Rodeo: Sparse Nonparametric Regression in High Dimensions

John Lafferty[1]
Computer Science Department
and CALD[3]
Carnegie Mellon University

Larry Wasserman[2]
Department of Statistics
and CALD[3]
Carnegie Mellon University

June 16, 2005

Abstract

We present a method for simultaneously performing bandwidth selection and variable selection in nonparametric regression. The method starts with a local linear estimator with large bandwidths, and incrementally decreases the bandwidth in directions where the gradient of the estimator with respect to bandwidth is large. When the unknown function satisfies a sparsity condition, the approach avoids the curse of dimensionality. The method—called *rodeo* (regularization of derivative expectation operator)—conducts a sequence of hypothesis tests, and is easy to implement. A modified version that replaces testing with soft thresholding may be viewed as solving a sequence of lasso problems. When applied in one dimension, the rodeo yields a method for choosing the locally optimal bandwidth.

Keywords: Nonparametric regression, sparsity, local linear smoothing, adaptive estimation, bandwidth estimation, variable selection.

I. Introduction

Estimating a high dimensional regression function is notoriously difficult due to the "curse of dimensionality." Minimax theory precisely characterizes the curse. Let

$$Y_i = m(X_i) + \epsilon_i, \quad i = 1, \ldots, n \tag{1.1}$$

where $X_i = (X_i(1), \ldots, X_i(d)) \in \mathbb{R}^d$ is a d-dimensional covariate, $m : \mathbb{R}^d \to \mathbb{R}$ is the unknown function to estimate, and $\epsilon_i \sim N(0, \sigma^2)$. Then if m is in $W_2(c)$, the d-dimensional Sobolev ball of

[1]Research supported by NSF grants CCR-0122481, IIS-0312814, and IIS-0427206.
[2]Research supported by NIH grants R01-CA54852-07 and MH57881 and NSF grant DMS-0104016.
[3]Center for Automated Learning and Discovery

order two and radius c, it is well known that

$$\liminf_{n\to\infty} n^{4/(4+d)} \inf_{\widehat{m}_n} \sup_{m\in W_2(c)} \mathcal{R}(\widehat{m}_n, m) \;>\; 0\,, \tag{1.2}$$

where $\mathcal{R}(\widehat{m}_n, m) = \mathbb{E}_m \int (\widehat{m}_n(x) - m(x))^2\, dx$ is the risk of the estimate $\widehat{m}_n$ constructed on a sample of size n. Thus, the best rate of convergence is $n^{-4/(4+d)}$, which is impractically slow if d is large.

However, for some applications it is reasonable to expect that the true function only depends on a small number of the total covariates. Suppose that m satisfies such a sparseness condition, so that

$$m(x) = m(x_R) \tag{1.3}$$

where $x_R = (x_j :\; j \in R)$, $R \subset \{1, \ldots, d\}$ is a subset of the d covariates, of size $r = |R| \ll d$. We call $\{x_j\}_{j\in R}$ the *relevant variables*. Under this sparseness assumption we can hope to achieve the better minimax convergence rate of $n^{-4/(4+r)}$ if the r relevant variables can be isolated. Thus, we are faced with the problem of variable selection in nonparametric regression.

A large body of previous work has addressed this fundamental problem, which has led to a variety of methods to combat the curse of dimensionality. Many of these are based on very clever, though often heuristic techniques. For additive models of the form $f(x) = \sum_j f_j(x_j)$, standard methods like stepwise selection, C_p and AIC can be used (Hastie et al., 2001). For spline models, Zhang et al. (2005) use likelihood basis pursuit, essentially the lasso adapted to the spline setting. CART (Breiman et al., 1984) and MARS (Friedman, 1991) effectively perform variable selection as part of their function fitting. Support vector regression can be seen as creating a sparse representation using basis pursuit in a reproducing kernel Hilbert space (Girosi, 1997). There is also a large literature on Bayesian methods, including methods for sparse Gaussian processes (Tipping, 2001; Smola and Bartlett, 2001; Lawrence et al., 2003); see George and McCulloch (1997) for a brief survey. More recently, Li et al. (2005) use independence testing for variable selection and Bühlmann and Yu (2005) introduced a boosting approach. While these methods have met with varying degrees of empirical success, they can be challenging to implement and demanding computationally. Moreover, these methods are typically very difficult to analyze theoretically, and so come with no formal guarantees. Indeed, the theoretical analysis of sparse *parametric* estimators such as the lasso (Tibshirani, 1996) is difficult, and only recently has significant progress been made on this front (Donoho, 2004; Fu and Knight, 2000).

In this paper we present a new approach for sparse nonparametric function estimation that is both computationally simple and amenable to theoretical analysis. We call the general framework *rodeo*, for "regularization of derivative expectation operator." It is based on the idea that bandwidth and variable selection can be simultaneously performed by computing the infinitesimal change in a nonparametric estimator as a function of the smoothing parameters, and then thresholding these derivatives to get a sparse estimate. As a simple version of this principle we use hard thresholding, effectively carrying out a sequence of hypothesis tests. A modified version that replaces testing with soft thresholding may be viewed as solving a sequence of lasso problems. The potential appeal of this approach is that it can be based on relatively simple and theoretically well understood nonparametric techniques such as local linear smoothing, leading to methods that are simple to

implement and can be used in high dimensional problems. Moreover, we show that they can achieve near optimal minimax rates of convergence, and therefore circumvent the curse of dimensionality when the true function is indeed sparse. When applied in one dimension, our method yields a locally optimal bandwidth and is similar to the estimators of Ruppert (1997) and Lepski et al. (1997). We present a series of experiments on synthetic and real data that demonstrate the effectiveness of the approach.

In the following section we outline the basic rodeo approach, which is actually a general strategy that can be applied to a wide range of nonparametric estimators. We then specialize in Section 3 to the case of local linear smoothing, since the asymptotic properties of this smoothing technique are fairly well understood. In particular, we build upon the analysis of Ruppert and Wand (1994) for local linear regression. In Section 4 we present some simple examples of the rodeo, before proceeding to an analysis of its properties in Section 5. Our main theoretical result characterizes the asymptotic running time, selected bandwidths, and risk of the algorithm, where we allow the data dimension to grow with the number of examples. Finally, in Section 6 we present further examples and discuss several extensions of the basic version of the rodeo considered in the earlier sections.

II. Rodeo: The Main Idea

The key idea in our approach is as follows. Fix a point x and let $\widehat{m}_h(x)$ denote an estimator of $m(x)$ based on a vector of smoothing parameters $h = (h_1, \ldots, h_d)$. If c is a scalar, then we write $h = c$ to mean $h = (c, \ldots, c)$.

Let $M(h) = \mathbb{E}(\widehat{m}_h(x))$ denote the mean of $\widehat{m}_h(x)$. For now, assume that x_i is one of the observed data points and that $\widehat{m}_0(x) = Y_i$. In that case, $m(x) = M(0) = \mathbb{E}(Y_i)$. If $P = (h(t) : 0 \leq t \leq 1)$ is a smooth path through the set of smoothing parameters with $h(0) = 0$ and $h(1) = 1$ (or any other fixed, large bandwidth) then

$$\begin{aligned} m(x) &= M(0) = M(1) + M(0) - M(1) \\ &= M(1) - \int_0^1 \frac{dM(h(s))}{ds}\, ds \\ &= M(1) - \int_0^1 \left\langle D(s), \dot{h}(s) \right\rangle ds \end{aligned} \tag{2.1}$$

where

$$D(h) = \nabla M(h) = \left(\frac{\partial M}{\partial h_j}, \ldots, \frac{\partial M}{\partial h_j} \right)^T \tag{2.2}$$

is the gradient of $M(h)$ and $\dot{h}(s) = \frac{dh(s)}{ds}$ is the derivative of $h(s)$ along the path. A biased, low variance estimator of $M(1)$ is $\widehat{m}_1(x)$. An unbiased estimator of $D(h)$ is

$$Z(h) = \left(\frac{\partial \widehat{m}_h(x)}{\partial h_1}, \ldots, \frac{\partial \widehat{m}_h(x)}{\partial h_d} \right)^T. \tag{2.3}$$

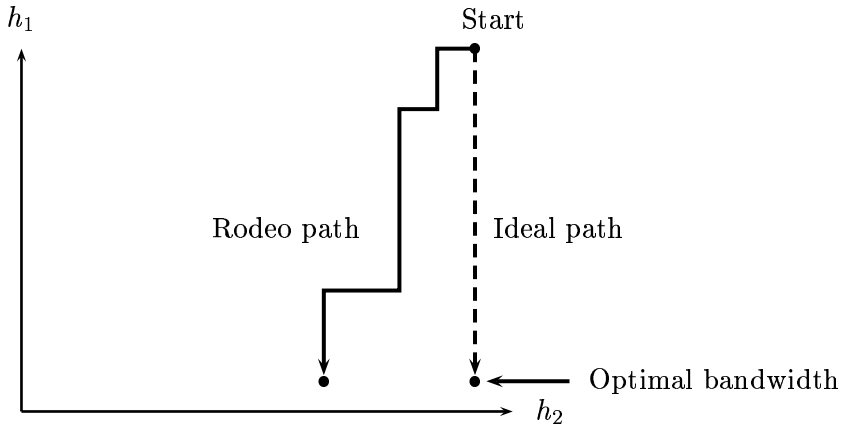

Figure 1: Conceptual illustration: The bandwidths for the relevant variables (h_1) are shrunk, while the bandwidths for the irrelevant variables (h_2) are kept relatively large.

The naive estimator

$$\widehat{m}(x) = \widehat{m}_1(x) - \int_0^1 \big\langle Z(s), \dot{h}(s) \big\rangle ds \tag{2.4}$$

is identically equal to $\widehat{m}_0(x) = Y_i$, which has poor risk since the variance of $Z(h)$ is large for small h. However, our sparsity assumption on m suggests that there should be paths for which $D(h)$ is also sparse. Along such a path, we replace $Z(h)$ with an estimator $\widehat{D}(h)$ that makes use of the sparsity assumption. Our estimate of $m(x)$ is then

$$\widetilde{m}(x) = \widehat{m}_1(x) - \int_0^1 \big\langle \widehat{D}(s), \dot{h}(s) \big\rangle ds\,. \tag{2.5}$$

To implement this idea we need to do two things: (i) we need to find a sparse path and (ii) we need to take advantage of this sparseness when estimating D along that path.

The key observation is that if x_j is irrelevant, then we expect that changing the bandwidth h_j for that variable should cause only a small change in the estimator $\widehat{m}_h(x)$. Conversely, if x_j is relevant, then we expect that changing the bandwidth h_j for that variable should cause a large change in the estimator. Thus, $Z_j = \partial \widehat{m}_h(x)/\partial h_j$ should discriminate between relevant and irrelevant covariates. To simplify the procedure, we can replace the continuum of bandwidths with a discrete set where each $h_j \in \mathcal{B} = \{h_0, \beta h_0, \beta^2 h_0, \ldots\}$ for some $0 < \beta < 1$. Moreover, we can proceed in a greedy fashion by estimating $D(h)$ sequentially with $h_j \in \mathcal{B}$ and setting $\widehat{D}_j(h) = 0$ when $h_j < \widehat{h}_j$, where $\widehat{h}_j$ is the first h such that $|Z_j(h)| < \lambda_j(h)$ for some threshold λ_j. This greedy version, coupled with the hard threshold estimator, yields $\widetilde{m}(x) = \widehat{m}_{\widehat{h}}(x)$. A conceptual illustration of the idea is shown in Figure 1.

To further elucidate the idea, consider now the one-dimensional case $x \in \mathbb{R}$, so that

$$m(x) = M(1) - \int_0^1 \frac{dM(h)}{dh} dh = M(1) - \int_0^1 D(h)\, dh. \tag{2.6}$$

Suppose that $\widehat{m}_h(x) = \sum_{i=1}^n Y_i \ell_i(x,h)$ is a linear estimator, where the weights $\ell_i(x,h)$ depend on a bandwidth h. In this case

$$Z(h) = \sum_{i=1}^n Y_i \ell_i'(x,h) \tag{2.7}$$

where the prime denotes differentiation with respect to h. Then we set

$$\widetilde{m}(x) = \widehat{m}_1(x) - \int_0^1 \widehat{D}(h)\, dh \tag{2.8}$$

where $\widehat{D}(h)$ is an estimator of $D(h)$. Now,

$$Z(h) \approx N(b(h), s^2(h)) \tag{2.9}$$

where, for typical smoothers, $b(h) \approx Ah$ and $s^2(h) \approx C/nh^3$ for some constants A and C. Take the hard threshold estimator

$$\widehat{D}(h) = Z(h) I\big(|Z(h)| > \lambda(h)\big) \tag{2.10}$$

where $\lambda(h)$ is chosen to be slightly larger than $s(h)$. An alternative is the soft-threshold estimator

$$\widehat{D}(h) = \text{sign}(Z(h))(|Z(h)| - \lambda(h))_+. \tag{2.11}$$

The greedy algorithm, coupled with the hard threshold estimator, yields a bandwidth selection procedure based on testing. This approach to bandwidth selection is very similar to that of Lepski et al. (1997), who take

$$\widehat{h} = \max\{h \in \mathcal{H} :\ \phi(h,\eta) = 0 \text{ for all } \eta < h\} \tag{2.12}$$

where $\phi(h,\eta)$ is a test for whether $\widehat{m}_\eta$ improves on $\widehat{m}_h$. This more refined test leads to estimators that achieve good spatial adaptation over large function classes. Our approach is also similar to a method of Ruppert (1997) that uses a sequence of decreasing bandwidths and then estimates the optimal bandwidth by estimating the mean squared error as a function of bandwidth. Our greedy approach only tests whether an infinitesimal change in the bandwidth from its current setting leads to a significant change in the estimate, and is more easily extended to a practical method in higher dimensions.

III. The Multivariate Rodeo Using Local Linear Regression

Now we present the multivariate rodeo in detail. We use local linear smoothing as the basic method since it is known to have many good properties. Let $x = (x(1), \ldots, x(d))$ be some target point at which we want to estimate m. Let $\widehat{m}_H(x)$ denote the local linear estimator of $m(x)$ using bandwidth matrix H. Thus,

$$\widehat{m}_H(x) = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x Y \equiv S_x Y \tag{3.1}$$

where $e_1 = (1, 0, \ldots, 0)^T$,

$$X_x = \begin{pmatrix} 1 & (X_1 - x)^T \\ \vdots & \vdots \\ 1 & (X_n - x)^T \end{pmatrix}, \tag{3.2}$$

W_x is diagonal with (i,i) element $K_H(X_i - x)$ and $K_H(u) = |H|^{-1/2}K(H^{-1/2}u)$. The estimator $\widehat{m}_H$ can be written as

$$\widehat{m}_H(x) = \sum_{i=1}^{n} G(X_i, x, h)\, Y_i \tag{3.3}$$

where

$$G(u, x, h) = e_1^T (X_x^T W_x X_x)^{-1} \begin{pmatrix} 1 \\ (u-x)^T \end{pmatrix} K_H(u - x) \tag{3.4}$$

is called the *effective kernel*. One can regard local linear regression as a refinement of kernel regression where the effective kernel G adjusts for boundary bias and design bias; see Fan (1992), Hastie and Loader (1993) and Ruppert and Wand (1994).

We assume that the covariates are random with density $f(x)$ and that x is interior to the support of f. We make the same assumptions as Ruppert and Wand (1994) in their analysis of the bias and variance of local linear regression. In particular:

(i) The kernel K has compact support with zero odd moments and there exists $\nu_2 = \nu_2(K) \neq 0$ such that

$$\int uu^\top K(u)\, du = \nu_2(K) I \tag{3.5}$$

where I is the $d \times d$ identity matrix.

(ii) The sampling density $f(x)$ is continuously differentiable and strictly positive.

In the version of the algorithm that follows, we take K to be a product kernel and H to be diagonal with elements $h = (h_1, \ldots, h_d)$ and we write $\widehat{m}_h$ instead of $\widehat{m}_H$.

Our method is based on the statistic

$$Z_j = \frac{\partial \widehat{m}_h(x)}{\partial h_j} = \sum_{i=1}^{n} G_j(X_i, x, h) Y_i \tag{3.6}$$

where

$$G_j(u, x, h) = \frac{\partial G(u, x, h)}{\partial h_j}. \tag{3.7}$$

Let

$$\mu_j \equiv \mu_j(h) = \mathbb{E}(Z_j | X_1, \ldots, X_n) = \sum_{i=1}^{n} G_j(X_i, x, h) m(X_i) \tag{3.8}$$

and

$$s_j^2 \equiv s_j^2(h) = \mathbb{V}(Z_j | X_1, \ldots, X_n) = \sigma^2 \sum_{i=1}^{n} G_j(X_i, x, h)^2. \tag{3.9}$$

In Section 6.A we explain how to estimate σ; for now, assume that σ is known. The hard thresholding version of the rodeo algorithm is described in Figure 2. We make use of a sequence c_n satisfying $dc_n = \Omega(\log n)$, where we write $f(n) = \Omega(g(n))$ if $f(n) \geq cg(n) > 0$ for n sufficiently large.

Rodeo: Hard thresholding version

1. *Select* parameter $0 < \beta < 1$ and initial bandwidth h_0, satisfying $1 \leq h_0 \leq \log^{\ell/d} n$, for a fixed constant ℓ. Let c_n be a sequence satisfying

$$dc_n = \Omega(\log n) \tag{3.10}$$

2. *Initialize* the bandwidths, and activate all covariates:
   (a) $h_j = h_0,\ j = 1, 2, \ldots, d.$
   (b) $\mathcal{A} = \{1, 2, \ldots, d\}$
3. *While* $\mathcal{A}$ *is nonempty*, do for each $j \in \mathcal{A}$:
   (a) Compute the estimated derivative expectation: Z_j (equation 3.6) and s_j (equation 3.9).
   (b) Compute the threshold $\lambda_j = s_j\sqrt{2\log(dc_n)}$.
   (c) If $|Z_j| > \lambda_j$, then set $h_j \leftarrow \beta h_j$; otherwise remove j from $\mathcal{A}$.
4. *Output* bandwidths $h^\star = (h_1, \ldots, h_d)$ and estimator $\widetilde{m}(x) = \widehat{m}_{h^\star}(x)$.

Figure 2: The hard thresholding version of the rodeo, which can be applied using the derivatives Z_j of any nonparametric smoother.

To derive an explicit expression for Z_j, equivalently G_j, we use

$$\frac{\partial A^{-1}}{\partial h} = -A^{-1}\frac{\partial A}{\partial h}A^{-1} \tag{3.11}$$

to get that

$$\begin{aligned} Z_j &= \frac{\partial \widehat{m}_h(x)}{\partial h_j} && (3.12)\\ &= e_1^\top (X^\top W X)^{-1} X^\top \frac{\partial W}{\partial h_j} Y - e_1^\top (X^\top W X)^{-1} X^\top \frac{\partial W}{\partial h_j} X (X^\top W X)^{-1} X^\top W Y && (3.13)\\ &= e_1^\top (X^\top W X)^{-1} X^\top \frac{\partial W}{\partial h_j} (Y - X\widehat{\alpha}) && (3.14) \end{aligned}$$

where $\widehat{\alpha} = (X^\top W X)^{-1} X^\top W Y$ is the coefficient vector for the local linear fit (and we have dropped the dependence on the local point x in the notation).

Note that the factor $|H|^{-1} = \prod_{i=1}^d 1/h_i$ in the kernel cancels in the expression for $\widehat{m}$, and therefore

we can ignore it in our calculation of Z_j. Assuming a product kernel we have

$$W = \text{diag}\left(\prod_{j=1}^{d} K((X_{1j}-x_j)/h_j), \ldots, \prod_{j=1}^{d} K((X_{nj}-x_j)/h_j)\right) \tag{3.15}$$

and $\partial W/\partial h_j = WL_j$ where

$$L_j = \text{diag}\left(\frac{\partial \log K((X_{1j}-x_j)/h_j)}{\partial h_j}, \ldots, \frac{\partial \log K((X_{nj}-x_j)/h_j)}{\partial h_j}\right) \tag{3.16}$$

and thus

$$\begin{aligned} Z_j &= e_1^\top (X^\top W X)^{-1} X^\top W L_j (Y - X\widehat{\alpha}) && (3.17)\\ &= e_1^\top B L_j (I - XB) Y && (3.18)\\ &= G_j(x,h)^\top Y && (3.19) \end{aligned}$$

where $B = (X^\top W X)^{-1} X^\top W$.

For example, with the Gaussian kernel $K(u) = \exp(-u^2/2)$ we have

$$L_j = \frac{1}{h_j^3}\,\text{diag}\left((X_{1j}-x_j)^2, \ldots, (X_{nj}-x_j)^2\right) \tag{3.20}$$

and for the Epanechnikov kernel $K(u) = (5-x^2)\,\mathbb{I}(|x| \le \sqrt{5})$ we have

$$\begin{aligned} L_j = \frac{1}{h_j^3}\,\text{diag}\Bigg(&\frac{2(X_{1j}-x_j)^2}{5-(X_{1j}-x_j)^2/h_j^2}\,\mathbb{I}(|X_{1j}-x_j| \le \sqrt{5}h_j), \ldots, && (3.21)\\ &\frac{2(X_{nj}-x_j)^2}{5-(X_{nj}-x_j)^2/h_j^2}\,\mathbb{I}(|X_{1j}-x_j| \le \sqrt{5}h_j)\Bigg) && (3.22) \end{aligned}$$

The calculations for other kernels are similar.

IV. Examples

In this section we illustrate the rodeo on some examples. We return to the examples later when we discuss estimating σ, as well as a global (non-local) version of the rodeo.

A. Two Relevant Variables

In the first example, we take $m(x) = 5x_1^2x_2^2$ with $d = 10$, $\sigma = .5$ with $x_i \sim \text{Uniform}(0,1)$. The algorithm is applied to the local linear estimates around the test point $x_0 = (\frac{1}{2}, \ldots, \frac{1}{2})$, with $\beta = 0.8$. Figure 3 shows the bandwidths averaged over 100 runs of the rodeo, on data sets of size $n = 500$. The second example shows the algorithm applied to the function $m(x) = 2(x_1+1)^3 + 2\sin(10x_2)$, in this case in $d = 20$ dimensions with $\sigma = 1$.

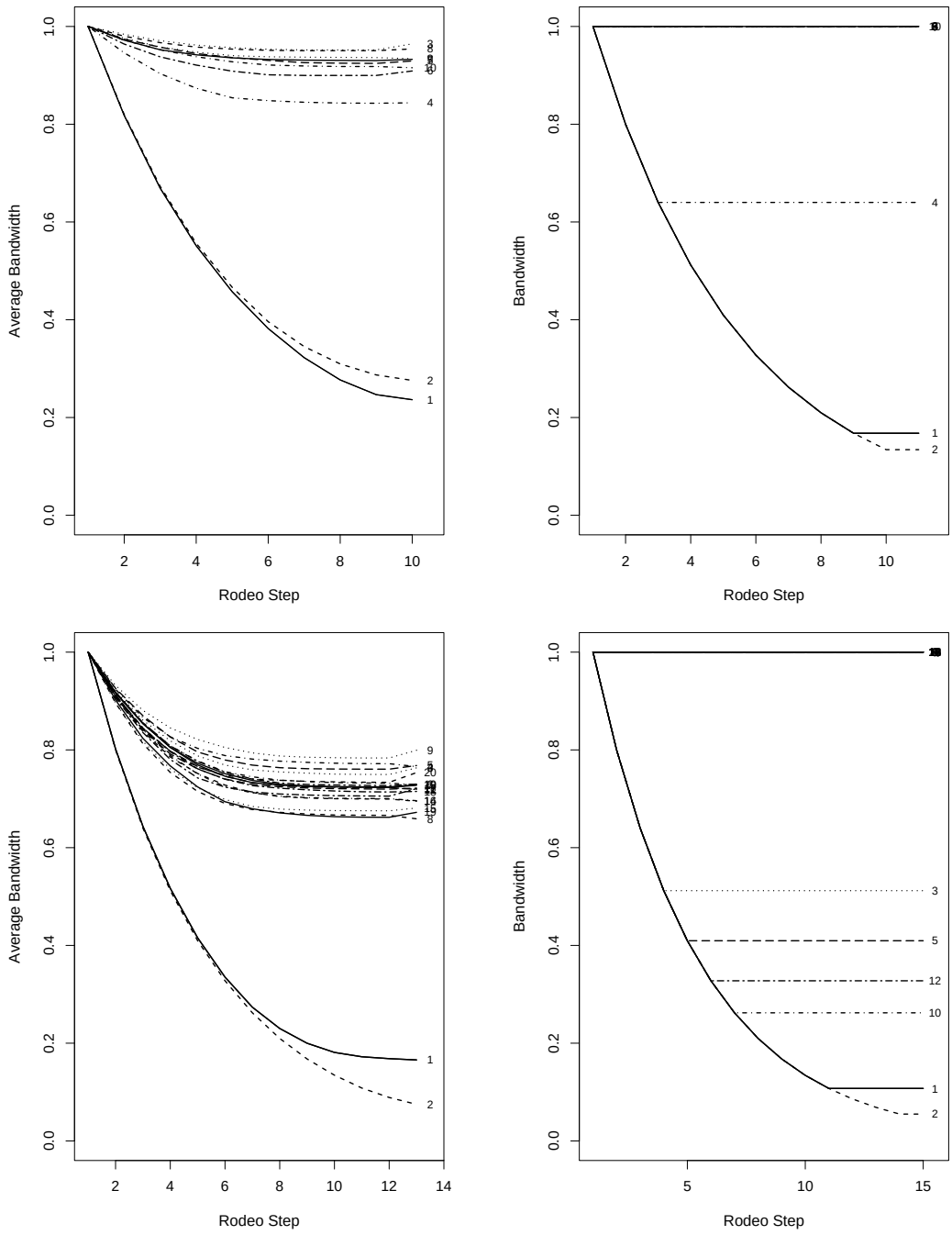

Figure 3: Rodeo run on two synthetic data sets of size $n = 500$, showing average bandwidths over 100 runs (left) and bandwidths on a single run of the algorithm (right). In the top plots $m(x) = 5x_1^2x_2^2$ with $d = 10$ and $\sigma = .5$; in the bottom plots $m(x) = 2(x_1 + 1)^3 + 2\sin(10x_2)$, $d = 20$ and $\sigma = 1$. The figures show that the bandwidths for the relevant variables x_1 and x_2 are shrunk, while the bandwidths for the irrelevant variables remain large.

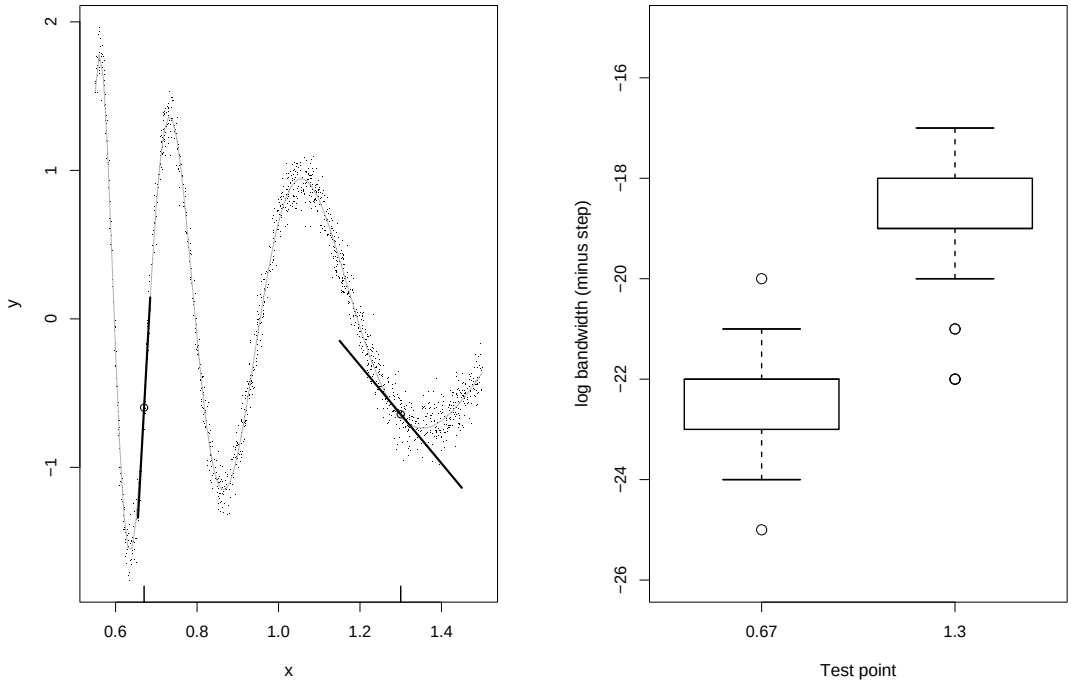

Figure 4: A one-dimensional example. Here the underlying function is $m(x) = (1/x)\sin(15/x)$, and $n = 1500$ data points are sampled, $x \sim \text{Uniform}(0,1) + \frac{1}{2}$. The left plot shows the local linear fit at two test points; the right plot shows the final log bandwidth, $\log_{1/\beta} h_\star$, (equivalently, minus the number of steps) of the rodeo over 50 randomly generated data sets.

The plots demonstrate how the bandwidths h_1 and h_2 of the relevant variables are shrunk, while the bandwidths of the relevant variables tend to remain large.

B. A One-Dimensional Example

The next figure illustrates the algorithm in one dimension. The underlying function in this case is $m(x) = (1/x)\sin(15/x)$, and $n = 1500$ data points are sampled as $x \sim \text{Uniform}(0,1) + \frac{1}{2}$. The algorithm is run at two test points; the function is more rapidly varying near the test point $x = 0.67$ than near the test point $x = 1.3$, and the rodeo appropriately selects a smaller bandwidth at $x = 0.67$. The right plot of Figure 4 displays boxplots for logarithm of the final bandwidth, in the base $1/\beta$ (equivalently, minus the number of steps in the algorithm), averaged over 50 randomly generated data sets.

The figure illustrates how smaller bandwidths are selected where the function is more rapidly varying. Indeed, as we show in the next section, in one dimension the algorithm selects the locally optimal bandwidth with high probability.

V. Properties of the Rodeo

Now we give some results on the properties of the resulting estimator. Formally, we use a triangular array approach so that $m(x)$, $f(x)$, d and r can all change as n changes. For convenience of notation we assume that the covariates are numbered such that the relevant variables x_j correspond to $1 \leq j \leq r$ and the irrelevant variables x_j correspond to $r+1 \leq j \leq d$. We write $Y_n = \widetilde{O}_P(a_n)$ to mean that $Y_n = O_p(b_n a_n)$ where b_n is logarithmic in n. We assume throughout that m has continuous third order derivatives in a neighborhood of x.

To begin, we have the following technical lemmas on the mean and variance of Z_j.

Lemma 5.1. *Suppose that x is interior to the support of f. Suppose that K is a product kernel with bandwidth matrix $H = diag(h_1^2, \ldots, h_d^2)$. If f is uniform then*

$$\mu_j = 0 \text{ for all } j \in R^c. \tag{5.1}$$

More generally, assuming that r is bounded, we have the following when $h_j \to 0$: If $j \in R^c$ the derivative of the bias is

$$\mu_j = \frac{\partial}{\partial h_j}\mathbb{E}[\widehat{m}_H(x) - m(x)] \;\; = \;\; -tr\left(H_R \mathcal{H}_R\right) \nu_2^2 \left(\nabla_j \log f(x)\right)^2 h_j + o_P(h_j) \tag{5.2}$$

where the Hessian of $m(x)$ is $\mathcal{H} = \begin{pmatrix} \mathcal{H}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ and $H_R = diag(h_1^2, \ldots, h_r^2)$. For $j \in R$ we have

$$\mu_j = \frac{\partial}{\partial h_j}\mathbb{E}[\widehat{m}_H(x) - m(x)] \;\; = \;\; h_j \nu_2 m_{jj}(x) + o_P(h_j). \tag{5.3}$$

Remark 5.2. Special treatment is needed if x is a boundary point; see Theorem 2.2 of Ruppert and Wand (1994).

Proof. We follow the setup of Ruppert and Wand (1994) except for one difference: the irrelevant variables have different leading terms in the expansions than relevant variables.

Let D_m be the gradient of m at x, and let

$$Q = ((X_1 - x)^T \mathcal{H}(X_1 - x), \ldots, (X_n - x)^T \mathcal{H}(X_n - x))^T. \tag{5.4}$$

Note that D_m and Q are only functions of the relevant variables. Then

$$m(X_i) = m(x) + (X_i - x)^T D_m + \frac{1}{2} Q_i + \xi_i \tag{5.5}$$

where ξ_i is the third order remainder term and so, with $M = (m(X_1), \ldots, m(X_n))^T$,

$$M = X_x \begin{pmatrix} m(x) \\ D_m \end{pmatrix} + \frac{1}{2} Q + \xi \tag{5.6}$$

where $\xi = (\xi_1, \ldots, \xi_n)^T$. Since $S_x X_x (m(x), D_m)^T = m(x)$, the bias $b(x) = \mathbb{E}(\widehat{m}_H(x)) - m(x)$ is given by

$$\begin{aligned} b(x) &= S_x M - M = \frac{1}{2} S_x Q + S_x \xi = \frac{1}{2} S_x Q + o_P(\mathrm{tr}(H_R)) && (5.7) \\ &= \frac{1}{2} (X_x^T W_x X_x)^{-1} X_x^T W_x Q + o_P(\mathrm{tr}(H_R)). && (5.8) \end{aligned}$$

From the calculations in Ruppert and Wand we have

$$\frac{1}{n}(X_x^T W_x X_x) = \begin{pmatrix} f(x) + o_P(1) & \nu_2 D^\top H^\top + o_P(\mathbf{1}^\top H) \\ \nu_2 H D + o_P(H\mathbf{1}) & \nu_2 f(x) H + o_P(H) \end{pmatrix}. \quad (5.9)$$

where D is the gradient of f.

We can write this as $(A + vv^\top)$ where

$$A = \begin{pmatrix} f(x) - 1 & 0 \\ 0 & \nu_2 f(x) H \end{pmatrix} \quad (5.10)$$

and

$$v^\top = (1 + o_P(1),\ \nu_2 D^\top H^\top + o_P(\mathbf{1}^\top H^\top)) \quad (5.11)$$

since then

$$vv^\top = \begin{pmatrix} 1 + o_P(1) & \nu_2 D^\top H^\top + o_P(\mathbf{1}^\top H^\top) \\ \nu_2 H D + o_P(H\mathbf{1}) & o_P(H) \end{pmatrix} \quad (5.12)$$

We now apply the matrix inversion lemma (Woodbury formula)

$$(A + vv^\top)^{-1} = A^{-1} - A^{-1} v (1 + v^\top A^{-1} v)^{-1} v^\top A^{-1} \quad (5.13)$$

Note that

$$A^{-1} = \begin{pmatrix} \frac{1}{f(x)-1} & 0 \\ 0 & \frac{H^{-1}}{\nu_2 f(x)} \end{pmatrix} \quad (5.14)$$

and

$$1 + v^\top A^{-1} v = \frac{f(x)}{f(x) - 1} + \frac{\nu_2 D^\top H D}{f(x)} + o_P(\mathrm{tr}(H)) \quad (5.15)$$

Also,

$$A^{-1} v v^\top A^{-1} = \begin{pmatrix} \frac{1}{(f(x)-1)^2} + o_P(1) & \frac{D^\top}{f(x)(f(x)-1)} + o_P(1) \\ \frac{D}{f(x)(f(x)-1)} + o_P(1) & \frac{H^{-1}}{\nu_2 f(x)} o_P(1) \end{pmatrix} \quad (5.16)$$

We thus get that

$$\begin{aligned} &\frac{1}{n}(X_x^T W_x X_x)^{-1} = && (5.17) \\ &= \begin{pmatrix} \frac{1}{f(x)-1} & 0 \\ 0 & \frac{H^{-1}}{\nu_2 f(x)} \end{pmatrix} - \left(\frac{f(x) - 1}{f(x)} + o_P(1) \right) \begin{pmatrix} \frac{1}{(f(x)-1)^2} + o_P(1) & \frac{D^\top}{f(x)(f(x)-1)} + o_P(1) \\ \frac{D}{f(x)(f(x)-1)} + o_P(1) & \frac{H^{-1}}{\nu_2 f(x)} o_P(1) \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{f(x)} + o_P(1) & -\frac{D^\top}{f(x)^2} + o_P(1) \\ -\frac{D}{f(x)^2} + o_P(1) & \frac{H^{-1}}{\nu_2 f(x)}(1 + o_P(1)). \end{pmatrix} && (5.18) \end{aligned}$$

Now

$$\frac{1}{n}X_x^T W_x Q = \begin{pmatrix} \frac{1}{2}\nu_2 \mathrm{tr}(H\mathcal{H}) + o_P(\mathrm{tr}(H)) \\ \int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u) f(x + H^{1/2}(u))du + o_P(H^{3/2}\mathbf{1}) \end{pmatrix}, \tag{5.19}$$

and

$$\begin{aligned} &\int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u) f(x + H^{1/2}(u))du \\ &\quad = f(x)\int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u)du \\ &\qquad + \int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u)(D^T H^{1/2}u)du + o_P(H^{5/2}\mathbf{1}) \qquad (5.20) \\ &\quad = \int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u)(D^T H^{1/2}u)du + o_P(H^{5/2}\mathbf{1}). \qquad (5.21) \end{aligned}$$

So,

$$\frac{1}{n}X_x^T W_x Q = \begin{pmatrix} \frac{1}{2}\nu_2 \mathrm{tr}(H\mathcal{H}) + o_P(\mathrm{tr}(H)) \\ \int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u)(D^T H^{1/2}u)du + o_P(H^{3/2}) \end{pmatrix}. \tag{5.22}$$

Hence,

$$\begin{aligned} b_1 &= \frac{1}{2}\left(\frac{1}{f(x)} + o_P(1), -\frac{D^T}{f^2(x)} + o_P(1)\right) \times \\ &\quad \times \begin{pmatrix} \frac{1}{2}\nu_2 \mathrm{tr}(H\mathcal{H}) + o_P(\mathrm{tr}(H)) \\ \int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(H^{1/2}u)(D^T H^{1/2}u)du + o_P(H^{3/2}) \end{pmatrix} \qquad (5.23) \\ &= \frac{\nu_2 \mathrm{tr}(H\mathcal{H})}{4f(x)} - \frac{1}{2f^2(x)}\int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}(D^T H^{1/2}u)^2 du + o_P(\mathrm{tr}(H)). \end{aligned} \tag{5.24}$$

Now we use the fact that K is a product kernel, with bandwidth matrix $H = \mathrm{diag}(h_1^2, \ldots, h_d^2)$, and define

$$v_R = \begin{pmatrix} u_1 h_1 \\ \vdots \\ u_r h_r \end{pmatrix}. \tag{5.25}$$

Then

$$\int K(u)\left\{(H^{1/2}u)^T \mathcal{H}(H^{1/2}u)\right\}\left(D^T H^{1/2}u\right)^2 du = \int K(u)(v_R^T R v_R)\left(\sum_{i=1}^d D_i u_i h_i\right)^2 du \tag{5.26}$$

and for $j \in R^c$,

$$\begin{aligned}
\frac{\partial b_1}{\partial h_j} &= -\frac{1}{f(x)^2} \int K(u)(v_R^T \mathcal{H}_R v_R) \Big(\sum_{i=1}^{d} D_i u_i h_i \Big) D_j u_j \, du + o_P(h_j) && (5.27) \\
&= -\frac{1}{f(x)^2} \int K(u)(v_R^T \mathcal{H}_R v_R) \left(D_j^2 u_j^2 \right) h_j \, du + o_P(h_j) && (5.28) \\
&= -\frac{D_j^2 h_j}{f(x)^2} \int_{u_R} K(u_R)(v_R^T \mathcal{H}_R v_R) \, du_R \int_{u_j} K(u_j) u_j^2 \, du_j + o_P(h_j) && (5.29) \\
&= -\mathrm{tr}\left(H_R \mathcal{H}_R\right) \, \nu_2^2 \, \left(\nabla_j \log f(x)\right)^2 h_j + o_P(h_j) && (5.30)
\end{aligned}$$

where the second equality follows from the fact that we are using a product kernel and the terms in the sum $(\sum_{i \neq j} D_i u_i h_i)$ result in integrands of odd order in u_i when $i \neq j$. Similarly, the last equality follows from assumption (i) since

$$\int K(u) u_i u_j \mathcal{H}_{ij} \, du \quad = \quad \delta_{ij} \, \nu_2 \, \mathcal{H}_{ij} \, . \tag{5.31}$$

The last statement follows from the results in Ruppert and Wand. □

By similar calculations we get:

Lemma 5.3. *Let*

$$C = \left(\frac{\sigma^2 R(K)}{4 f(x)} \right) \tag{5.32}$$

where $R(K) = \int K(u)^2 \, du$. *Then, if* $h_j = o(1)$,

$$s_j^2 = \mathbb{V}\mathrm{ar}(Z_j | X_1, \ldots, X_n) = \frac{C}{n h_j^2} \left(\prod_{k=1}^{d} \frac{1}{h_k} \right) \Big(1 + o_P(1) \Big). \tag{5.33}$$

Our main theoretical result characterizes the asymptotic running time, selected bandwidths, and risk of the algorithm. An important aspect of our analysis is that we allow the data dimension d to grow with number of examples n, which we view as important since modern data sets are often high dimensional, even with $d = O(n)$. However, in order to allow the dimension d to grow with n, we need to make assumptions on the functions m and f.

(A1) For each $j > r$, there is a constant k_j such that

$$\mu_j^2(h) = o_P(\tfrac{1}{d}) \;\text{ if }\; h_j \leq \tfrac{1}{\log^{k_j/d} n} \tag{5.34}$$

(A2) For each $j \leq r$, $|m_{jj}(x)| > 0$ and there exists $c > 0$ such that

$$|\mu_j(h)| \geq c h_j |m_{jj}(x)|. \tag{5.35}$$

Explanation of the Assumptions. To give the intuition behind these assumptions, recall from Lemma 5.1 that

$$\mu_j = \begin{cases} A_j h_j + o_P(h_j) & j \leq r \\ B_j h_j + o_P(h_j) & j > r \end{cases} \tag{5.36}$$

where

$$A_j = \nu_2 m_{jj}(x), \qquad B_j = -\text{tr}(H\mathcal{H})\nu_2^2 (\nabla_j \log f(x))^2. \tag{5.37}$$

Moreover, $\mu_j = 0$ when the sampling density f is uniform or the data are on a regular grid. Consider Assumption (A1). If f is uniform then this assumption is automatically satisfied since then $\mu_j(s) = 0$ for $j > r$. More generally, μ_j is approximately proportional to $(\nabla_j \log f(x))^2$ for $j > r$ which implies that $|\mu_j| \approx 0$ for irrelevant variables if f is sufficiently smooth in the variable x_j. Hence, Assumption (A1) can be interpreted as requiring that f is sufficiently smooth in the irrelevant dimensions. If d is constant and f is smooth, then this assumption is *implied* by Lemma 5.1. However, we allow for the possibility that d grows with n. Assumption (A1) in this case says that μ_j must be small even for bandwidths close to one. Such an assumption is required in order to control the variance of the estimator, which will otherwise grow exponentially in d.

Now consider Assumption (A2). Equation (5.36) ensures that μ_j is proportional to $h_j|m_{jj}(x)|$ for small h_j. Assumption (A2) ensures that this is true even if h_j is not small. This assumption can be eliminated by starting the algorithm at a bandwidth that decreases slowly with n in which case (A2) is *implied* by Lemma 5.1. But we prefer to start at larger h_j and keep the assumption.

Theorem 5.4. *Suppose that $d = O(n)$, that $r = O(1)$, and that assumptions (A1) and (A2) hold. Then T_n, the number of iterations until the rodeo stops, satisfies*

$$\mathbb{P}\left(\frac{1}{4+r}\log_{1/\beta}\left(\frac{A_{min}^2 c^2 n}{4C\log(c_n d)\log^k n}\right) \leq T_n \leq \frac{1}{4+r}\log_{1/\beta}\left(\frac{A_{max}^2 n \log^{\ell+1} n}{2C}\right)\right) \to 1 \tag{5.38}$$

where $A_{min} = \min_{j\leq r}|m_{jj}(x)|$, $A_{max} = \max_{j\leq r}|m_{jj}(x)|$, and $k = \max_{j>r} k_j$. Moreover, the algorithm outputs bandwidths $h^\star$ that satisfy

$$\mathbb{P}\left(h_j^\star \geq \frac{1}{\log^{k/d} n} \text{ for all } j > r\right) \to 1 \tag{5.39}$$

$$\mathbb{P}\left(n^{-\frac{1}{r+4}}\left(\frac{2C}{A_{max}^2\log^{\ell+1} n}\right)^{\frac{1}{r+4}} \leq h_j^\star \leq n^{-\frac{1}{r+4}}\left(\frac{4C\log^{k+5\ell} n \log(c_n d)}{c^2 A_{min}^2}\right)^{\frac{1}{r+4}}, \text{ all } j \leq r\right) \to 1\,. \tag{5.40}$$

Corollary 5.5. *Suppose that $A_{max} = O(\log^{\alpha_{max}} n)$ and $A_{min} = \Omega(\log^{\alpha_{min}} n)$ for constants α_{min} and α_{max}. Then under the conditions of Theorem 5.4, the risk $\mathcal{R}(h^\star)$ of the rodeo estimator satisfies*

$$\mathcal{R}(h^\star) = \widetilde{O}_P\left(n^{-4/(4+r)}\right). \tag{5.41}$$

Proof of Corollary 5.5. We have that the squared (conditional) bias is given by

$$\begin{aligned}
\text{Bias}^2(\widehat{m}_{h^\star})) &= \left(\sum_{j\le r} A_j h_j^2\right)^2 + o_P(\text{tr}(H^\top H)) && (5.42)\\
&= \sum_{i,j\le r} A_i A_j h_i^2 h_j^2 + o_P(\text{tr}(H^\top H)) && (5.43)\\
&= \widetilde{O}_P(n^{-4/(4+r)}) && (5.44)
\end{aligned}$$

by Theorem 5.4. Similarly, from Ruppert and Wand (1994) and Theorem 5.4 the (conditional) variance is

$$\begin{aligned}
\mathbb{V}\text{ar}(\widehat{m}_{h^\star}) &= \frac{1}{n}\prod_i \frac{1}{h_i}\frac{R(K)}{f(x)}\sigma(1+o_P(1)) && (5.45)\\
&= \widetilde{O}_P(n^{-1+r/(r+4)}) && (5.46)\\
&= \widetilde{O}_P(n^{-4/(4+r)}) && (5.47)
\end{aligned}$$

where $R(K) = \int K(u)^2\,du$. The result follows from the bias-variance decomposition. □

To prove the theorem we will make use of a version of Mill's inequality, modified for non-zero mean random variables as in Donoho and Johnstone (1994).

Lemma 5.6. *Let $Z \sim N(\theta, 1)$. Then*

$$\mathbb{P}_\theta(|Z| > t) \le \frac{1}{t}e^{-t^2/2} + \frac{\theta^2}{4}. \tag{5.48}$$

Proof. Let

$$g(\theta, t) = \mathbb{P}_\theta(|Z| > t) = \Phi(-t-\theta) + 1 - \Phi(t-\theta). \tag{5.49}$$

Hence, $g'(0) = 0$ and

$$\left|\frac{\partial^2 g(\theta,t)}{\partial\theta^2}\right| = \left|(t-\theta)\phi(t-\theta) - (t+\theta)\phi(t+\theta)\right| \tag{5.50}$$

so that

$$\sup_{t,\theta}|g''| \le 2\sup_u |u\phi(u)| \le \frac{1}{2}. \tag{5.51}$$

Also,

$$g(0) = \mathbb{P}_0(|Z| > t) \le \frac{2\phi(t)}{t} = \sqrt{\frac{2}{\pi}}\frac{1}{t}e^{-t^2/2} \le \frac{1}{t}e^{-t^2/2}. \tag{5.52}$$

Finally,

$$g(\theta, t) \le g(0) + \frac{\theta^2 \sup|g''|}{2} \le \frac{1}{t}e^{-t^2/2} + \frac{\theta^2}{4}. \tag{5.53}$$

which gives the statement of the lemma. □

Proof of Theorem 5.4. In what follows, the calculations are understood as being conditional on $X_1, \ldots, X_n$. When $h_j = o(1)$, we ignore the $o_P(1)$ terms in the asymptotic expressions for s_j and μ_j, it being understood that these hold, except on a set of probability tending to 0.

First consider $j > r$. Let $V = \{j > r : \ h_j \leq (\log n)^{-k_j/d}\}$. Then,

$$\begin{aligned}
\mathbb{P}(|Z_j| > \lambda_j, \text{ for some } j \in V) &\leq \sum_{j \in V} \mathbb{P}(|Z_j| > \lambda_j) = \sum_{j \in V} \mathbb{P}\left(\frac{|Z_j|}{s_j} > \frac{\lambda_j}{s_j}\right) && (5.54)\\
&\leq \sum_{j \in V} \left(\frac{s_j}{\lambda_j} e^{-\lambda_j^2/(2s_j^2)} + \frac{\mu_j^2}{4}\right) && (5.55)\\
&\leq \frac{1}{c_n \sqrt{2\log(dc_n)}} + \frac{1}{4}\sum_{j \in V} \mu_j^2. && (5.56)
\end{aligned}$$

From Assumption (A1), we have that $\sum_{j \in V} \mu_j^2 = o_P(1)$, and thus this sum tends to zero with n. From condition (3.10), we have also that

$$\frac{1}{c_n \sqrt{2\log(dc_n)}} = O\left(\frac{1}{\sqrt{\log\log n}}\right). \tag{5.57}$$

Therefore, with probability tending to 1, $h_j \geq \log^{-k/d} n$ for each $j > r$.

Now consider $j \leq r$. By (5.35), $|\mu_j| \geq c h_j |m_{jj}(x)|$. Without loss of generality assume that $c h_j m_{jj}(x) > 0$.

We claim that in iteration t of the algorithm, if

$$t \ \leq \ \frac{1}{4+r} \log_{1/\beta}\left(\frac{c^2 n A_{\min}^2}{4C \log^k n \log(c_n d)}\right) \tag{5.58}$$

then

$$\mathbb{P}(h_j = h_0 \beta^t, \text{ for all } j \leq r) \longrightarrow 1 \tag{5.59}$$

To show this, note that (5.58) is written equivalently as

$$\left(\frac{1}{\beta}\right)^{t(4+r)} \ \leq \ \frac{c^2 n A_{\min}^2}{4C \log^k n \log(c_n d)}. \tag{5.60}$$

Except on an event of vanishing probability,

$$\prod_{j>r} \frac{1}{h_j} \ \leq \ \log^k n\,. \tag{5.61}$$

So on this event, we have

$$\begin{aligned}
\frac{\lambda_j^2}{h_j^2} &= \frac{2s_j^2 \log(c_n d)}{h_j^2} = \frac{2C \log(c_n d)}{n h_j^4} \prod_i \frac{1}{h_i} && (5.62)\\
&\leq \frac{2C \log^k n \log(c_n d)}{n} \left(\frac{1}{\beta}\right)^{(4+r)t} \leq \frac{c^2 A_{\min}^2}{2} \leq \frac{c^2 m_{jj}(x)^2}{2} && (5.63)
\end{aligned}$$

which implies that

$$cm_{jj}(x)h_j \geq 2\lambda_j \tag{5.64}$$

and hence

$$\frac{cm_{jj}(x)h_j - \lambda_j}{s_j} \geq \frac{\lambda_j}{s_j} = 2\sqrt{\log(c_n d)} \tag{5.65}$$

for each $j \leq r$. Now,

$$\begin{aligned}
\mathbb{P}(\text{rodeo halts}) &= \mathbb{P}(|Z_j| < \lambda_j \text{ for all } j \leq r) \leq \mathbb{P}(|Z_j| < \lambda_j \text{ for some } j \leq r) && (5.66)\\
&\leq \sum_{j\leq r} \mathbb{P}(|Z_j| < \lambda_j) \leq \sum_{j\leq r} \mathbb{P}(Z_j < \lambda_j) && (5.67)\\
&\leq \sum_{j\leq r} \mathbb{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{\mu_j - \lambda_j}{s_j}\right) && (5.68)\\
&\leq \sum_{j\leq r} \mathbb{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{cm_{jj}(x)h_j - \lambda_j}{s_j}\right) && (5.69)\\
&\leq \frac{r}{2c_n d\sqrt{\log(c_n d)}} && (5.70)
\end{aligned}$$

Finally, summing over all iterations $s \leq t$ gives

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{s\leq t}\bigcup_{j\leq r}\left\{|Z_j^{(s)}| < \lambda_j^{(s)}\right\}\right) &\leq \frac{tr}{2c_n d\sqrt{\log(c_n d)}} && (5.71)\\
&\leq \frac{r}{r+4}\,\frac{\log_{1/\beta}\left(\dfrac{c^2 n A_{\min}^2}{4C\log^k n \log(c_n d)}\right)}{2c_n d\sqrt{\log(c_n d)}} && (5.72)\\
&= O\left(\frac{1}{\sqrt{\log\log n}}\right) && (5.73)
\end{aligned}$$

by (3.10). Thus, the bandwidths h_j for $j \leq r$ satisfy, with high probability,

$$\begin{aligned}
h_j \;=\; h_0\beta^t &\leq h_0\left(\frac{4C\log^k n\log(c_n d)}{c^2 A_{\min}^2 n}\right)^{1/(4+r)} && (5.74)\\
&= n^{-1/(4+r)} h_0\left(\frac{4C\log^k n\log(c_n d)}{c^2 A_{\min}^2}\right)^{1/(4+r)} && (5.75)\\
&\leq n^{-1/(4+r)}\left(\frac{4C\log^{k+5\ell} n\log(c_n d)}{c^2 A_{\min}^2}\right)^{1/(4+r)} && (5.76)
\end{aligned}$$

using the condition

$$h_0 \leq \log^{\ell/d} n = \left(\log^{\ell(4+r)/d} n\right)^{1/(4+r)} \leq \left(\log^{5\ell} n\right)^{1/(4+r)} \tag{5.77}$$

since generally $(4+r)/d \leq 5$.

We next show that the algorithm is unlikely to reach iteration s, if

$$s \;\geq\; \frac{1}{4+r}\log_{1/\beta}\left(\frac{A_{\max}^2 n\log^{\ell+1} n}{2C}\right) \tag{5.78}$$

Specifically, if s satisfies (5.78) then $\mathbb{P}\left(h_j \leq h_0\beta^s, \text{ for all } j \leq r\right) \longrightarrow 0$. To show this, note that (5.78) can be written as

$$\left(\frac{1}{\beta}\right)^{s(4+r)} \;\geq\; \frac{A_{\max}^2 n\log n\log^{\ell} n}{2C} \tag{5.79}$$

$$\;\geq\; \frac{A_{\max}^2 n\log(n)\, h_0^d}{2C} \tag{5.80}$$

This implies that

$$\frac{1}{h_j^4}\prod_k \frac{1}{h_k} \geq \frac{1}{h_0^d}\left(\frac{1}{\beta}\right)^{s(4+r)} \;\geq\; \frac{A_{\max}^2 n\log n}{2C}. \tag{5.81}$$

since $h_i \leq h_0$ and $h_0 \geq 1$. Therefore

$$A_{\max}^2 \;\leq\; \frac{2}{\log n}\frac{s_j^2}{h_j^2} \tag{5.82}$$

and thus

$$A_j h_j \;\leq\; \frac{\sqrt{2\log n}}{\log n}\, s_j\,. \tag{5.83}$$

It follows that in iteration s

$$\mathbb{P}(|Z_j| \geq \lambda_j \text{ for some } j \leq r) \;\leq\; \sum_{j\leq r}\mathbb{P}\left(\frac{|Z_j - A_j h_j|}{s_j} \geq \frac{\lambda_j - A_j h_j}{s_j}\right) \tag{5.84}$$

$$\leq\; \sum_{j\leq r}\left(\frac{s_j}{\lambda_j - A_j h_j}\right)\exp\left(-\frac{1}{2}\left(\frac{\lambda_j - A_j h_j}{s_j}\right)^2\right) \tag{5.85}$$

$$\leq\; \frac{r}{n}\frac{1}{\sqrt{2\log n}(1-1/\log n)}e^{2-1/\log n} \;\to 0 \tag{5.86}$$

since $r = O(n)$. Therefore, the bandwidths h_j for $j \leq r$ satisfy, with high probability,

$$h_j \;\geq\; h_0\beta^s \;\geq\; h_0\left(\frac{2C}{A_{\max}^2\, n\log^{\ell+1} n}\right)^{1/(4+r)} \tag{5.87}$$

$$=\; n^{-1/(4+r)}\, h_0\,\left(\frac{2C}{A_{\max}^2\,\log^{\ell+1} n}\right)^{1/(4+r)} \tag{5.88}$$

$$\geq\; n^{-1/(4+r)}\,\left(\frac{2C}{A_{\max}^2\,\log^{\ell+1} n}\right)^{1/(4+r)} \tag{5.89}$$

which gives the statement of the theorem. □

VI. Some Modifications, Examples, and Remarks

In this section we discuss several extensions of the basic hard thresholding version of the rodeo, including a soft thresholding version, a global rather than local bandwidth selection procedure, the use of testing and generalized cross validation, and connections to least angle regression. Further numerical examples are also given to illustrate these ideas.

A. Estimating σ

The algorithm requires that we insert an estimate $\widehat{\sigma}$ of σ in (3.9). An estimator for σ can be obtained by generalizing a method of Rice (1984). For $i < \ell$, let

$$d_{i\ell} = \|X_i - X_\ell\|. \tag{6.1}$$

Fix an integer J and let $\mathcal{E}$ denote the set of pairs (i, ℓ) corresponding the J smallest values of $d_{i\ell}$. Now define

$$\widehat{\sigma}^2 = \frac{1}{2J} \sum_{i,\ell \in \mathcal{E}} (Y_i - Y_\ell)^2. \tag{6.2}$$

Then,

$$\mathbb{E}(\widehat{\sigma}^2) \;=\; \sigma^2 + \text{bias} \tag{6.3}$$

where

$$\text{bias} \;\leq\; D \sup_x \sum_{j=1}^{r} \left| \frac{\partial m_j(x)}{\partial x_j} \right| \tag{6.4}$$

with D given by

$$D = \max_{i,\ell \in \mathcal{E}} \|X_i - X_\ell\|. \tag{6.5}$$

There is a bias-variance tradeoff: large J makes $\widehat{\sigma}^2$ positively biased, and small J makes $\widehat{\sigma}^2$ highly variable. Note, however, that the bias is mitigated by sparsity (small r).

A more robust estimate may result from taking

$$\widehat{\sigma}^2 = \frac{\sqrt{\pi}}{2} \,\text{median}\,\{|Y_i - Y_\ell|\}_{i,\ell \in \mathcal{E}} \tag{6.6}$$

where the constant comes from observing that if X_i is close to X_ℓ, then $|Y_i - Y_\ell| \sim |N(0, 2\sigma^2)| = \sqrt{2}\sigma|Z|$, where Z is a standard normal with $\mathbb{E}|Z| = \sqrt{2/\pi}$.

Now we redo the earlier examples, taking σ as unknown. Figure 5 shows the result of running the algorithm on the examples of Section 4.A, however now estimating the noise using estimate (6.6). For the higher dimensional example, with $d = 20$, the noise variance is over-estimated, with the primary result that the irrelevant variables are more aggressively thresholded out; compare Figure 5 to Figure 3.

Although we do not pursue it in this paper, there is also the possibility of allowing $\sigma(x)$ to be a function of x and estimating it locally.

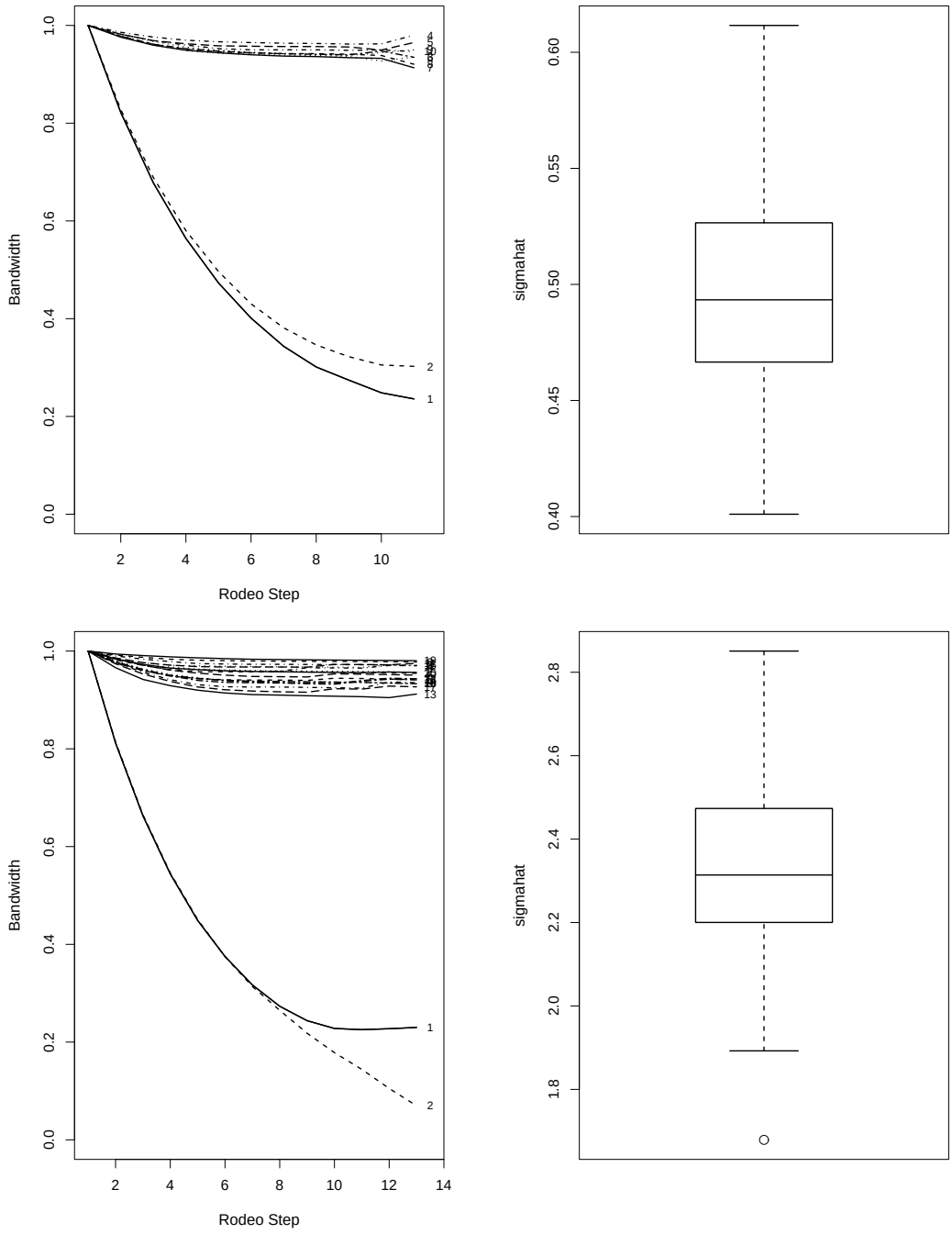

Figure 5: Rodeo run on the examples of Section 4.A, but now estimating the noise using the estimate $\widehat{\sigma}$ discussed in Section 6.A. Top: $\sigma = .5$, $d = 10$; bottom: $\sigma = 1$, $d = 20$. In higher dimensions the noise is over-estimated (right plots), which results in the irrelevant variables being more aggressively eliminated; compare Figure 3.

B. Subtracting off a Linear Lasso

Local linear regression is a nonparametric method that contains linear regression as a special case when $h \to \infty$. If the true function is linear but only a subset of the variables are relevant, then the rodeo will fail to separate the relevant and irrelevant variables since relevance is defined in terms of departures from the limiting parametric model. Indeed, the results depend on the Hessian of m which is zero in the linear case. The rodeo may return a full linear fit with all variables. A simple modification fixes this problem. First, do linear variable selection using, say, the lasso (Tibshirani, 1996). Then run the rodeo on the residuals from that fit.

C. Other Estimators and Other Paths

We have taken the estimate

$$\widehat{D}_j(h) = Z_j(h) I(|Z_j(h)| > \lambda_j) \tag{6.7}$$

with the result that

$$\widetilde{m}(x) = \widehat{m}_{h_0}(x) - \int_0^1 \langle \widehat{D}(s), \dot{h}(s) \rangle ds = \widehat{m}_{h\star}(x). \tag{6.8}$$

There are many possible generalizations. First, we can replace $\widehat{D}$ with the soft-thresholded estimate

$$\widehat{D}_j(t) \;=\; \text{sign}(Z_j(h)) \, (|Z_j(h)| - \lambda_j)_+ \tag{6.9}$$

where the index t denotes the t^{th} step of the algorithm. Since h_j is updated multiplicatively as $h_j \leftarrow \beta h_j$, the differential $dh_j(t)$ is given by $dh_j(t) = (1-\beta)h_j$. Using the resulting estimate of $D(t)$ and finite difference approximation for $\dot{h}(t)$ leads to the algorithm detailed in Figure 6.

Figure 7 shows a comparison of the hard and soft thresholding versions of the rodeo on the example function $m(x) = 2(x_1+1)^3 + 2\sin(10\,x_2)$ in $d = 10$ dimensions with $\sigma = 1$; β was set to 0.9. For each of 100 randomly generated datasets, a random test point $x \sim \text{Uniform}(0,1)^d$ was generated, and the difference in losses was computed:

$$(\widetilde{m}_{\text{hard}}(x) - m(x))^2 - (\widetilde{m}_{\text{soft}}(x) - m(x))^2 \; . \tag{6.10}$$

Thus, positive values indicate an advantage for soft thresholding, which is seen to be slightly more robust on this example.

Another natural extension would be to consider more general paths than paths that are restricted to be parallel to the axes. We leave this direction to future work.

D. Global Version

We have focused on estimation of m locally at a point x. The idea can be extended to carry out global bandwidth and variable selection by averaging over multiple evaluation points $x_1, \ldots, x_k$. These could be points interest for estimation, could be randomly chosen, or could be taken to be identical to the observed X_is.

1. *Select* parameter $0 < \beta < 1$ and initial bandwidth h_0, satisfying $1 \leq h_0 \leq \log^{\ell/d} n$, for a fixed constant ℓ. Let c_n be a sequence satisfying $dc_n = \Omega(\log n)$.

2. *Initialize* the bandwidths, and activate all covariates:

   (a) $h_j = h_0,\ j = 1, 2, \ldots, d.$

   (b) $\mathcal{A} = \{1, 2, \ldots, d\}$

   (c) Initialize step, $t = 1$.

3. *While* $\mathcal{A}$ *is nonempty*

   (a) Set $dh_j(t) = 0,\ j = 1, \ldots, d.$

   (b) Do for each $j \in \mathcal{A}$:

      (1) Compute the estimated derivative expectation Z_j and s_j.

      (2) Compute the threshold $\lambda_j = s_j\sqrt{2\log(dc_n)}$.

      (3) If $|Z_j| > \lambda_j$, then set $dh_j(t) = (1-\beta)\, h_j$ and $h_j \leftarrow \beta h_j$; otherwise remove j from $\mathcal{A}$.

      (4) Set $\widehat{D}_j(t) = \text{sign}(Z_j(h))\,(|Z_j(h)| - \lambda_j)_+$.

   (c) Increment step, $t \leftarrow t + 1$.

4. *Output* bandwidths $h^\star = (h_1, \ldots, h_d)$ and estimator

$$\widetilde{m}(x) = \widehat{m}_{h_0}(x) - \sum_{s=1}^{t} \big\langle \widehat{D}(s), dh(s) \big\rangle. \tag{6.11}$$

Figure 6: The soft thresholding version of the rodeo.

Averaging the Z_js directly leads to a statistic whose mean for relevant variables is asymptotically $k^{-1}h_j \sum_{i=1}^{k} m_{jj}(x_i)$. Because of sign changes in $m_{jj}(x)$, cancellations can occur resulting in a small value for the statistic. To eliminate the sign cancellation, we square the statistic.

Let $x_1, \ldots, x_k$ denote the evaluation points. Let

$$Z_j(x_i) = \sum_{s=1}^{n} Y_s G_j(X_s, x_i). \tag{6.12}$$

Then define the statistic

$$T_j \equiv \frac{1}{k}\sum_{i=1}^{k} Z_j^2(x_i) = \frac{1}{k} Y^T P_j Y \tag{6.13}$$

where $P_j = \mathcal{G}_j \mathcal{G}_j^T$, with $\mathcal{G}_j(s,i) = G_j(X_s, x_i)$.

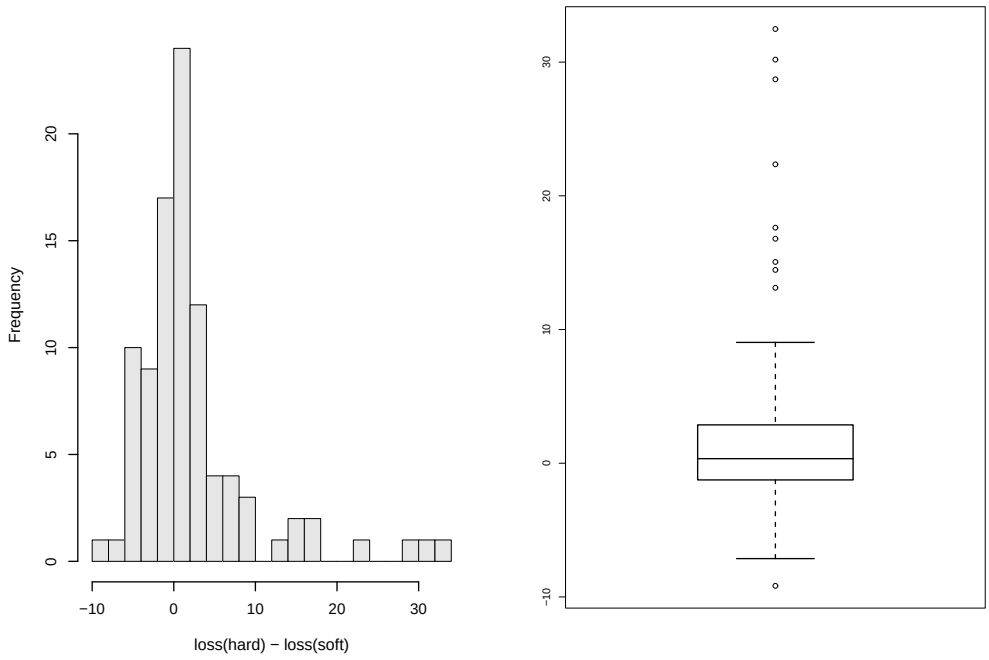

Figure 7: Comparison of hard and soft thresholding. The true function is $m(x) = 2(x_1 + 1)^3 + 2\sin(10\, x_2)$, $d = 10$ and $\sigma = 1$. The hard and soft thresholding versions of the rodeo were compared on 100 randomly generated datasets, with a single random test point x chosen for each; $\beta = 0.9$. The plots show two views of the difference of losses, $(\widetilde{m}_{\text{hard}}(x) - m(x))^2 - (\widetilde{m}_{\text{soft}}(x) - m(x))^2$; positive values indicate an advantage for soft thresholding.

If $j \in R^c$ then we have $\mathbb{E}(Z_j(x_i)) = o(1)$, so it follows that, conditionally,

$$\mathbb{E}(T_j) = \frac{\sigma^2}{k}\text{tr}(P_j) + o_P(1) \tag{6.14}$$

$$\mathbb{V}(T_j) = \frac{2\sigma^4}{k^2}\text{tr}(P_j P_j) + o_P(1). \tag{6.15}$$

We take the threshold to be

$$\lambda_j = \frac{\widehat{\sigma}^2}{k}\text{tr}(P_j) + 2\sqrt{\frac{2\widehat{\sigma}^4}{k^2}\text{tr}(P_j P_j)\log(c_n d)}. \tag{6.16}$$

We give an example of this algorithm in the following section.

E. Greedy Rodeo and LARS

The rodeo is related to least angle regression (LARS) (Efron et al., 2004). In forward stagewise linear regression, one performs variable selection incrementally. LARS gives a refinement where at

each step in the algorithm, one adds the covariate most correlated with the residuals of the current fit, in small, incremental steps. LARS takes steps of a particular size: the smallest step that makes the largest correlation equal to the next-largest correlation. Efron et al. (2004) show that the lasso can be obtained by a simple modification of LARS.

The rodeo can be seen as a nonparametric version of forward stagewise regression. Note first that Z_j is essentially the correlation between the Y_is and the $G_j(X_i, x, h)$s (the change in the effective kernel). Reducing the bandwidth is like adding in more of that variable. Suppose now that we make the following modifications to the rodeo: (i) change the bandwidths one at a time, based on the largest $Z_j^* = Z_j/\lambda_j$, (ii) reduce the bandwidth continuously, rather than in discrete steps, until the largest Z_j is equal to the next largest. This version can then be thought of as a nonparametric formulation of LARS.

In fact, we can go further and embed the rodeo within LARS to get a fast nonparametric method. We do this by replacing the derivatives of the fit in the rodeo with differences. Then we iterate variable selection with bandwidth selection.

- Set $h = (h_0, \ldots, h_0)$. Define d-dimensional pseudo-covariates $\widetilde{X}_i$, $i = 1, \ldots, n$, by

$$\widetilde{X}_i(j) = G_j(X_i, x, h), \quad j = 1, \ldots, d. \tag{6.17}$$

  Now run the LARS algorithm, regressing the Y_i's on the pseudo-covariates, up to some pre-defined stopping point. This step essentially chooses relevant variables at the resolution of the starting bandwidth $h = (h_0, \ldots, h_0)$.

- Define new pseudo-covariates $\widetilde{X}_i$, $i = 1, \ldots, n$, by

$$\widetilde{X}_i(j) = G_j(X_i, x, h') - G_j(X_i, x, h), \quad j = 1, \ldots, d \tag{6.18}$$

  where h' has h_j replaced by βh_j. Note that adding the j^{th} covariate corresponds to reducing the bandwidth from h_j to βh_j. Now run the LARS algorithm up to some pre-defined stopping point.

- Repeat the last step until a stopping criterion is satisfied.

One advantage of this method is that it can be implemented using existing LARS software. We leave the development of the theory for this approach to future work. Some examples of the greedy version of this algorithm follow.

E.1 Diabetes example

Figure 8 shows the result of running the greedy version of the rodeo on the diabetes dataset used by Efron et al. (2004) to illustrate LARS . The algorithm averages Z_j^* over a randomly chosen set of $k = 100$ data points, and reduces the bandwidth for the variable with the largest value; note that no estimate of σ is required. The resulting variable ordering is seen to be very similar to, but

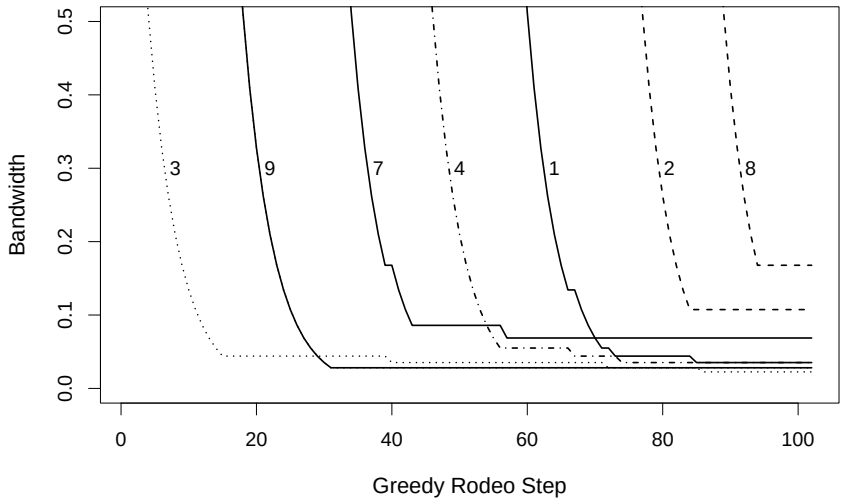

Figure 8: Greedy rodeo on the diabetes data, used to illustrate LARS (Efron et al., 2004). A set of $k = 100$ of the total $n = 442$ points were sampled ($d = 10$), and the bandwidth for the variable with largest average $|Z_j|/\lambda_j$ was reduced in each step.

different from, the ordering obtained from the parametric LARS fit. The variables were selected in the order 3 (body mass index), 9 (serum), 7 (serum), 4 (blood pressure), 1 (age), 2 (sex), 8 (serum), 5 (serum), 10 (serum), 6 (serum). The LARS algorithm adds variables in the order 3, 9, 4, 7, 2, 10, 5, 8, 6, 1. One notable difference is in the position of the age variable.

E.2 Turlach's example

In the discussion to the LARS paper, Berwin Turlach (Turlach, 2004) gives an interesting example of where LARS and the lasso fails. The function is

$$Y = \left(X_1 - \frac{1}{2}\right)^2 + X_2 + X_3 + X_4 + X_5 + \varepsilon \tag{6.19}$$

with ten variables $X_i \sim \text{Uniform}\,(0, 1)$ and $\sigma = 0.05$. Although X_1 is a relevant variable, it is uncorrelated with Y, and LARS and the lasso miss it.

Figure 9 shows the greedy algorithm on this example, where bandwidth corresponding to the largest average Z_j^* is reduced in each step. We use kernel regression rather than local linear regression as the underlying estimator. The variables x_2, x_3, x_4, x_5 are selected first in every run. Variable x_1 is selected fifth in 72 of the 100 runs; a typical run of the algorithm is shown in the left plot. In contrast, as discussed in Turlach (2004), LARS selects x_1 in position 5 about 25% of the time.

Figure 10 shows bandwidth traces for this example using the global algorithm described in Section 6.D with $k = 20$ evaluation points randomly subselected from the data, and σ taken to be known. Before starting the rodeo, we subtract off a linear least squares fit. The first plot shows

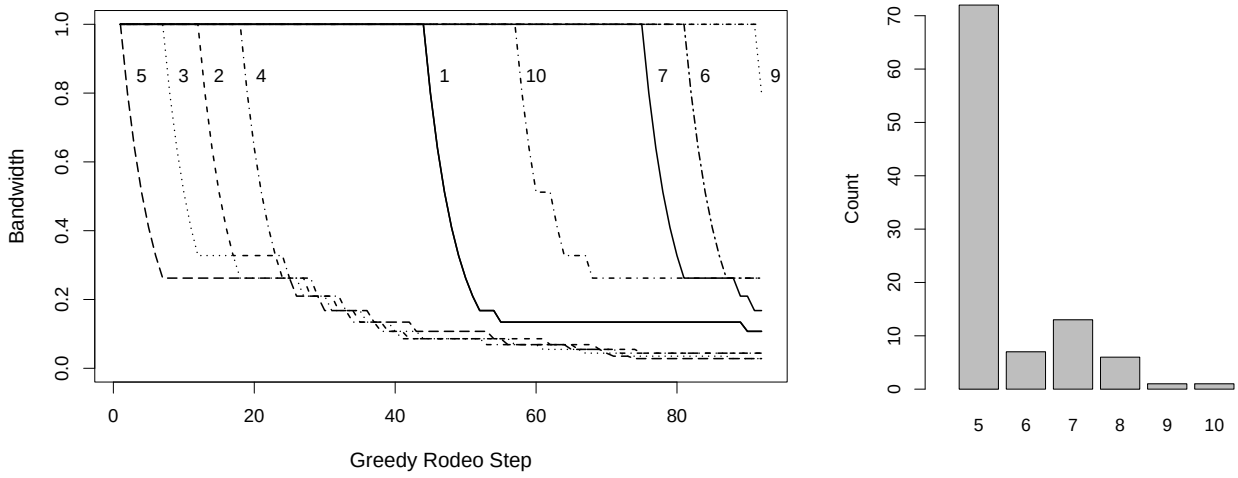

Figure 9: Left: A typical run of the greedy algorithm on Turlach's example. The bandwidths are first reduced for variables x_2, x_3, x_4, x_5, and then the relevant, but uncorrelated with Y variable x_1 is added to the model; the irrelevant variables follow. Right: Histogram of the position variable x_1 is selected, over 100 runs of the algorithm

$h_1, \ldots, h_5$. The lowest line is h_1 which shrinks the most since m is a nonlinear function of x_1. The other curves are the linear effects. The right plot shows the traces for $h_6, \ldots, h_{10}$, the bandwidths for the irrelevant variables.

F. The Cross-Validation Rodeo

One can incorporate other tests into the rodeo as well. Here we describe a test based on generalized cross validation (GCV).

Recall that $\widehat{m}_h(x) = S_x Y$ where $S_x = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x$. An estimate of the risk of the estimator is the GCV score (Wahba, 1990) defined by

$$\widehat{\mathcal{R}}_{\text{gcv}}(h_1, \ldots, h_d) = (1 - \nu/n)^{-2} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{m}_h(X_i))^2 \tag{6.20}$$

where $\nu = \sum_{i=1}^{n} S_{X_i}(i)$ is the effective degrees of freedom. Define the test statistic T_j by

$$T_j = \widehat{\mathcal{R}}_{\text{gcv}}(h_1, \ldots, h_j, \ldots, h_d) - \widehat{\mathcal{R}}_{\text{gcv}}(h_1, \ldots, \beta h_j, \ldots, h_d). \tag{6.21}$$

To assess the significance of T_j a permutation approach can be used. Randomly permute the values of the j^{th} covariate and recompute the statistic. Repeat this k times to yield values $T_j^{(1)}, \ldots, T_j^{(k)}$.

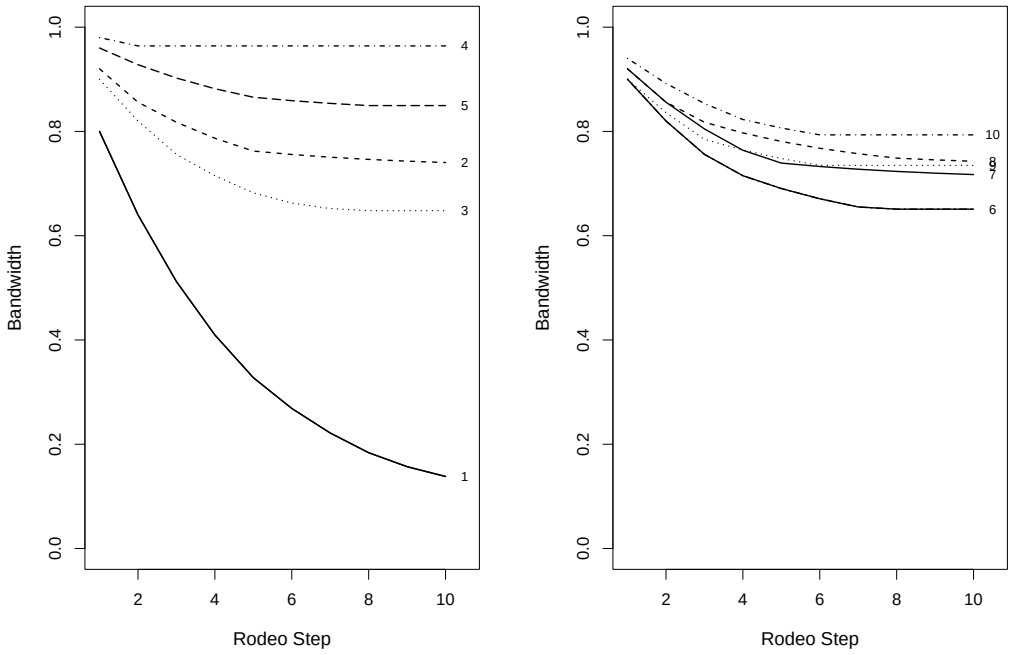

Figure 10: The global rodeo averaged over 10 runs on Turlach's example. The left plot shows the bandwidths for the five relevant variables. Since the linear effects (variables two through five) have been subtracted off, bandwidths h_2, h_3, h_4, h_5 are not shrunk. The right plot shows the bandwidths for the other, irrelevant, variables.

Using the estimated p-value

$$p_j = \frac{1}{k}\sum_{i=1}^{k} \mathbb{I}(T_j > T_j^{(i)}), \tag{6.22}$$

the algorithm replaces h_j with βh_j if

$$p_j < \lambda \equiv \frac{\alpha}{dc_n} \tag{6.23}$$

where α is set by the user. An advantage of this method is that it does not require estimating σ.

Figure 11 shows one run of the cross-validation rodeo, with $\alpha = 0.05$, for Example 1 where $m(x) = 5x_1^2x_2^2$ with $d = 10$. The lower two traces correspond to h_1 and h_2. The traces for the irrelevant variables $x_3, \ldots, x_{10}$ are the same and correspond to the top dashed line.

G. Non-normal errors

We have assumed that the residuals are normally distributed. In fact, by the central limit theorem, Z_j is approximately normal even if the residuals are not. However, it is possible to eliminate the

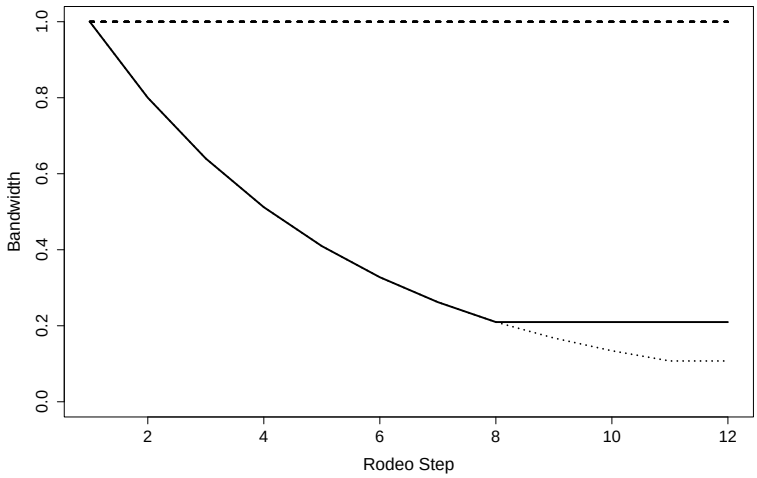

Figure 11: One run of the cross-validation rodeo on the first example with $m(x) = 5x_1^2x_2^2$. The lower two traces correspond to h_1 and h_2. The top dashed line is the trace for $h_3, \ldots, h_{10}$.

normality assumption altogether by replacing the Gaussian tail inequalities in the analysis with Markov's inequality. Of course, this leads to somewhat weaker results.

H. Local Likelihood and Generalized Linear Models

The lasso extends naturally to generalized linear models by regularizing the coefficients with an L_1 penalty. Ng (2004) has given a covering number analysis of the lasso applied to logistic regression, for cases where the underlying decision function is sparse. A nonparametric version of the lasso, as a form of basis pursuit for spline models, is described by Zhang et al. (2005). The rodeo can be naturally extended to nonparametric classification using local likelihood and generalized linear models by suitably redefining the statistic Z_j. We expect that similar analytic results will obtain.

I. Very Large Dimensions

If d is extremely large, say $d \gg n$, then the current method is not feasible. Here we suggest an approach to such cases. We will report in greater detail in a future paper.

First, we replace local linear regression with kernel regression since the latter is still well defined even when $d > n$. For large enough bandwidth h_0, $\widehat{m}_{h_0}(x) \approx \overline{Y}$ and we take this as a starting bandwidth. We compute Z_j as before but it may no longer be reasonable to assume that $\mu_j = \mathbb{E}(Z_j) \approx 0$. Instead, we try to separate Z_j into small effects and large effects. That is, we model the Z_js as a mixture:

$$Z_j \sim (1-a)F_0 + aF_1 \tag{6.24}$$

where a denotes the fraction of large effects, F_0 is the distribution of Z for small effects and F_1

is the distribution of Z for large effects. Using recent techniques in multiple testing (Efron, 2005; Genovese and Wasserman, 2004) we can estimate the mean Δ and variance s^2 of F_0. The estimates are biased but under the sparsity condition that a is small, the bias is small. Then we use the thresholds

$$\lambda_j = \widehat{\Delta} + s\sqrt{2\log(dc_n)}. \tag{6.25}$$

In the case where the variables do not separate nicely into large and small effects, we will end up stopping early and choosing large bandwidths. In a very high dimensional problem where the variables do not separate well, this is, arguably, not an unreasonable solution.

References

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and regression trees*. Wadsworth Publishing Co Inc.

Bühlmann, P. and Yu, B. (2005). Boosting, model selection, lasso and nonnegative garrote. *Technical report, Berkeley* .

Donoho, D. (2004). For most large underdtermined systems of equations, the minimal ℓ^1-norm near-solution approximates the sparest near-solution. *Technical report, Stanford* .

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** 425–455.

Efron, B. (2005). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association* **99** 96–104.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32** 407–499.

Fan, J. (1992). Design-adaptive nonparametric regression. *Journal of the American Statistical Association* **87** 998–1004.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* **19** 1–67.

Fu, W. and Knight, K. (2000). Asymptotics for lasso type estimators. *The Annals of Statistics* **28** 1356–1378.

Genovese, C. and Wasserman, L. (2004). A stochastic process approach to false discovery control. *The Annals of Statistics* **32** 1035–1061.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistical Sinica* **7** 339–373.

Girosi, F. (1997). An equivalence between sparse approximation and support vector machines. Tech. rep., Massachusetts Institute of Technology. A.I. Memo No. 1606.

HASTIE, T. and LOADER, C. (1993). Local regression: Automatic kernel carpentry. *Statistical Science* **8** 120–129.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. H. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.

LAWRENCE, N. D., SEEGER, M. and HERBRICH, R. (2003). Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*. MIT Press.

LEPSKI, O. V., MAMMEN, E. and SPOKOINY, V. G. (1997). Optimal spatial adaptation to inhomogeneous smoothness: An approach based on kernel estimates with variable bandwidth selectors. *The Annals of Statistics* **25** 929–947.

LI, L., COOK, R. D. and NACHSTEIM, J., CHRISTOPHER (2005). Model-free variable selection. *J. R. Statist. Soc. B.* **67** 285–299.

NG, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In *21st International Conference on Machine Learning, ICML-04*.

RICE, J. (1984). Bandwidth choice for nonparametric regression. *The Annals of Statistics* **12** 1215–1230.

RUPPERT, D. (1997). Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation. *Journal of the American Statistical Association* **92** 1049–1062.

RUPPERT, D. and WAND, M. P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics* **22** 1346–1370.

SMOLA, A. and BARTLETT, P. (2001). Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems 13*. MIT Press.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, Methodological* **58** 267–288.

TIPPING, M. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1** 211–244.

TURLACH, B. (2004). Discussion of "least angle regression". *The Annals of Statistics* **32** 494–499.

WAHBA, G. (1990). *Spline models for observational data*. SIAM [Society for Industrial and Applied Mathematics].

ZHANG, H., WAHBA, G., LIN, Y., VOELKER, M., FERRIS, R. K. and KLEIN, B. (2005). Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association* **99** 659–672.