Multiscale Robust Regression, Multiscale Influence Analysis and Application to Edge Detection

Gilad Lerman

Department of Mathematics, University of Minnesota, 127 Vincent Hall, 206 Church St. S.E., Minneapolis, MN 55455 USA

Joseph McQuown

Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794 USA

Bud Mishra

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012 USA
NYU School of Medicine, 550 First Avenue, New York, NY 10016 USA

Summary. The Multiscale Strip Construction, or in short MSC (Lerman et al., 2007), was initially proposed to identify enriched spots in ChIP-on-chip arrays. Here we extend this method to the setting of robust high-dimensional curve estimation. We establish its robustness as well as accurateness by bounding both the empirical influence function and the empirical bias of approximation at a sufficiently large set of points. We also demonstrate in practice the robustness and accurateness of MSC by comparing it with other successful methods of robust regression on artificial data sets with significant outlier components. A new application of the high-dimensional MSC to edge detection is introduced and tested.

Keywords: Robust regression, Influence function, Multiscale analysis

1. Introduction

The Multiscale Strip Construction, or MSC (Lerman et al., 2007), is a recent heuristic strategy for robust regression and local variance estimation. It was originally designed in order to identify enriched spots in ChIP-on-chip arrays. Ideally, those spots correspond to promoters that bind to a transcription factor of interest. Standard statistical methods have not performed well with such bioinformatic data because of the nontrivial structure of their noise and outliers, which are hard to model (Buck and Lieb, 2004).

In this paper we extend the MSC to the higher-dimensional setting of robust curve estimation in any Euclidean space. We provide here full mathematical and statistical details of the method (in its extended form) and establish its properties so that it is not any more a heuristic procedure.

The theoretical goal of this paper is to justify the inherent robustness of MSC through rigorous analysis. The main result here (Theorem 3.1) shows that there exists a sufficiently large set of points whose empirical influence function (Huber, 1981; Cook and Weisberg, 1982) and empirical bias of approximation are both well-controlled. The finer the scales of the local regions obtained by MSC, the better the control is.

In addition to such theoretical foundations, this paper also numerically demonstrates the success of MSC in various artificial instances, while comparing it with different versions of

the LOESS algorithm (Cleveland, 1979; Cleveland and Devlin, 1988; Cleveland and Loader, 1996). Moreover, it suggests a new application of the MSC algorithm to edge detection of images, in particular, ones corrupted with noise. The idea is to search for outliers (deviating from the main curve) in the high-dimensional set of pixel neighborhoods of a given gray-level image.

The strategies of the MSC are related to those of Arias-Castro et al. (2006) and some of its references. Indeed, the idea of the MSC is to zoom in on the given data in a top-down multiscale procedure and select local regions of various scales. A global nonlinear model is obtained by combining the local information at those regions.

The paper is organized as follows. In Section 2 we formulate the MSC method in a general high-dimensional setting, while maintaining the intrinsic dimension to be one. Theoretical properties of the method are studied in Section 3, in particular, we analyze its robustness and accurateness, bound its speed and quantify the smooth of its output. In Section 4 we test the accuracy, robustness and speed of the MSC method on artificial data sets, while comparing it with other commonly used methods. We also test there the MSC with the new application of edge detection using high-dimensional pixel neighborhoods. We conclude with a discussion in Section 5.

2. Description of the High-dimensional MSC Method

The input of the MSC algorithm is data

$$E = \{x_i, \mathbf{y}_i\}_{i=1}^N \subseteq \mathbb{R}^{D+1}\,,$$

where $D \geq 1$ ($x_i \in \mathbb{R}$ and $\mathbf{y}_i \in \mathbb{R}^D$), and the following parameters: l_0, c_0, n_0, λ_0, $n_{\rm sh}$, q_0 and α_0. The output includes the following estimators: 1) A regression function $\widetilde{\mathbf{C}}$ which explains the last D coordinates by the first coordinate and is robust to outliers. 2) A conditional standard deviation $\widetilde{S}$, which is robust to outliers. More precisely, the covariance matrix of $\mathbf{y}$ given x is modeled as a scalar matrix with elements $\widetilde{S}^2(x)$. 3) Ranking for data points as outliers. 4) Set of detected outliers.

The initial step of the algorithm forms a least squares regression line (of $\mathbf{y}$ given x) and then transforms the data by subtracting from its second to last coordinates the corresponding coordinates of this line. Consequently, the x-axis is the new regression line. The next steps are described in the following subsections, while Figure 3 demonstrates the algorithm for a very particular data.

2.1. Formation of multiscale grids, regions and lines

For any given half-closed, half-open interval Q, its dyadic children, Q_L and Q_R (left and right respectively), are the two half-closed, half-open intervals of equal lengths whose union is Q.

The algorithm fixes an interval $Q_0 := [a_0, b_0)$ of almost minimal length containing the projection of E onto the x-axis. It then repeatedly partitions Q_0 to dyadic children, but not more than l_0 times. We denote the set of all such intervals by $\mathcal{D}(Q_0)$. If $Q \in \mathcal{D}(Q_0) \setminus \{Q_0\}$, then we denote by P_Q the dyadic parent of Q according to the grid $\mathcal{D}(Q_0)$ and we also define $P_{Q_0} := Q_0$.

For each $Q \in \mathcal{D}(Q_0)$ the algorithm associates an infinite strip

$$\mathrm{Str}(Q) = Q \times \mathbb{R}^D\,,$$

and recursively constructs the regions $\mathrm{Cyl}(Q)$, $\mathrm{Out}(Q)$ and line L_Q from top to bottom levels as described below (see also demonstration in Figure 1).

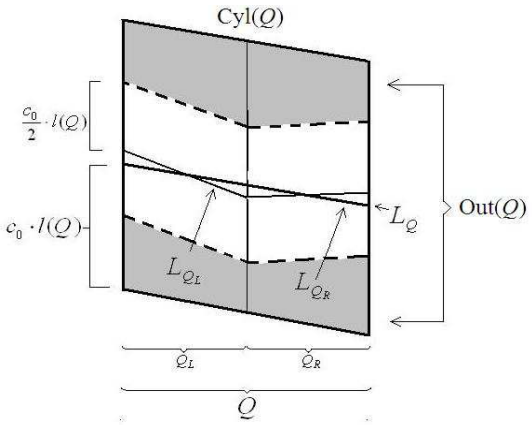


Fig. 1. Regions and lines when $D = 1$.

If $Q = Q_0$, then L_{Q_0} is the transformed x-axis and $\mathrm{Cyl}(Q_0) = \mathrm{Str}(Q_0)$. Moreover, the lines $L_{Q_{0L}}$ and $L_{Q_{0R}}$ are the regression lines for the strips $\mathrm{Str}(Q_{0L})$ and $\mathrm{Str}(Q_{0R})$ respectively (one may first remove from those strips the top and bottom $\varepsilon/2$-quantiles, where $\varepsilon \ll \alpha_0$). If $Q \subsetneq Q_0$, then the regression line L_Q has been defined in the previous level and its corresponding regression function is denoted by $\mathbf{y} = \mathbf{L}_Q(x)$. The region $\mathrm{Cyl}(Q)$ has the form

$$\mathrm{Cyl}(Q) = \{(x, \mathbf{y}) \in \mathrm{Cyl}(P_Q) \cap \mathrm{Str}(Q) : \|\mathbf{y} - \mathbf{L}_Q(x)\|_2 \le c_0 \cdot \ell(Q)\} \ .$$

The algorithm then forms the lines L_{Q_L} and L_{Q_R} as the least squares regression lines for $\mathrm{Str}(Q_L) \cap \mathrm{Cyl}(Q)$ and $\mathrm{Str}(Q_R) \cap \mathrm{Cyl}(Q)$ respectively (the corresponding regression functions are $\mathbf{y} = \mathbf{L}_{Q_L}(x)$ and $\mathbf{y} = \mathbf{L}_{Q_R}(x)$). Finally, the algorithm sets the following region of "putative local outliers":

$$\mathrm{Out}(Q) = \left\{(x, \mathbf{y}) \in \mathrm{Cyl}(Q) \cap \mathrm{Str}(Q_L) : \ \|\mathbf{y} - \mathbf{L}_{Q_L}(x)\|_2 > \frac{c_0 \cdot \ell(Q)}{2}\right\} \bigcup$$
$$\left\{(x, \mathbf{y}) \in \mathrm{Cyl}(Q) \cap \mathrm{Str}(Q_R) : \ \|\mathbf{y} - \mathbf{L}_{Q_R}(x)\|_2 > \frac{c_0 \cdot \ell(Q)}{2}\right\}.$$

2.2. Computation of Local quantities

The algorithm computes at each visited interval Q the following quantities: f_Q, F_Q, $\sigma_{\mathbf{Y}|X}(Q)$ and $\sigma_X(Q)$. Their definitions below apply the notation $|A|$ (for general $A \subseteq \mathbb{R}^D$) to designate the number of points in $A \cap E$.

The fraction f_Q is the ratio of "putative local outliers" to the total number of points projected on Q. That is,

$$f_Q = \frac{|\mathrm{Out}(Q)|}{|\mathrm{Str}(Q)|}.$$

The quantity F_Q adds up such fractions of all parent-intervals (including current interval), that is,

$$F_Q = \sum_{\substack{Q' \in \mathcal{D}(Q_0) \\ Q' \supseteq Q}} f_{Q'}.$$

We remark that distant analogs of this quantity are the square functions of harmonic analysis (Stein, 1993) or the J function (Bishop and Jones, 1994; Lerman, 2003). The algorithm computes F_Q with a top-down procedure: First, it initializes $F_Q \equiv 0$ for all $Q \in \mathcal{D}(Q_0)$. Then, it applies the reduction formula (from coarse levels to fine levels):

$$F_Q = F_{P_Q} + f_Q.$$

The quantities $\sigma_{\mathbf{Y}|X}(Q)$ and $\sigma_X(Q)$ estimate the following local versions of standard deviations in the regions $R(Q) = \mathrm{Str}(Q) \cap \mathrm{Cyl}(P_Q)$:

$$\sigma_{\mathbf{Y}|X}(Q) = \left(\frac{1}{|R(Q)|} \sum_{(x,\mathbf{y}) \in R(Q) \cap E} \|\mathbf{y} - \mathbf{L}_Q(x)\|_2^2 \right)^{\frac{1}{2}}$$

and

$$\sigma_X(Q) = \left(\frac{1}{|R(Q)|} \sum_{(x,\mathbf{y}) \in R(Q) \cap E} \left(x - \frac{\sum_{(x,\mathbf{y}) \in R(Q) \cap E} x}{|R(Q)|} \right)^2 \right)^{\frac{1}{2}}.$$

2.3. Stopping-time Criteria

While proceeding from top to bottom levels, the algorithm stops at an interval $Q' = [a_{Q'}, b_{Q'}) \in \mathcal{D}(Q_0)$ (together with all of its descendants in $\mathcal{D}(Q_0)$) if one of the following conditions is satisfied:

1. $F_{Q'} > \alpha_0.$ (1)
2. $|\mathrm{Cyl}(Q')| < n_0.$ (2)
3. $\sigma_X^2(Q) < \lambda_0 \cdot \ell(Q)^2.$ (3)
4. $\ell = \ell_0.$ (4)

We will use the following sets of stopping-time intervals and their parents throughout the rest of the paper:

$$\begin{aligned}
\mathcal{Q} &= \{Q \in \mathcal{D}(Q_0) : \ Q \text{ is a stopping-time interval}\}, \\
\mathcal{B} &= \{Q \in \mathcal{Q} : \ \ F_Q < \alpha_0 \text{ or } \sigma_X^2(Q) < \lambda_0 \cdot \ell(Q)^2\}, \\
\mathcal{P} &= \{P \in \mathcal{D}(Q_0) : \ \exists\, Q \in \mathcal{Q} \text{ so that } Q \subseteq P\}.
\end{aligned}$$

2.4. Output of MSC

2.4.1. Main Functions Computed by MSC

The output of the algorithm includes the functions $\widetilde{\mathbf{C}}$ and $\widetilde{S}$ (or alternatively $\widehat{S}$). These functions are defined below for any $x \in Q_0$, but in practice only computed for the projection

of the data onto Q_0. For simplicity we describe them as piecewise linear or piecewise constant functions and present their smoothed versions at the end of this subsection. We will frequently use here the notation χ_Q for the indicator function of the interval Q, i.e. $\chi_Q(x) = 1$ if $x \in Q$ and $\chi_Q(x) = 0$ otherwise.

The algorithm computes the functions $\widetilde{\mathbf{C}}$ and $\widetilde{S}$ by the formulas:

$$\widetilde{\mathbf{C}}(x) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} \mathbf{L}_Q(x) \cdot \chi_Q(x) + \sum_{Q \in \mathcal{B}} \mathbf{L}_{P_Q}(x) \cdot \chi_Q(x) \tag{5}$$

and

$$\widetilde{S}(x) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} \sigma_{\mathbf{Y}|X}(Q) \cdot \chi_Q(x) + \sum_{Q \in \mathcal{B}} \sigma_{\mathbf{Y}|X}(P_Q) \cdot \chi_Q(x)\,.$$

The optional estimator $\widehat{S}$ extends the estimates of $\widetilde{S}$ outside the regions $\{\mathrm{Cyl}(Q)\}_{Q \in \mathcal{Q}}$ while assuming that the data can be locally approximated by a restriction of a normal distribution. The algorithm computes $\widehat{S}$ by the formula

$$\widehat{S}(x) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} \widehat{\sigma}_Q \cdot \chi_Q(x) + \sum_{Q \in \mathcal{B}} \widehat{\sigma}_{P_Q} \cdot \chi_Q(x)\,,$$

where for each $Q \in \mathcal{Q}$, $\widehat{\sigma}_Q$ is the solution of the following equation with $b_Q = c_0 \cdot \ell(Q)$:

$$\sigma_{\mathbf{Y}|X}(Q)^2 = -\frac{2}{\sqrt{2\pi}} \cdot \widehat{\sigma}_Q \cdot b_Q \cdot e^{-\frac{b_Q^2}{2 \cdot \widehat{\sigma}_Q^2}} + \frac{\widehat{\sigma}_Q^2}{2} \cdot \left(\mathrm{erf}\left(\frac{b_Q}{\sqrt{2} \cdot \widehat{\sigma}_Q} \right) - \mathrm{erf}\left(\frac{-b_Q}{\sqrt{2} \cdot \widehat{\sigma}_Q} \right) \right). \tag{6}$$

This formula is explained in Appendix A. In practice $\widetilde{S}$ and $\widehat{S}$ are very similar and thus $\widetilde{S}$ can be used if it is important to avoid parametric assumptions.

The functions $\widetilde{\mathbf{C}}$, $\widetilde{S}$ and $\widehat{S}$ are smoothed as follows: First, the algorithm generates n_{sh} instances of piecewise constant functions as above but according to different shifted grids. Then it averages these piecewise constant functions over all those instances.

2.4.2. Ranking and Identification of Outliers

The algorithm assigns scores $\widetilde{R}$ and $\widehat{R}$ to any point $(x, \mathbf{y}) \in E$ by:

$$\widetilde{R}(x, \mathbf{y}) = \frac{\|\mathbf{y} - \widetilde{\mathbf{C}}(x)\|_2}{\widetilde{S}(x)} \quad \text{and} \quad \widehat{R}(x, \mathbf{y}) = \frac{\|\mathbf{y} - \widetilde{\mathbf{C}}(x)\|_2}{\widehat{S}(x)}\,.$$

It then assess the significance of outliers, by assigning p-values as follows:

$$p\text{-value}(x, \mathbf{y}) = \frac{2}{\sqrt{2\pi}} \int_{\widehat{R}(x,\mathbf{y})}^{\infty} e^{-\frac{t^2}{2}}\, \mathrm{d}t = \left(1 - \mathrm{erf}\left(\frac{\widehat{R}(x, \mathbf{y})}{\sqrt{2}} \right) \right). \tag{7}$$

Outliers are detected by controlling the False Discovery Rate (FDR) following Benjamini and Hochberg (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001). That is, given an FDR level q_0, it orders the computed p-values: $p_{(1)} \leq \ldots \leq p_{(N)}$ and sets

$$p^* = p\text{-value}\left(\max\left\{ i :\ p_{(i)} \leq q_0 \cdot \frac{i}{N} \right\} \right). \tag{8}$$

It then identifies as outliers the points with p-values less than or equal to p^*.

2.5. Refinement of the function F_Q

We suggest here a slight modification of the function F_Q. Numerical experiments with artificial and experimental data indicate that this modification usually yields slightly better or comparable identification results. However, our method for determining α_0 (described below in Section 2.6) seems to be more effective with our previously described version.

At each visited interval Q with left and right subintervals Q_L and Q_R, the modified algorithm computes

$$f_{Q_L} = \frac{|\mathrm{Out}(Q) \cap \mathrm{Str}(Q_L)|}{|\mathrm{Str}(Q_L)|} \quad \text{and} \quad f_{Q_R} = \frac{|\mathrm{Out}(Q) \cap \mathrm{Str}(Q_R)|}{|\mathrm{Str}(Q_R)|}.$$

The quantity F_Q is initialized for Q_0 by $F_{Q_0} \equiv 0$ for all $Q \in \mathcal{D}(Q_0)$. When visiting an interval Q (from coarsest level to finest level), the algorithm computes recursively F_{Q_L} and F_{Q_R} as follows:

$$F_{Q_L} = F_Q + f_{Q_L} \quad \text{and} \quad F_{Q_R} = F_Q + f_{Q_R}.$$

The stopping-time conditions are then applied independently in both intervals Q_L and Q_R.

2.6. Choice of the Main Parameter

The choice of most parameters of the algorithm is pretty straightforward (Lerman et al., 2007), except for the parameter α_0. In order to choose the optimal value of α_0 one may apply the MSC algorithm with different values of α_0 as well as p-values, p (or alternatively FDR-levels), and record the corresponding numbers of detected outliers, $N_{\mathrm{out}}(\alpha_0, p)$. The optimal value for α_0 is the most significant jump in the quantity $N_{\mathrm{out}}(\alpha_0, p)$ across various p-values. This strategy is justified in Appendix B. Different examples are demonstrated in Figure 2.

3. Properties of the MSC

Multiscale Influence and Bias of MSC

The $\mathbb{R}^D$-valued empirical influence function of the estimator $\widetilde{\mathbf{C}}$ at the empirical distribution P_N (of the N data points) is defined for any $(x, \mathbf{y}) \in \mathbb{R}^{D+1}$ by the formula

$$\mathrm{IF}_{\widetilde{\mathbf{C}}, P_N}(x) = \lim_{\epsilon \to 0+} \frac{\widetilde{\mathbf{C}}((1-\epsilon)P_N + \epsilon\delta_{(x,\mathbf{y})}) - \widetilde{\mathbf{C}}(P_N)}{\epsilon}.$$

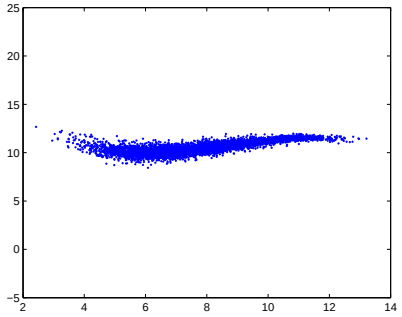
The empirical bias of the data point $(x, \mathbf{y})$ will refer here to the quantity $\|\mathbf{y} - \widetilde{\mathbf{C}}(x)\|_2$ (as opposed to the bias $\|\mathbf{y} - \mathbf{C}(x)\|_2$, where $\mathbf{C}$ is the true regression function.

We bound both the empirical influence function and the empirical bias of approximation at almost all data points in the following way.

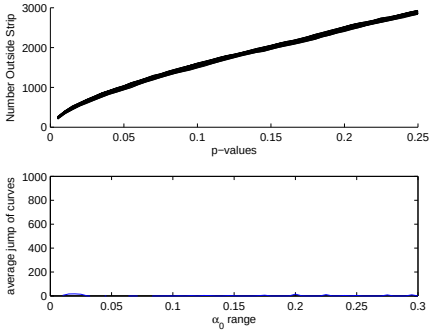
THEOREM 3.1. *Assume that the MSC is applied with the parameter $0 < \alpha_0 < 1$ to a data of N points with corresponding empirical distribution P_N. Then one can form a subset G of the data such that*

$$\frac{|G|}{N} > (1 - \alpha_0), \tag{9}$$

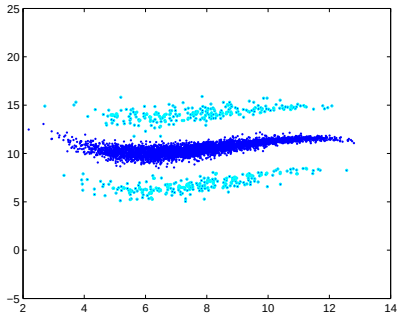
$$\mathrm{IF}_{\widetilde{\mathbf{C}}, P_N}(x) \le \frac{2 \cdot c_0}{\lambda_0} \quad \textit{for all} \quad x \in G, \tag{10}$$



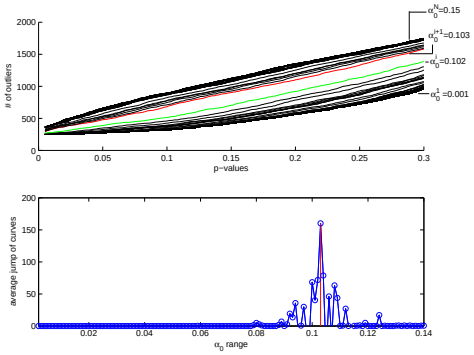
(a) data with no outlier component



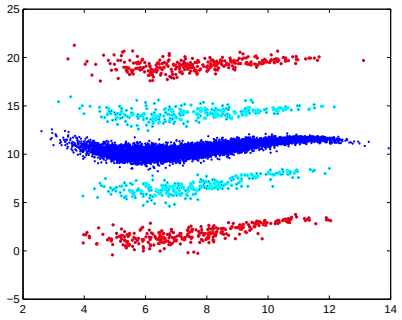
(b) profile curves and jump analysis for data in (a)



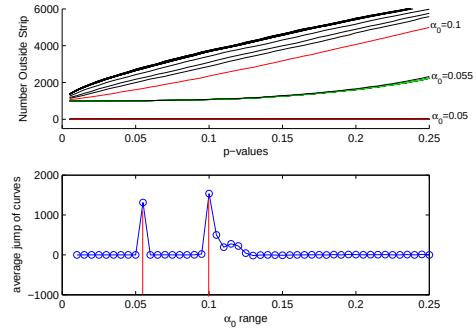
(c) data with one layer of symmetric outliers (10%)



(d) profile curves and jump analysis for data in (c)

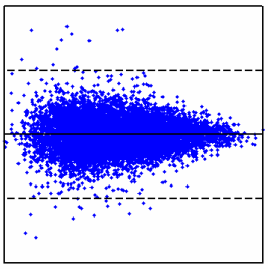


(e) data with two layers of symmetric outliers (each 5%)

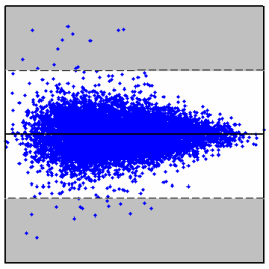


(f) profile curves and jump analysis for data in (e)

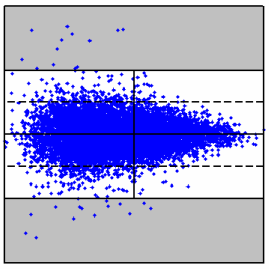
Fig. 2. Results of our method for determining α_0 for three artificial data.



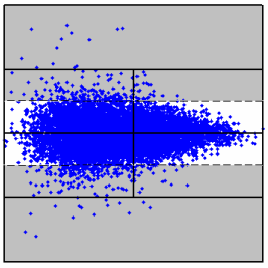
(a) Data around the line (interval Q_0); Cyl(Q_0) is a square



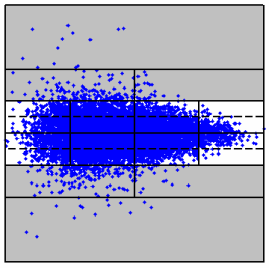
(b) Points in Out(Q_0) (gray area) are detected as "outliers" and excluded



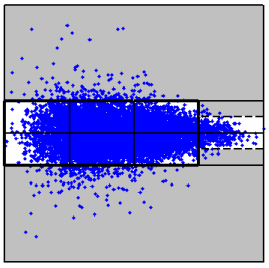
(c) 2^{nd} level of MSC: Q_0 partitioned into Q_1, Q_2 & regions formed



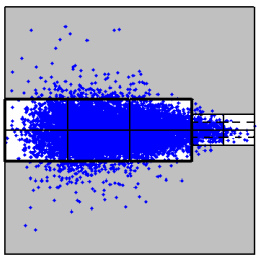
(d) New putative "outliers" in Out(Q_1) and Out(Q_2) (new gray area) are excluded



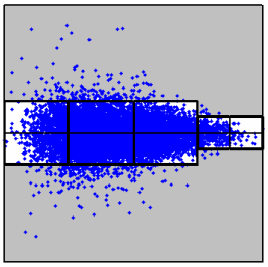
(e) 3rd level of algorithm: Further partition and identification of new regions



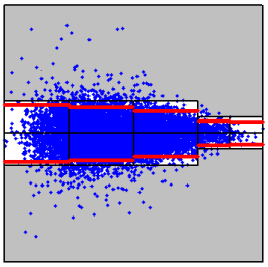
(f) Algorithm stops at 3 leftmost squares while continues at rightmost square



(g) 4th level of MSC, proceeding in two rightmost squares



(h) Final "stopping-time" squares, i.e., Cyl(Q) for "stopping-time" Q's



(i) Strip $\widetilde{S}$ (in red), estimating std in each "stopping-time" square

Fig. 3. Pictorial representation of how the algorithm works when applied to an artificial and simple set ($C = 0$).

and

$$\left\|\mathbf{y}-\widetilde{\mathbf{C}}(x)\right\|_2 \le 2\cdot c_0\cdot\frac{1}{n_{\rm sh}}\sum_{i=1}^{n_{\rm sh}}\ell\left(Q^i_{\rm stop}(x)\right)\quad \textit{for all}\ \ x\in G, \tag{11}$$

where $\ell(Q^i_{\rm stop}(x))$ *is the length of the unique stopping-time interval containing* x *in the* i^{th} *shift of the initial grid.*

We prove Theorem 3.1 in Appendix C. Here we demonstrate the significance of its statement by comparing MSC with another common method. This method starts with the same initial transformation (shifting the coordinates of $\mathbf{y}$ by those of the regression line). It then computes the interval Q_0 as in Subsection 2.1, partitions Q_0 into intervals of equal length, and excludes from the points projected onto those intervals their top and bottom $\alpha_0/2$-quantiles. The set G is the union of the points left at all local intervals. The curve $\widetilde{\mathbf{C}}$ is then formed by local linear regressions of the points in the new regions around the local intervals (i.e., excluding the top and bottom $\alpha_0/2$-quantiles).

Clearly, the new set G satisfies equation (9). However, the empirical influence function of points in it can be rather large in some places. In particular, as the size of all these intervals decreases, the empirical influence function often increases.

Assume next that the latter algorithm uses the whole interval Q_0 without partitioning it (while removing points at the top and bottom $\alpha_0/2$-quantiles of the whole set). In this case, equations (9) and (10) still hold, but the empirical bias of approximation is rather large. On the other hand, the MSC has smaller empirical biases at smaller stopping-time intervals as expressed by equation (11). In view of this example and Theorem 3.1, we observe that the MSC algorithm is most powerful when having relatively small stopping-time scales. Indeed, it then obtains small empirical bias of approximation while maintaining relatively small empirical influences (except at a sufficiently small and designated set).

Complexity of the algorithm

The computational complexity of the algorithm is summarized as follows and proved in Appendix D.

PROPOSITION 3.1. *The storage and speed of the MSC algorithm for a data set of* N *points in* $\mathbb{R}^D$*, when using* ℓ_0 *levels and* $n_{\rm sh}$ *shifts, are of order* $O(N\cdot D)$ *and* $O(N\cdot D\cdot\ell_0\cdot n_{\rm sh})$ *respectively.*

On the Quality of Smoothing

Here we assume two additional stopping-time conditions with the parameters $\delta_0,\theta_0>0$ at any visited interval Q:

$$\sigma_{\mathbf{Y}|X}(Q)>\delta_0\cdot\ell(Q), \tag{12}$$

$$L_Q\cap x\text{-axis}\neq\emptyset\ \ \text{and}\ \ \tan(\text{ang}(L_Q,x\text{-axis}))>\theta_0. \tag{13}$$

Using those two optional conditions and extending the definition of the set $\mathcal{B}$ as follows

$$\mathcal{B}=\{Q\in\mathcal{Q}:\ \ F_Q<\alpha_0 \text{ or any of equations (3), (12),(13) is satisfied}\},$$

we claim that the algorithm produces smooth estimators $\widetilde{\mathbf{C}}$ and $\widetilde{S}$ in the following way (the proof is in appendix E).

PROPOSITION 3.2. *If the MSC algorithm is applied with the additional stopping-time conditions of equations* (12) *and* (13) *(with parameters* δ_0 *and* θ_0 *respectively). Then as the number of shifts,* n_{sh}*, approaches infinity the estimators* $\widetilde{\mathbf{C}}$ *and* $\widetilde{S}$ *converge to Lipschitz functions with constants not larger than* $2 \cdot \sqrt{D-1} \cdot \theta_0$ *and* $4 \cdot \delta_0 \cdot (l_0 + 1)$ *respectively.*

4. Numerical Experiments

4.1. Simulated Data

We evaluate the MSC algorithm on simulated data, while comparing it with other common and powerful techniques. Since those other methods mainly apply to planar data we take $D = 1$.

We fix two arbitrary parameters $\varepsilon, r > 0$ and generate data by sampling N i.i.d. random variables from a distribution on $\mathbb{R}^2$ with pdf of the form

$$f(x, y) = \varepsilon \cdot f_{\mathrm{in}}(x, y) + (1 - \varepsilon) \cdot f_{\mathrm{out}}(x, y)\,.$$

The "stable" pdf, $f_{\mathrm{in}}(x, y)$, is formed by a mixture of ten two-dimensional Gaussian distributions around the curve

$$C(x) = 10 + \sqrt{x}\sin^2(rx)\,.$$

The "outliers" pdf, $f_{\mathrm{out}}(x, y)$, is also a mixture of two distributions (with equal weights). Each one of these is obtained by adding and subtracting respectively a 10-dimensional vector from the vector of local means of the inner pdf $f_{\mathrm{in}}(x, y)$. The software used to generate such samples is available at http://www.math.umn.edu/∼lerman/supp/multistrip. A sample simulated that way is exemplified in Figure 4.

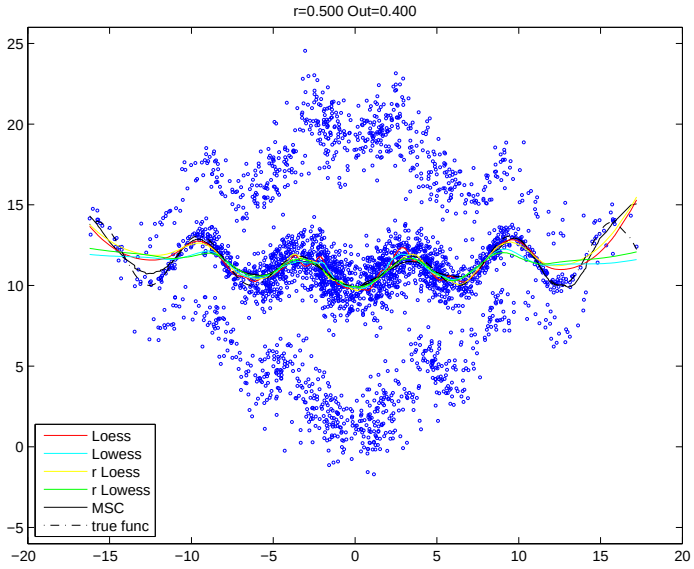


Fig. 4. Illustration of simulated data with $r = 0.5$, $\varepsilon = 0.4$ and the estimators $\widetilde{C}$ by MSC and four versions of LOESS

Table 1. Comparison of MSSE and CPU time for MSC (both optimal α_0 and $\alpha_0 = 0.2$), 'loess', 'rloess', 'lowess' and 'rlowess', using 3000 points of synthetically created data. CPU time in seconds recorded in Matlab® using a dual processor Intel Core 2.66GHz with 2GB of RAM; MSSE is averaged for each ε over all $r = \{0.25, 0.33, 0.42, 0.50, 0.58, 0.67, 0.75, 0.83, 0.92, 1\}$.

% outliers (ε)		MSC	MSC ($\alpha_0 = 0.2$)	'loess'	'rloess'	'lowess'	'rlowess'
1	CPU	0.88	1.08	1.84	6.86	1.67	5.82
	MSSE	7.02	7.36	10.98	10.44	11.17	10.45
5	CPU	0.86	1.13	1.93	6.72	1.71	5.77
	MSSE	6.85	7.15	14.63	10.99	14.21	10.56
10	CPU	0.85	1.07	2.11	6.99	1.82	5.69
	MSSE	7.06	7.50	17.64	11.80	16.97	11.17
20	CPU	0.86	0.98	2.32	7.02	1.91	5.97
	MSSE	7.58	7.77	20.36	11.76	19.58	11.90
30	CPU	0.98	0.82	2.41	7.55	1.95	6.26
	MSSE	8.15	9.11	23.12	12.68	21.97	12.52
40	CPU	0.98	0.60	2.47	9.53	2.03	7.27
	MSSE	9.68	16.91	24.32	17.17	23.25	16.63

We applied MSC to such data with the refinement of F_Q described in Subsection 2.5. We compared it with four different instances of the LOESS algorithm (Cleveland, 1979; Cleveland and Devlin, 1988; Cleveland and Loader, 1996). Those instances use the following options in the Matlab® function *smooth*: 'lowess', 'loess', 'rlowess', 'rloess'. The first one, 'lowess', is a local regression algorithm using weighted linear least squares and a first degree polynomial model. The second one, 'loess', is similar but uses a second degree polynomial model. The other two, 'rlowess' and 'rloess', are robust versions of 'lowess' and 'loess' respectively that assign lower weights to outliers.

Our comparison between the different algorithms is based on the quality of estimating C, which is measured by the Mean Sum of Squares Error (MSSE). Specifically, we have drawn 30 i.i.d. samples of data and recorded the corresponding values of the estimated local means by $\widetilde{C}^i$, $i = 1, \ldots, 30$. We then compute the MSSE by the formula

$$\mathrm{MSSE}(\widetilde{C}) := \sqrt{\frac{1}{30N} \sum_{i=1}^{30} \sum_{j=1}^{N} \|C(x_j) - \widetilde{C}^i(x_j)\|_2^2}\,.$$

We have tested the different algorithms for different values of ε while averaging over different r values ($r = 0.25, 0.33, 0.42, 0.50, 0.58, 0.67, 0.75, 0.83, 0.92, 1$). For all of the four versions of LOESS we have optimized the smoothing bandwidth h so that it minimizes the lowest MSSE. On the other hand, for MSC we have both optimized α_0 and fixed $\alpha_0 = 0.2$.

The values of the MSSE and the actual CPU times for the different algorithms are recorded in Table 1. We have used Matlab® and a Dual processor Intel Core 2.66GHz with 2GB of RAM.

We note that the MSSE of MSC is significantly lower than the MSSE of the other common algorithms for robust regression. We also note that the MSC algorithm is faster than the four other methods. Indeed, conventional implementations of local smoothers are $O(N^2 \cdot h)$, where h is the kernel span (B. Seifert, 1994). Robust smoothers, in particular, 'rlowess' and 'rloess', perform several re-weighted iterations at each point, adding a (pos-

sibly) hefty coefficient onto the number of flops. On the other hand the MSC algorithm is $O(N \cdot \ell_0 \cdot n_{\rm sh})$ (see Proposition 3.1).

4.2. Edge Detection via MSC for High-dimensional Pixel Data

We also consider high-dimensional data sets of pixel neighborhoods (equivalently patches) of gray-level images. We map those $k \times k$ patches back to the original image by assigning to each one of them its "center" (it is the exact center if k is odd). We assume that such data is concentrated around a curve with outliers that correspond to edges in the original image. This assumption can be verified in two different ways: First of all, by viewing the projection of the data and detected curves and strips on two dimensional or three dimensional spaces. Second of all, by checking whether the detected outliers indeed correspond to edges of the image.

Figure 5 demonstrates the edge detection obtained by MSC with three different images. We applied the following percentages of outliers: $10\%, 20\%, 30\%$ and the following sizes of pixel neighborhoods: $k = 3, 5, 7, 9$.

Figure 6 explores the effect of additive noise (with SNR of -3.52dB) on the MSC algorithm and other algorithms used by the ©Matlab's EDGE function. We remark that for high levels of noise, it is instructive to use relatively high values for the pixel neighborhoods, e.g., 7, 9 and 11.

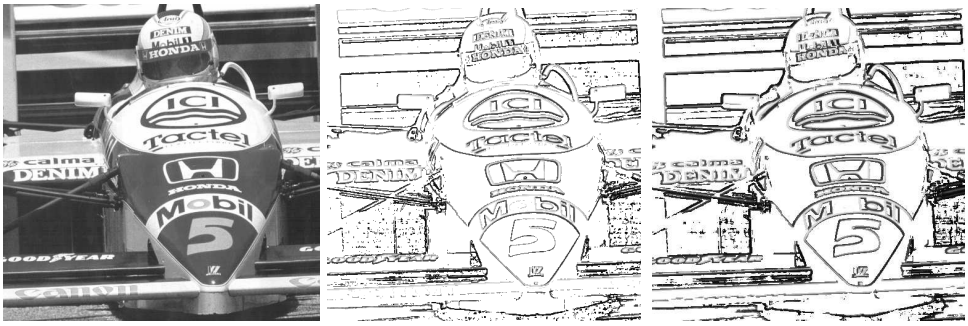
5. Discussion

We conclude this work describing three possible extensions of interest to us.

More general regression (higher-dimensional and smoother): It is not hard to generalize the framework presented here to include the regression of the last $D - d + 1$ coordinates on the first d of them, where $d > 1$. For this purpose dyadic intervals along the line are replaced by dyadic partitions of d-dimensional cubes or rectangles. Also one can allow higher order of smoothness of the output of the MSC. For example, by locally fitting higher-order polynomial approximations and applying partitions of unity smoother than the characteristic functions.

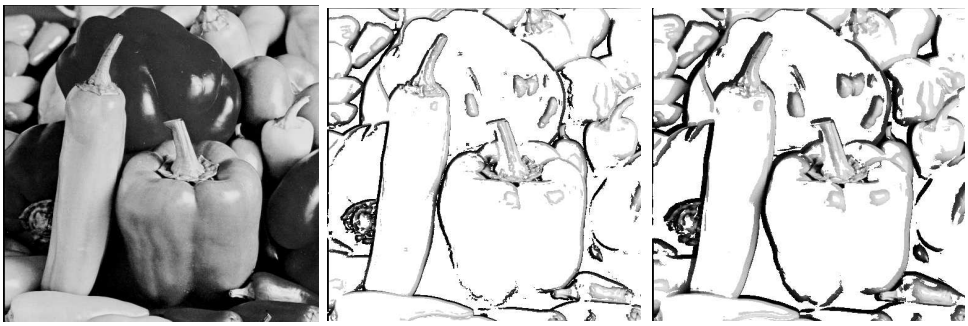
Further theoretical justification of MSC: Theorem 3.1 shows that the MSC algorithm is most successful when the sizes of the stopping-time intervals are sufficiently small. An interesting and nontrivial theoretical problem is to determine how effective the MSC algorithm in getting to sufficiently small scales (talking into account restrictions imposed by large local variances). Peter Jones and the first author of this paper have partially addressed this question, while modifying some steps of the original algorithm and using a toy model where $C = 0$ (i.e., $E(\mathbf{Y}|X) = 0$) and the underlying "standard deviation", S, is a general A_p weight (which is defined e.g., in Stein (1993)). The mathematical details of this answer are beyond the scope of this paper. We are curious about other related analysis.

The structure of high-dimensional pixel data: We are interested in a careful study of the properties of the main curve and the outliers of pixel neighborhood data for a large database of images. In particular, we would like to know the dependence of such features on the images (for example, pure texture images show a somewhat different behavior). We also remark that this framework could be further generalized to color images, where the



(a) Race car (b) 20% outliers ("edges"), $k = 3$ (c) 30% outliers ("edges"), $k = 5$

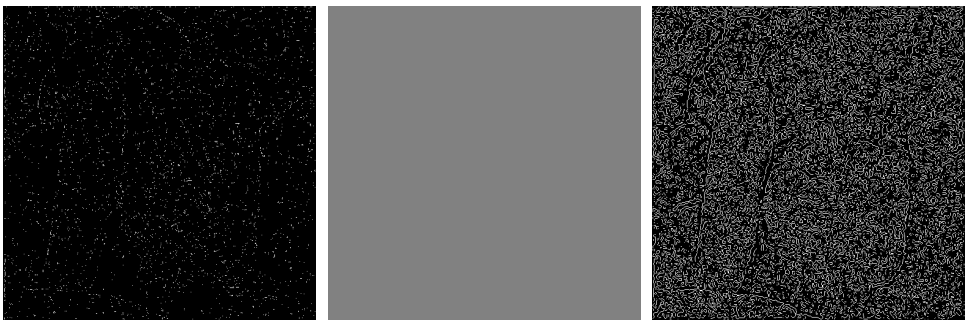
(d) Paoulina (e) 10% outliers ("edges"), $k = 5$ (f) 20% outliers ("edges"), $k = 7$



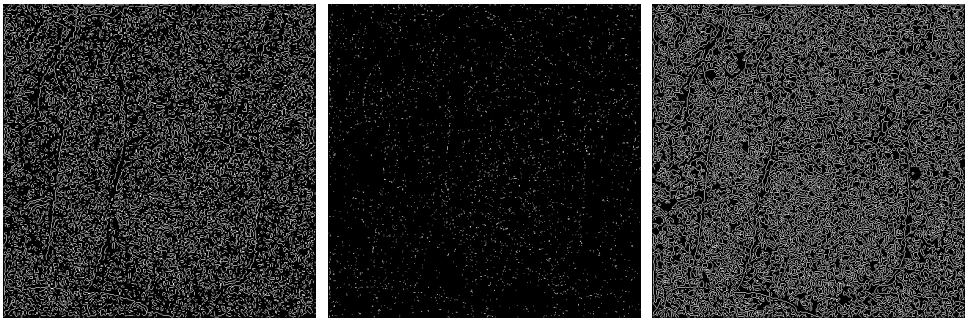
(g) peppers (h) 20% outliers ("edges"), $k = 5$ (i) 30% outliers ("edges"), $k = 9$

Fig. 5. Edges via MSC for three different images and various parameters.

(a) Peppers with Gaussian noise (b) 10% outliers ("edges"), $k = 9$ (c) 20% outliers ("edges"), $k = 11$



(d) "Prewitt" method (e) "Roberts" method (f) "Sobel" method



(g) "Log" method (h) "Zerocross" method (i) "Canny" method

Fig. 6. Edge detection for gray-level peppers with additive noise drawn from $N(0, \frac{3}{2} \cdot \sigma)$, where σ is the standard deviation of all pixel values. Algorithms used: MSC and those of ©Matlab's EDGE function. Here MSC first projected the data onto its top 9 principal components.

main curve is usually replaced by a mixture of three curves which are relatively flat (hybrid flat models for such data are studied by Huang et al. (2004)).

Acknowledgments: We would like to thank Alexandre Blais and Brian D. Dynlacht for introducing us to problems in the analysis of ChIP-on-chip microarrays, where we have used a version of the algorithm described here. We thank Ronald R. Coifman, James Glimm, Peter W. Jones and Yuhong Yang for commenting on earlier versions of this paper. Special thanks to Guangliang Chen for useful discussions and for improving our software. Special thanks to Mark Green and IPAM (UCLA) for inviting GL and JM to take part in their bioinformatics as well as multiscale geometry meetings, where discussions of similar topics stimulated our research. GL would like to thank David L. Donoho for supporting his visit to Stanford and his hospitality at the time of developing this work. This research has been supported by NSF grant #0612608 (GL) as well as grants (obtained by BM) from NSF's ITR program, Defense Advanced Research Projects Agency (DARPA), and New York State Office of Science, Technology & Academic Research (NYSTAR).

A. Explanation of Formula (6)

Formula (6) is based on the following observation: If $b > 0$, $X \sim N(0, \widehat{\sigma}^2)$ is a normal random variable with density function $f = \frac{1}{\sqrt{2\pi \cdot \widehat{\sigma}^2}} \cdot e^{-\frac{x^2}{2 \cdot \widehat{\sigma}^2}}$ and ${\sigma_b}^2 := \int_{-b}^{b} x^2 f(x)\,\mathrm{d}x$, then

$$ {\sigma_b}^2 = -\frac{2}{\sqrt{2\pi}} \cdot \widehat{\sigma} \cdot b \cdot e^{-\frac{b^2}{2 \cdot \widehat{\sigma}^2}} + \frac{\widehat{\sigma}^2}{2} \cdot \left(\mathrm{erf}\left(\frac{b}{\sqrt{2} \cdot \widehat{\sigma}} \right) - \mathrm{erf}\left(\frac{-b}{\sqrt{2} \cdot \widehat{\sigma}} \right) \right). $$

This observation immediately follows from the following integration by parts:

$$ {\sigma_b}^2 = \int_{-b}^{b} x^2 f(x)\,\mathrm{d}x = \frac{-\widehat{\sigma}^2}{\sqrt{2\pi\widehat{\sigma}^2}} \int_{-b}^{b} x \cdot \frac{\mathrm{d}}{\mathrm{d}x} e^{-\frac{x^2}{2\widehat{\sigma}^2}}\,\mathrm{d}x = \frac{-\widehat{\sigma}}{\sqrt{2\pi}} \left[x e^{-\frac{x^2}{2\widehat{\sigma}^2}} \right]_{-b}^{b} + \widehat{\sigma}^2 \int_{-b}^{b} f(x)\,\mathrm{d}x. $$

B. Justification of the Procedure for Selecting α_0

We justify our method for choosing α_0 in the setting of planar data (i.e., $D = 1$) generated with a constant percentage of outliers independent of the location x. In order to do it, we need to understand for any given value of α_0 the dependence of N_{out}, the number of outliers detected by the algorithm, to the threshold p-value. Equivalently, we need to understand for any given value of α_0, the dependence of N_{out} on the constant $B \equiv B(\alpha_0)$ such that outliers are identified outside the strips $\widetilde{\mathbf{C}}(x) + B \cdot \widehat{S}(x)$ and $\widetilde{\mathbf{C}}(x) - B \cdot \widehat{S}(x)$.

We start by considering such dependence and its sensitivity to changes of α_0 in the very special case where there is no outlier component in our model. In this case, it is clear that the curves describing the dependence of N_{out} on B, or on the corresponding p-values, vary continuously with the parameter α_0. In Figure 2(a) we describe a sample from this model, whereas in Figure 2(b) we show that our method does not detect any jump, i.e., no layer of outliers.

Assume next that the underlying curve lies on the x-axis, i.e., the regression function is $\mathbf{y} = \mathbf{0}$. Moreover, the distribution is a mixture of two homoscedastic components: stable and outlier, and the weight of the latter component is p_{out}. We denote the constant standard variation of the stable component by σ_{in}. We assume that the outlier component is

also bimodal and symmetric around 0 with constant means $\pm\mu$ and constant conditional variances around each mode, such that $\sigma_{\text{in}}, \sigma_{\text{out}} \lessapprox \mu$. Note that if $\alpha_0 > p_{\text{out}} + \varepsilon$, where $\varepsilon > 0$ is a sufficiently small constant, then we expect the algorithm to peel out most of the outliers and thus use mainly points sampled from the stable distribution to estimate the stable standard deviation. Therefore, in a similar way to the case above ($p_{\text{out}} = 0$), we expect continuous variation in the profile curves. Similarly, if $\alpha_0 < p_{\text{out}} - \varepsilon$, the algorithm uses points sampled from all the three components of the trimodal distribution. The estimate of the standard deviation varies continuously with α_0 and we thus expect continuous variation in the profile curve. However, when $\alpha_0 \sim p_{\text{out}}$, we encounter a transition from an underlying unimodal distribution to trimodal distribution (within the strip $\widetilde{\mathbf{C}}(x) \pm B \cdot \widehat{S}(x)$). Therefore, for a fixed B (sufficiently large) we expect a jump in the number of outliers detected by the algorithm when transitioning from $p_{\text{out}} - \varepsilon$ to $p_{\text{out}} + \varepsilon$. One can extend this argument to a more general setting. In Figure 2(c) we describe a sample from a trimodal distribution, where the true regression curve C is not a line. Next, in Figure 2(d) we show that our method detects a jump around $\alpha_0 = p_{\text{out}} = 0.1$.

The argument above also applies to several modes of outliers and explains how one can get a few jumps of profile curves. In Figure 2(e) we describe a sample from a distribution with two layers of outliers, whereas in Figure 2(f) we show that our method detects two jumps, around the expected fraction of layers of outliers: 5% and 10%.

If we apply the output function $\widetilde{S}$ (instead of $\widehat{S}$), then additional artificial jumps may occur. This happens due to discontinuities in the sizes of the regions $\{\text{Cyl}(Q)\}$ and the corresponding discontinuities in the standard deviations estimated in those regions. The application of $\widehat{S}$ corrects the problem by extending the estimates of the standard deviations outside the restricted regions. However, we suspect that it is possible to have artificial jumps when the underlying stable distribution is far from the normal assumption. Nevertheless, we have simulated various artificial data sets with distributions far from normal and could not observe such a phenomenon when using $\widehat{S}$.

Our experience with real data sets (Lerman et al., 2007) indicated that jumps could occur at a whole range of α_0 values. We believe that they represent local changes in the densities, but it is also possible that they are artificial as discussed above. Consequently, we have decided to choose in such cases the value of α_0 corresponding to the first significant jump in the number of outliers (instead of most significant one).

C. Proof of Theorem 3.1

We assume that $n_{\text{sh}} = 1$. The general case of $n_{\text{sh}} \geq 1$ is obtained by averaging the estimates below (adapted to different shifts when needed). We also assume that Q_0 is not a stopping interval (which is the case with the appropriate choice of parameters).

We form the function $\text{Stb}(x, \mathbf{y})$ and the desired set G as follows:

$$\text{Stb}(x, \mathbf{y}) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} \chi_{\text{Cyl}(Q)}(x, \mathbf{y}) + \sum_{Q \in \mathcal{B}} \chi_{\text{Cyl}(P_Q) \cap \text{Str}(Q)}(x, \mathbf{y}) \,,$$

$$G = \{(x, \mathbf{y}) \in E : \text{Stb}(x, \mathbf{y}) = 1\} \,.$$

C.1. Proof of Equation (9)

We first note the following relation between the function Stb and the regions $\{\mathrm{Out}(Q)\}_{Q\in\mathcal{P}}$:

$$\{(x,\mathbf{y})\in Q_0\times\mathbb{R}^D : \mathrm{Stb}(x,\mathbf{y})=0\} = \mathrm{Str}(Q_0)\setminus\left(\bigcup_{Q\in\mathcal{Q}\setminus\mathcal{B}}\mathrm{Cyl}(Q)\bigcup\bigcup_{Q\in\mathcal{B}}(\mathrm{Cyl}(P_Q)\cap\mathrm{Str}(Q))\right)$$
$$\subseteq \mathrm{Str}(Q_0)\setminus\bigcup_{Q\in\mathcal{Q}}\mathrm{Cyl}(Q)=\bigcup_{Q\in\mathcal{P}}\mathrm{Out}(Q)\,.$$

We denote by Π_L the projection operator from $\mathbb{R}^D$ onto L and observe that

$$|E\setminus G|\le\sum_{Q\in\mathcal{P}}|\mathrm{Out}(Q)|=\sum_{Q\in\mathcal{P}}f_Q\,|\mathrm{Str}(Q)|=\sum_{Q\in\mathcal{P}}f_Q\sum_{x\in E}\chi_{\mathrm{Str}(Q)}(x)=\sum_{Q\in\mathcal{P}}f_Q\sum_{x\in\Pi_L(E)}\chi_Q(x)$$
$$=\sum_{x\in\Pi_L(E)}\sum_{Q\in\mathcal{P}}f_Q\cdot\chi_Q(x)\le\sum_{x\in\Pi_L(E)}\sum_{Q\in\mathcal{Q}}F_{P_Q}(x)\le\sum_{x\in\Pi_L(E)}\alpha_0=\alpha_0\cdot|\Pi_L(E)|=\alpha_0\cdot|E|.$$

Consequently,

$$|G|\ge(1-\alpha_0)\cdot|E|=(1-\alpha_0)\cdot N\,.$$

C.2. Proof of equations (10) *and* (11)

Equation (11) immediately follows from the definitions of the set G and the regions $\{\mathrm{Cyl}(Q)\}$, $\{\mathrm{Cyl}(P_Q)\}$. For the rest of this subsection we establish equation (10).

We assume first that $Q\in\mathcal{Q}\setminus\mathcal{B}$ and denote $R(Q)=\mathrm{Str}(Q)\cap\mathrm{Cyl}(P_Q)$. We recall that L_Q is the least squares regression line for $R(Q)$. If $(x_i,\mathbf{y}_i)\in\mathrm{R}(Q)\cap E$, then by applying the formula for empirical influence of linear regression (Cook and Weisberg, 1982, Section 3.4.1) we obtain that

$$\|\mathrm{IF}_{\widetilde{\mathbf{C}},P_N}(x_i,\mathbf{y}_i)\|_2=\frac{\left(x_i-\frac{1}{|R(Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(Q)}x_i'\right)\|y_i-\widetilde{\mathbf{C}}(x_i)\|_2}{\frac{1}{|R(Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(Q)}\left(x_i'-\frac{1}{|R(Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(Q)}x_i'\right)^2}\le\frac{|Q|\,2\,c_0\,|Q|}{\lambda_0\,|Q|^2}=\frac{2\,c_0}{\lambda_0}.$$

Next, we assume that $Q\in\mathcal{B}$, so that its corresponding local regression line was computed in $R(P_Q)=\mathrm{Str}(P_Q)\cap\mathrm{Cyl}(P_{P_Q})$. In this case, if $(x_i,\mathbf{y}_i)\in R(P_Q)\cap E$, then

$$\|\mathrm{IF}_{\widetilde{\mathbf{C}},P_N}(x_i,\mathbf{y}_i)\|_2=\frac{\left(x_i-\frac{1}{|R(P_Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(P_Q)}x_i'\right)\|y_i-\widetilde{\mathbf{C}}(x_i)\|_2}{\frac{1}{|R(P_Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(P_Q)}\left(x_i'-\frac{1}{|R(P_Q)|}\sum_{(x_i',\mathbf{y}_i')\in R(P_Q)}x_i'\right)^2}\le\frac{2\,c_0}{\lambda_0}\,.$$

D. Proof of Proposition 3.1

Since the shifted grids are processed independently and the regions within each grid are disjoint, the storage of the MSC algorithm is of order $O(N\cdot D)$.

In order to bound the time complexity of the algorithm, we first restrict it to the grid $\mathcal{D}(Q_0)$. We note that the computation performed in the region $\mathrm{Cyl}(Q)$ (for any interval Q considered by the algorithm) is of order $O(|\mathrm{Cyl}(Q)|\cdot D)$. Indeed, the main computation at each interval involves finding two $\mathbb{R}^D$-valued least squares regression lines of $|\mathrm{Cyl}(Q)\cap\mathrm{Str}(Q_L)|$ and $|\mathrm{Cyl}(Q)\cap\mathrm{Str}(Q_R)|$ points respectively. We remark that when $Q=Q_0$, the

additional computation of the initialization step (described in the beginning of Section 2) is also accounted. Therefore, the speed of the algorithm, when using only one grid, is of order

$$D\cdot\sum_{Q\in\mathcal{Q}\cup\mathcal{P}}|\mathrm{Cyl}(Q)|\le D\cdot\sum_{j=0}^{\ell_0}\sum_{\substack{Q\in\mathcal{Q}\cup\mathcal{P}\\ \ell(Q)=2^{-j}\cdot\ell(Q_0)}}|\mathrm{Cyl}(Q)|\le D\cdot(\ell_0+1)\cdot|\mathrm{Str}(Q_0)|=D\cdot(\ell_0+1)\cdot N.$$

We conclude the proposition by noting that the above estimate holds for any of the n_{sh} shifted grids.

E. Proof of Proposition 3.2

For simplicity of notation, we shift and scale the data so that $Q_0^*=[0,1]$. For any $0\le\gamma\le 1$, we denote $Q_\gamma=[-\gamma,2-\gamma)$. Let $\widetilde{S}_\gamma$ and $\widetilde{\mathbf{C}}_\gamma$ be the stepwise constant and linear functions respectively corresponding to the dyadic grid $\mathcal{D}(Q_\gamma)$. We define

$$\widetilde{S}_T(x)=\int_0^1\widetilde{S}_\gamma(x)\,\mathrm{d}\gamma\quad\text{and}\quad\widetilde{\mathbf{C}}_T=\int_0^1\widetilde{\mathbf{C}}_\gamma(x)\,\mathrm{d}\gamma.$$

We will prove Proposition 3.2 by showing that

$$\|\widetilde{S}_T\|_{\mathrm{Lip}}\le 4\cdot\delta_0\cdot(l_0+1)\tag{14}$$

and

$$\|\widetilde{\mathbf{C}}_T\|_{\mathrm{Lip}}\le 2\cdot\sqrt{D-1}\cdot\theta_0\,.\tag{15}$$

We first prove equation (14). For each fixed $0<\gamma\le 1$, we denote by $\mathcal{Q}(Q_\gamma)$ the set of all stopping-time intervals with respect to the grid $\mathcal{D}(Q_\gamma)$. For each interval $Q\in\mathcal{Q}(Q_\gamma)$, we denote by $\widetilde{S}_Q$, the value of the function $\widetilde{S}$ on the interval Q ($\widetilde{S}_Q=\sigma_{\mathbf{Y}|X}(Q)$ if $Q\in\mathcal{Q}\setminus\mathcal{B}$ and $\widetilde{S}_Q=\sigma_{\mathbf{Y}|X}(P_Q)$ otherwise). We also denote the left and right value of any interval $Q\in\mathcal{Q}(Q_\gamma)$ by a_Q and b_Q, that is, $Q=[a_Q,b_Q)$. Following the above notation we represent $\widetilde{S}_T$ as follows:

$$\widetilde{S}_T(x)=\int_0^1\sum_{Q'\in\mathcal{Q}(Q_\gamma)}\widetilde{S}_{Q'}\cdot\chi_{Q'}(x)\,\mathrm{d}\gamma=\int_0^1\sum_{Q'\in\mathcal{Q}(Q_\gamma)}\widetilde{S}_{Q'}\cdot\chi_{[a_{Q'},b_{Q'})}(x)\,\mathrm{d}\gamma.$$

The distributional derivative of $\widetilde{S}_{T,j}$ obtains the following form:

$$\frac{\partial}{\partial x}\widetilde{S}_T(x)=\int_0^1\sum_{Q'\in\mathcal{Q}(Q_\gamma)}\widetilde{S}_{Q'}\cdot(\delta(x-a_{Q'})-\delta(-x+b_{Q'}))\,\mathrm{d}\gamma,\tag{16}$$

where $\delta(\cdot)$ is the Dirac distribution. Denote

$$D_+(x)=\int_0^1\sum_{Q'\in\mathcal{Q}(Q_\gamma)}\widetilde{S}_{Q'}\cdot\delta(x-a_{Q'})\,\mathrm{d}\gamma\quad\text{and}\quad D_-(x)=\int_0^1\sum_{Q'\in\mathcal{Q}(Q_\gamma)}\widetilde{S}_{Q'}\cdot\delta(-x+b_{Q'})\,\mathrm{d}\gamma.$$

Next, we bound the function $D_+(x)$. Let $S_{hft}(x)$ denote the set of all possible shifts γ of $[0,1)$, so that there exists an interval Q' in $\mathcal{D}(Q_\gamma)$ such that $x = a_{Q'}$. Also if $\gamma \in S_{hft}(x)$, then we denote by $Q(x,\gamma)$ the interval Q' in $\mathcal{D}(Q_\gamma)$ such that $x = a_{Q'}$. Using this notation we may write D_+ as follows:

$$D_+(x) = \sum_{\gamma(x)\in S_{hft}(x)} \widetilde{S}_{Q(x,\gamma(x))}.$$

We note that the stopping-time condition of equation (12) and the definition of $\widetilde{S}_Q$ (for any interval Q) imply that

$$\widetilde{S}_{Q(x,\gamma(x))} \le 2\cdot\delta_0\cdot\ell(Q(x,\gamma(x))). \tag{17}$$

Furthermore, we have that for any $x \in [0,1)$, the set $S_{hft}(x)$ may contain only one of the following points: $\gamma_{k,j}(x) = -x + k\cdot 2^{-j+1}$, $j = 0,\ldots,l_0$, $k = 0,\ldots,2^j-1$. That is, for any level j ($j = 0,\ldots,l_0$), there are at most 2^j possible elements of $S_{hft}(x)$ whose corresponding coefficients ($\widetilde{S}_{Q(x,\gamma(x))}$, where $\gamma \in S_{hft}(x)$) are not exceeding $2\cdot\delta_0\cdot 2^{-j}$ (as implied by equation (17)). Consequently,

$$D_+ \le 2\cdot\delta_0\cdot\sum_{j=0}^{l_0} 2^{-j}\cdot 2^j \le 2\cdot\delta_0\cdot(\ell_0+1).$$

Similarly we obtain the same upper bound on $-D_-$ and thus conclude that

$$\left|\frac{\partial}{\partial x}\widetilde{S}_T(x)\right| \le 4\cdot\delta_0\cdot(l_0+1). \tag{18}$$

The Lipschitz bound for $\widetilde{\mathbf{C}}$ is established similarly. Indeed, note that the formal analog of equation (16) for $\frac{\partial}{\partial x}\widetilde{\mathbf{C}}_T(x)$ has the following form:

$$\frac{\partial}{\partial x}\widetilde{\mathbf{C}}_T(x) = \int_0^1 \sum_{Q'\in\mathcal{Q}(Q_\gamma)} \underline{L}_{Q'}\cdot(\delta(x-a_{Q'}) - \delta(-x+b_{Q'}))\,\mathrm{d}\gamma + \int_0^1 \sum_{Q'\in\mathcal{Q}(Q_\gamma)} \frac{\partial}{\partial x}\underline{L}_{Q'}\cdot\chi_{Q'}.$$

Equation (13) guarantees a bound similar to that of equation (18) on the first term of the above equation, where the constant δ_0 is replaced by θ_0. It also implies that for all $i = 2,\cdots,D$: $|(\frac{\partial}{\partial x}\underline{L}_{Q'}(x))_i| \le \theta_0$ and thus equation (15) is also concluded.

References

Arias-Castro, E., D. Donoho, and X. Huo (2006). Adaptive multiscale detection of filamentary structures embedded in a background of uniform random points. *Annals of Statistics 34*(1), 326–349.

B. Seifert, M. Brockman, J. E. T. G. (1994). Fast algorithms for non-parametric curve estimation. *Journal of Computational and Graphical Statistics 2*, 192–213.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B 57*(1), 289–300.

Benjamini, Y. and D. Yekutieli (2001). The control of the false discovery rate in multiple testing under dependency. *Ann. Statist. 29*(4), 1165–1188.

Bishop, C. J. and P. W. Jones (1994). Harmonic measure, L^2 estimates and the Schwarzian derivative. *J. Anal. Math. 62*, 77–113.

Buck, M. J. and J. D. Lieb (2004). Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics 83*, 349–360.

Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association 74*, 829–836.

Cleveland, W. S. and S. J. Devlin (1988). Locally-Weighted Regression: An Approach to Regression Analysis by Local Fitting. *journal of the American Statistical Association 83*, 596–610.

Cleveland, W. S. and C. L. Loader (1996). Smoothing by Local Regression: Principles and Methods. In W. Härdle and M. G. Schimek (Eds.), *Statistical Theory and Computational Aspects of Smoothing*, pp. 10–49. New York: Springer.

Cook, R. D. and S. Weisberg (1982). *Residuals and influence in regression*. Monographs on Statistics and Applied Probability. London: Chapman & Hall.

Huang, K., A. Y. Yang, and Y. Ma (2004). Sparse representation of images with hybrid linear models. In *ICIP*, pp. 1281–1284.

Huber, P. J. (1981). *Robust statistics*. New York: John Wiley & Sons Inc. Wiley Series in Probability and Mathematical Statistics.

Lerman, G. (2003). Quantifying curvelike structures of measures by using L_2 Jones quantities. *Comm Pure App Math 56*(9), 1294–1365.

Lerman, G., J. McQuown, A. Blais, B. D. Dynlacht, G. Chen, and B. Mishra (2007, Feb 1). Functional genomics via multiscale analysis: application to gene expression and chip-on-chip data. *Bioinformatics 23*(3), 314–20.

Stein, E. M. (1993). *Harmonic analysis: real-variable methods, orthogonality, and oscillatory integrals*, Volume 43 of *Princeton Mathematical Series*. Princeton, NJ: Princeton University Press.