot)$  is a scaled kernel,  $H$  is the set of bandwidth parameters and  $\mathcal{D}_N$  is the subset of  $N$  nearest neighbors in the training set. 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].

### 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  $K_{\mathcal{D}} : \mathcal{D}^2 \rightarrow \mathbb{R}_+$  that enables weighting each point and incorporating information on the data in the local neighborhood in relation to the local regression target.

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

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  $s(\cdot)$ , if  $s(y) = s(y')$  the weight should decrease as the distance between the predictors grows,  $s(y) = s(y') \implies K_{\mathcal{D}}((x, y), (x + u, y')) \geq K_{\mathcal{D}}((x, y), (x + v, y')) \quad \forall \|u\| \leq \|v\|$

In this work one such useful positive definite kernel is proposed that models two useful concepts. The first is that it tends to diminish as the distance between the explanatory variables increases to model stronger relationship between closer points, this is similar to the usual kernels used in (3). In addition the weights produced by the kernel also model the "importance" of  $y$  when  $x$  is fixed. 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:

$$K_{\mathcal{D}}((x, y), (x', y'); H_1, H_2) = K_1(x, x'; H_1) K_2((x, y), (x', y'); H_2) \quad (5)$$

$$= K_1(x, x'; H_1) \hat{p}(y | x; H_2) \hat{p}(y' | x'; H_2) \quad (6)$$

Where  $K_1 : \mathbb{R}^d \rightarrow \mathbb{R}_+$  and  $K_2 : \mathcal{D}^2 \rightarrow \mathbb{R}_+$  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 (3). 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

116 methods that could be employed to estimate the extent to which a data point is a local outlier  
 117 however in this work it is estimated inversely proportional to the estimated localized conditional  
 118 marginal distribution of the response variable at the location. The nonparametric conditional  
 119 density estimation is performed using the Parzen–Rosenblatt windows (kernel density estimator).

$$\hat{p}(y | x; H_2) = \hat{p}(x, y; H_2) / \hat{p}(x; H_2) \quad (7)$$

$$= \hat{p}(v; \mathbf{H}_v) / \hat{p}(x; \mathbf{H}_x) \quad (8)$$

$$= \frac{|\mathbf{H}_x|^{1/2} \sum_{i=1}^N K_v(\mathbf{H}_v^{-1/2}(v - V_i))}{|\mathbf{H}_v|^{1/2} \sum_{i=1}^N K_x(\mathbf{H}_x^{-1/2}(x - X_i))} \quad (9)$$

120 Where  $v = [x, y] \in \mathbb{R}^{d+1}$  is the concatenated vector of the predictors and the response; and  
 121  $\mathbf{H}_v, \mathbf{H}_x$  are bandwidth matrices. The hyperparameters of this model are similar in essence to the  
 122 standard local polynomial regression and include the span of included points, the kernels and  
 123 their associated bandwidths. Note that this estimator can be replaced with a robust alternative  
 124 and better results are anticipated by doing so however exploring this option is left for future work.

125

The optimization is invariant to the scale of the objective, therefore:

$$\hat{\beta}(x, y; \mathcal{D}_N, H) := \min_{\beta(x)} \sum_{i=1}^N \left( Y_i - \sum_{j=0}^p \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i) \hat{p}(y | x; H_2) \hat{p}(Y_i | X_i; H_2) \quad (10)$$

$$= \min_{\beta(x)} \sum_{i=1}^N \left( Y_i - \sum_{j=0}^p \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i) \hat{p}(Y_i | X_i; H_2) \quad (11)$$

Thus for a point  $(x, y)$ , in the limit as  $T, N \rightarrow \infty, H_1, H_2 \rightarrow 0$  and  $NH_1, NH_2 \rightarrow \infty$ :

$$\lim_{N \rightarrow \infty, H_1, H_2 \rightarrow 0} \sum_{i=1}^N \left( Y_i - \sum_{j=0}^p \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i) \hat{p}(Y_i | X_i; H_2) \quad (12)$$

$$= \mathbb{E}_{\hat{p}(Y | X=x)} \left[ \sum_{i=1}^N \left( Y_i - \sum_{j=0}^p \beta_j(x)(x - X_i)^j \right)^2 K_{H_1}(x - X_i) \right] \quad (13)$$

126 That is, with the choice of  $K_D$  as per equation (6) the asymptotic objective in equation (4) is  
 127 equivalent to the expectation of the LPR objective (2) under the estimated conditional distribution  
 128  $\hat{p}(Y | X = x)$ .

$$\lim_{N \rightarrow \infty, H_1, H_2 \rightarrow 0} \hat{\beta}(x, y; \mathcal{D}_N, H) := \min_{\gamma(x)} \mathbb{E}_{\hat{p}(Y | X=x)} [\mathcal{L}_{lpr}(x; \mathcal{D}_N, H)] \quad (14)$$

#### 129 4. Experiments and implementation notes

130 The proposed method was implemented in Python for the 1D predictor case. The distances  
 131 between pairs are normalized to the range  $[0, 1]$  in each neighborhood as in [3]. The simple Lapla-  
 132 cian kernel  $e^{-\|x-x'\|}$  is used for  $K_1(x, x'; h_1)$  since it gave better empirical results in tests than the

tricube kernel recommended in [3]. A factorized multidimensional KDE using scaled Gaussian kernels is used for estimating the joint density. Three methods for estimating the bandwidth for the response kernel were implemented: Scott’s rule [11], global LSCV and local LSCV. Whereas the bandwidth for the predictor’s kernel uses a simple function of the window size estimated empirically. Certain computations are omitted for efficiency since the local regression in equation (4) is invariant to the scale of the weights. This includes all scaling constants fixed in a given neighborhood concerning a specific local regression target such as the denominator in equation (9) and the computation of  $\hat{p}(y|x)$  in equation (6). Experiments were performed on a number of synthetic benchmarks to evaluate the effectiveness of the method’s linear and quadratic variants in comparison to other methods of the local polynomial regression family including LOWESS and iterative robust LOWESS [12], local linear and local constant kernel regression [12], local quadratic regression [13] and radial basis function network [13].

The experiments comprise a number settings including various non-linear synthetic curves with added noise. These have variants of dense and sparse data, homoscedastic and heteroscedastic noise characteristics and various neighborhood sizes. The results indicate there is no best universal method and different methods work better in a given setting. However the proposed method offers competitive performance in particular in heteroscedastic settings and generally improves on it’s direct counterparts LOESS/LOWESS and quadratic LPR using a single iteration. In addition it appears far less sensitive to the choice of neighborhood size and has substantial reduced variance in this respect. This last quality makes it an attractive choice since it is more likely for the analyst to select an ”appropriate” hyperparameter value when there is no ground truth available.

The experimental results are available at <https://nbviewer.org/github/yaniv-shulman/rsklpr/tree/main/src/experiments/> as interactive Jupyter notebooks [1].

## 5. Future work and research directions

This work proposes a new robust variant of LPR and as such there are many aspects that need to be explored and are left for future work. As the proposed method generalize the standard LPR it does not preclude replacing some of the standard LPR components in equation (4) with other and robust alternatives including robust methods for bandwidth selection and robust alternatives to the OLS loss. Another avenue for developing this framework is investigating additional kernels  $K_D$  and exploring the impact of robust density estimators on performance.

## 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] T. Gasser and H.-G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scandinavian Journal of Statistics*, 11:171–185, 1984.
- [7] R. A. Maronna, D. Martin, V. J. Yohai, and Hardcover. Robust statistics: Theory and methods. 2006.

- 177 [8] H.-G. Muller. Weighted local regression and kernel methods for nonparametric curve fitting. *Journal of the*  
178 *American Statistical Association*, 82(397):231–238, 1987.
- 179 [9] E. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9:141–142, 1964.
- 180 [10] M. Salibian-Barrera. Robust nonparametric regression: Review and practical considerations. *Econometrics and*  
181 *Statistics*, 2023.
- 182 [11] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and  
183 Statistics. Wiley, 2015.
- 184 [12] S. Seabold and J. Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in*  
185 *Science Conference*, 2010.
- 186 [13] sigvaldm. Localreg is a collection of kernel-based statistical methods. [https://github.com/sigvaldm/](https://github.com/sigvaldm/localreg)  
187 [localreg](https://github.com/sigvaldm/localreg).
- 188 [14] V. G. Spokoiny. Estimation of a function with discontinuities via local polynomial fit with an adaptive window  
189 choice. *The Annals of Statistics*, 26(4):1356 – 1378, 1998.
- 190 [15] C. J. Stone. Consistent nonparametric regression. *Annals of Statistics*, 5:595–620, 1977.
- 191 [16] P. Čížek and S. Sadıkoğlu. Robust nonparametric regression: A review. *WIREs Comput. Stat.*, 12(3), apr 2020.
- 192 [17] G. S. Watson. Smooth regression analysis. 1964.