Nonparametric Density Estimation in High Dimensions Using the Rodeo

Han Liu John Lafferty Larry Wasserman
Carnegie Mellon University

ABSTRACT

We consider the problem of estimating the joint density of a d-dimensional random vector $X = (X_1, X_2, ..., X_d)$ when d is large. We assume that the density is a product of a parametric baseline component and a nonparametric component. The nonparametric component depends on an unknown subset of the variables. If this subset is small, then nonparametric estimates with fast rates of convergence are possible. Using a modification of a previously developed nonparametric regression framework called *rodeo* (regularization of derivative expectation operator), we propose a method to exploit this fact. The method selects the bandwidths in an incremental way making it computationally attractive. We empirically show that the density rodeo works well even for very high-dimensional problems. When the unknown density function satisfies some suitably defined sparsity conditions, our approach avoids the curse of dimensionality and achieves an optimal converge rate of the risk. Because it is a greedy algorithm, bandwidth selection is fast. When the parametric baseline is a very smooth distribution, we also provide theoretical guarantees on the behavior of the method.

Keywords: nonparametric density estimation, sparsity, adaptive bandwidth selection, high dimensionality

1 Introduction

Let $X_1, X_2, ..., X_n$ be a sample from a distribution F with density f. We are interested in estimating the density f when the dimension d of X_i is moderate or large. Methods for estimating f include the kernel estimator [1, 2], local likelihood Hjort et al. and Loader [3, 4, 5] and others. These methods work very well for low-dimensional problems ($d \leq 3$) but are not effective for high-dimensional problems. The major difficulty is due to the intractable computational cost of cross validation when bandwidths need to be selected for each dimension, and the slow rates of convergence of the estimator. Density estimation in high dimensions are usually done by mixture models [6, 7, 8, 9]. However, mixture models with a fixed number of components are parametric and only useful to the extent that the assumed model is right. Mixture models without a fixed number of components are nonparametric and achieve, at best, the same rates as kernel estimators. In fact, the theoretical guaranttees with mixtures are generally not as good as for kernel estimators, see: Genovese and Wasserman [10] and Ghosal and van der Vaart [11]. Other methods for high dimensional density estimation include projection pursuit [12], log-spline model [13] and penalized likelihood [14].

In a d-dimensional space, minimax theory shows that the best convergence rate for the mean squared error under standard smoothness assumptions is $\mathcal{R}_{opt} = O(n^{-4/(4+d)})$ which represents the "curse of dimensionality" when d is large. In this paper we present a method that acheives faster rates of convergence when a certain sparsity assumption is satisfied. Morever, it is a greedy method and so is computationally expedient for large d.

The idea comes from a newly developed nonparametric regression framework called *rodeo* [15]. For the regression problem, $Y_i = m(X_i) + \epsilon_i$, $i = 1, \ldots, n$, where $X_i = (X_{i1}, ..., X_{id}) \in \mathbf{R}^d$ is a d-dimensional vector. Assuming that the true function only depends on r covariates $r \ll d$, the rodeo can simultaneously perform bandwidth selection and (implicitly) variable selection to achieve a better minimax convergence rate of $O(n^{-4/(4+r)})$ up to a logarithmic factor, as if the r relevant variables were explicitly isolated in advance. The purpose of this paper is to extend this idea to the nonparametric density estimation setting. Toward this goal, we need to first define an appropriate "sparsity" condition in the density estimation setting. Our key assumption is

$$f(x_1, \ldots, x_d) = g(x_R) b(x_1, \ldots, x_d) \tag{1}$$

where g is an unknown function, $x_R = (x_j : j \in R)$, R is a subset of $\{1, \ldots, d\}$ and b is a baseline density (completely known or known up to finitely many parameters). If the number of coordinates in R is small then we can exploit the fact that the nonparametric component g only depends on a small number of variables. Two examples of this model are $b(x) =$ uniform so that $f(x) = g(x_R)$ and $b(x) =$ Normal as in Hjort et al. [3, 4]. In this paper, We will consider two versions of the rodeo for density estimation problems: a local version and a global version. The local version estimates f at a given point x and results in a local bandwidth selection method. The global version estimates f at all x and results in a global bandwidth selection method.

This paper is organized as follows: In section 2, we derived the local rodeo algorithm for both kernel density estimator and local likelihood estimator. The rodeo algorithm for a semiparametric model when $b(x) =$ Normal is also shown. Section 3 and 4 describe the global rodeo algorithms and other variations. Section 5 uses both synthetic and real-world dataset to test our method. Section 6 specifies our main theoretical results about the asymptotic running time, selected bandwidths, and convergence rate of the risk. The conclusions and more discussion is in section 7. All the proofs are given in the appendix.

2 The Local Rodeo

Suppose first that data are on the unit cube $[0,1]^d$ and $b(x)$ is uniform. Let x be a d-dimensional target point at which we want to estimate $f(x)$. The kernel density estimator is

$$\widehat{f}_H(x) = \frac{1}{n\det(H)}\sum_{i=1}^{n}\mathcal{K}(H^{-1}(x-X_i)) \tag{2}$$

where $\mathcal{K}$ is a symmetric kernel, such that $\int \mathcal{K}(u)du = 1$, $\int u\mathcal{K}(u)du = 0_d$ while $\mathcal{K}_H(\cdot) = \frac{1}{\det(H)}\mathcal{K}(H^{-1}\cdot)$ and $H = diag(h_1, ..., h_d)$. We assume that $\mathcal{K}$ is a product kernel so

$$\widehat{f}_H(x) = \frac{1}{n\det(H)}\sum_{i=1}^{n}\mathcal{K}(H^{-1}(x-X_i)) = \frac{1}{n}\sum_{i=i}^{n}\prod_{j=1}^{d}\frac{1}{h_j}K\left(\frac{x_j-X_{ij}}{h_j}\right) \tag{3}$$

2.1 The Kernel Density Estimator Version

The density rodeo is based on the following idea. We start with a bandwidth matrix $H = diag(h_0, \ldots, h_0)$ where h_0 is large. We then compute test statistics $(Z_j : 1 \le j \le d)$ and we reduce bandwidth h_j if Z_j is large. The test statistic is

$$Z_j \;=\; \frac{\partial \widehat{f}_H(x)}{\partial h_j} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial h_j}\left(\prod_{k=1}^{d}\frac{1}{h_k}K\left(\frac{x_k-X_{ik}}{h_k}\right)\right) \equiv \frac{1}{n}\sum_{i=1}^{n}Z_{ji}. \tag{4}$$

Thus, $|Z_j|$ is large if changing h_j leads to a substantial difference in the estimator. To carry out the test, we compare Z_j to its variance

$$\sigma_j^2 = \mathrm{Var}(Z_j) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n}Z_{ji}\right) = \frac{1}{n}\mathrm{Var}(Z_{j1}) \tag{5}$$

We estimate σ_j^2 with $s_j^2 = v_j^2/n$ where v_j^2 is the sample variance of the Z_{ji}'s. The algorithm is given in Figure 1.
For a general kernel, we have that

$$Z_j \;=\; \frac{\partial \widehat{f}_H(x)}{\partial h_j} = -\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{h_j} + \frac{x_j-X_{ij}}{h_j^2}\widetilde{K}\left(\frac{x_j-X_{ij}}{h_j}\right)\right)\prod_{k=1}^{d}\frac{1}{h_k}K\left(\frac{x_k-X_{ik}}{h_k}\right) \tag{6}$$

where $\widetilde{K}(x) = \frac{d\log K(x)}{dx}$. In the case where K is Gaussian this becomes

$$\begin{aligned}
Z_j &= \frac{\partial \widehat{f}_H(x)}{\partial h_j} && (7)\\
&= \frac{1}{nh_j^3}\prod_{k=1}^{d}\frac{1}{h_k}\sum_{i=1}^{n}\left((x_j-X_{ij})^2-h_j^2\right)\prod_{k=1}^{d}K\left(\frac{x_k-X_{ik}}{h_k}\right) && (8)\\
&\propto \frac{1}{n}\sum_{i=1}^{n}\left((x_j-X_{ij})^2-h_j^2\right)\prod_{k=1}^{d}K\left(\frac{x_k-X_{ik}}{h_k}\right) && (9)\\
&= \frac{1}{n}\sum_{i=1}^{n}\left((x_j-X_{ij})^2-h_j^2\right)\exp\left\{-\sum_{k=1}^{d}\frac{(x_k-X_{ik})^2}{2h_k^2}\right\} && (10)
\end{aligned}$$

```
1. Select parameter $0 < \beta < 1$ and initial bandwidth $h_0$, where $h_0$ is slowly decreasing to zero:
       $h_0 = c_0/\log\log n$
   for some constant $c_0$. Let $c_n$ be a sequence satisfying $c_n = O(\frac{\log n}{n})$.
2. Initialize the bandwidths, and activate all dimensions:
     (a) $h_j = h_0, j = 1, 2, ..., d$.
     (b) $\mathcal{A} = \{1, 2, ..., d\}$.
3. While $\mathcal{A}$ is nonempty, do for each $j \in \mathcal{A}$
     (a) Compute the estimated derivative and variance: $Z_j$ and $s_j^2$.
     (b) Compute the threshold $\lambda_j = s_j\sqrt{2\log(nc_n)}$.
     (c) If $|Z_j| > \lambda_j$, then set $h_j \leftarrow \beta h_j$; otherwise remove $j$ from $\mathcal{A}$.
4. Output bandwidths $H^* = diag(h_1, ..., h_d)$ and estimator $\widetilde{f}(x) = \widehat{f}_{H^*}(x)$
```

Figure 1: The density rodeo algorithm.

Here, the constant of proportionality $\frac{1}{h_j^3}\prod_{k=1}^d \frac{1}{h_k}$ can be safely ignored to avoid overflow in the computation as $h_k \to 0$ for large d.

2.2 The Local Likelihood Version

Hjort et al. and Loader [3, 4, 5] formulate the local likelihood density estimation problems as:

$$\max_\theta \mathcal{L}(f, x) = \sum_{i=1}^n \mathcal{K}\left(H^{-1}(X_i - x)\right) \log f(X_i; \theta) - n \int_{\mathcal{X}} \mathcal{K}\left(H^{-1}(u - x)\right) f(u; \theta) du \qquad (11)$$

which is a localized version of the usual loglikelihood function for density estimation problems:

$$\max_\theta \mathcal{L}(f, x) = \sum_{i=1}^n \log f(X_i; \theta) - n \left(\int_{\mathcal{X}} f(u; \theta) du - 1\right) \qquad (12)$$

Since the true density function f is unknown, a polynomial is used to approximate the log density. The large-sample properties of the local likelihood estimator are parallel to those of local polynomial regression. The most appealing property of the resulting estimator is its good performance when facing boundary effects [5]. When assuming a product Gaussian kernel, the closed form of the local likelihood estimator can be written as

$$\widetilde{f}_H(x) \quad = \quad \widehat{f}_H(x) \exp\left\{ -\frac{1}{2} \sum_{k=1}^d h_k^2 \left(\frac{\sum_{i=1}^n \prod_{j=1}^d K\left(\frac{X_{ij}-x_j}{h_j}\right) \left(\frac{X_{ik}-x_k}{h_k^2}\right)}{\sum_{i=1}^n \prod_{j=1}^d K\left(\frac{X_{ij}-x_j}{h_j}\right)} \right)^2 \right\} \qquad (13)$$

which can be viewed as a standard kernel density estimator $\widehat{f}_H(x)$ multiplied by an exponential bias correction term. To evaluate $Z_m = \frac{\partial \widehat{f}_H(x)}{\partial h_m}$, $m = 1, ..., d$, define

$$\widehat{g}_k(x) = \frac{\partial}{\partial x_k}\widehat{f}_H(x) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}\frac{1}{h_j}K\left(\frac{X_{ij}-x_j}{h_j}\right)\left(\frac{X_{ik}-x_k}{h_k^2}\right) \tag{14}$$

Then

$$\begin{aligned} Z_m &= \frac{\partial}{\partial h_m}\left(\widehat{f}_H(x)\exp\left\{-\frac{1}{2}\sum_{k=1}^{d}h_k^2\left(\frac{\widehat{g}_k(x)}{\widehat{f}_H(x)}\right)^2\right\}\right) && (15)\\ &= \widetilde{f}_H(x)\left(\frac{\partial}{\partial h_m}\log\widehat{f}_H(x)\right) + \widetilde{f}_H(x)\frac{\partial}{\partial h_m}\left(-\frac{1}{2}\sum_{k=1}^{d}h_k^2\left(\frac{\widehat{g}_k(x)}{\widehat{f}(x)}\right)^2\right) && (16)\end{aligned}$$

Where $\frac{\partial}{\partial h_m}\log\widehat{f}_H(x) = \frac{\frac{\partial}{\partial h_m}\widehat{f}_H(x)}{\widehat{f}_H(x)}$ has been calculated in the previous section. The derivation of the second term, though quite involved, is straightforward. The same algorithm in figure 1 applies.

2.3 Other Baseline Densities

When using a different baseline (i.e. the Normal distribution), we use the semiparametric density estimator

$$\bar{f}_H(x) = \frac{\widehat{b}(x)\sum_{i=1}^{n}\mathcal{K}_H(X_i - x)}{n\int\mathcal{K}_H(u-x)\widehat{b}(u)du} \tag{17}$$

where $\widehat{b}(x)$ is a parametric density estimator at point x, its parameters are estimated by maximum likelihood. Since the parameters in the parametric form are easy to estimate, we treat them as known. The motivation of this estimator comes from local likelihood method in equation (11): instead of using a polynomial $P(x)$ to approximate the log density $\log f(x)$, we use $\log b(x) + P(x)$. Under this setting, we see that starting from a large bandwidth, if the true function is $b(x)$, the algorithm will tend to freeze the bandwidth decaying process for the estimator defined in expression (17).

Suppose that $b(x)$ is a multivariate normal density function with a diagnolized variance-covariance matrix Σ. When we use the product Gaussian kernels with bandwidth matrix H, a closed form estimator can be derived as

$$\bar{f}_H(x) \;=\; \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}K\left(\frac{X_{ij}-x_j}{h_j}\right)\sqrt{\frac{|H+\widehat{\Sigma}|}{|\widehat{\Sigma}|}}\exp\left\{-\frac{(x-\widehat{\mu})^T\left(\widehat{\Sigma}^{-1}-(H+\widehat{\Sigma})^{-1}\right)(x-\widehat{\mu})}{2}\right\}$$

where $\widehat{\mu}$ and $\widehat{\Sigma}$ are the M.L.E. for the normal distribution. More details about the derivation of this closed form are given in the appendix. For the mean and variance of a multivariate normal distribution. It's easy to see that the local likelihood estimator is a special case of this semiparametric estimator when $b(x)$ = uniform. The partial derivative of $\bar{f}_H(x)$ with respect to the bandwidth h_m $(m = 1, ..., d)$ is calculated as

$$Z_m = \frac{\partial \bar{f}_H(x)}{\partial h_m} = \sqrt{\prod_{j=1}^{d} \left(1 + \frac{h_j^2}{\widehat{\sigma^2}_j}\right)} \exp\left\{\sum_{j=1}^{d} \left(-\frac{(x_j - \widehat{\mu}_j)^2}{2\left(\widehat{\sigma_j^2}(\widehat{\sigma_j^2} + h_j^2)/h_j^2\right)}\right) \left(\frac{\partial \widehat{f}_H(x)}{\partial h_m} + M\widehat{f}_H(x)\right)\right\} \tag{18}$$

where

$$M = \frac{h_m(2(\widehat{\sigma^2}_m + h_m^2) + (x_m - \widehat{\mu}_m)^2)}{2(\widehat{\sigma^2}_m + h_m^2)^2} \tag{19}$$

and $\widehat{f}_H(x)$ is the standard kernel density estimator as defined in equation (3). The variance of Z_m is estimated using the bootstrap method (see section 4.1).

3 The Global Rodeo

Instead of using the local rodeo which corresponds to the adaptive density estimation, the idea could be easily extended to carry out global bandwidth selection, in which case each dimension uses a fixed bandwidth. The idea is by averaging the test statistics for multiple evaluation points $x_1, ..., x_k$, these points could be sampled from the empirical distribution of the observed sample points.

As pointed out by Lafferty & Wasserman [15], averaging the Z_js directly leads to a statistic whose mean for relevant variables is asymptotically $\frac{1}{k} h_j \sum_{i=1}^{k} f_{jj}(x_i)$. Because of sign changes in $f_{jj}(x)$, cancellations can occur resulting in a small value for the statistics. To avoid this problem, the statistic is squared. Let $x_1, ..., x_m$ denote the evaluation points and $Z_j(x_i)$ represents the derivative for the i-th evaluation point with respect to the bandwidth h_j. Therefore

$$Z_j(x_i) = \frac{1}{n} \sum_{k=1}^{n} Z_{jk}(x_i), \quad i = 1, ..., m, \quad j = 1, ..., d \tag{20}$$

Let $\gamma_{jk} = (Z_{j1}(x_k), Z_{j2}(x_k), ..., Z_{jm}(x_k))^T$ $(k = 1, ..., n)$. assuming that $\mathbf{Var}(\gamma_{jk}) = \Sigma_j$, denote $Z_{j\cdot} = (Z_{j1}, Z_{j2}, ..., Z_{jm})^T$, by the multivariate central limit theorem, we know that $\mathbf{Var}(Z_{j\cdot}) = \Sigma_j/n \equiv C_j$. Based on which, we define the test statistic

$$T_j = \frac{1}{m} \sum_{k=1}^{m} Z_j^2(x_k), \quad j = 1, ..., d \tag{21}$$

while

$$s_j = \sqrt{\mathbf{Var}(T_j)} = \frac{1}{m}\sqrt{\mathbf{Var}(Z_j^T Z_j)} = \frac{1}{m}\sqrt{2tr(C_j^2) + 4\widehat{\mu_j}^T C_j \widehat{\mu_j}} \tag{22}$$

where $\widehat{\mu} = \frac{1}{m} \sum_{i=1}^{m} Z_j(x_i)$. For the irrelevant dimension $j \in R^c$, as will be shown in section 6, $\mathbf{E} Z_j(x_i) = o_P(h_j)$, so that $\mathbf{E} T_j \approx \mathbf{Var}(Z_j(x_i))$. We use s_j^2 as an estimate for $\mathbf{Var}(Z_j(x_i))$,Therefore, we take the threshold to be

$$\lambda_j = s_j^2 + 2s_j\sqrt{\log(nc_n)} \tag{23}$$

Several examples of this algorithm and its comparison with the other algorithms are given in the experiment section, the theoretical properties of the global rodeo estimator should be analyzed in a way that is similar to the local version.

4 Extensions

4.1 Bootstrap Version

For the previous examples, the explicit expression for the Z_j and s_j^2 can be easily derived due to the existence of a closed form for the targeted density estimators. Sometimes, the density estimator $\widehat{f}_H(x)$ might not have a closed form expression. In these cases, we could still numerically evaluate the derivative Z_j as

$$Z_j = \frac{\widehat{f}_{H+\triangle h_j}(x) - \widehat{f}_H(x)}{\triangle h_j} \tag{24}$$

where $H + \triangle h_j$ means adding a small value $\triangle h_j$ on the j-th diagonal element of H. The variance of Z_j can be calculated by bootstrap, the algorithm is given in figure 2

```
THE BOOTSTRAP METHOD TO CALCULATE $s_j^2$

1. Draw a sample $X_1^*, ..., X_n^*$ of size $n$, with replacement:
       Repeat $B$ times for the following
         Compute the estimate $Z_{ji}^*$ from data $X_1^*, ..., X_n^*$, $i = 1, ..., B$ .
2. Compute the bootstrapped variance
       $s_j^2 = \frac{1}{B}\sum_{b=1}^{B}(Z_{ji}^* - \bar{Z}_{j\cdot})^2$.  where   $\bar{Z}_{j\cdot} = \frac{1}{B}\sum_{b=1}^{B}\widehat{Z}_j^*$
3. Output the resulted $s_j^2$.
```

Figure 2: The bootstrap method to calculate the s_j^2

This bootstrap version works for both local and global rodeo algorithms, thus provides a more general framework. We expect that similar analytic results will hold. However, bootstrap needs more computation. In cases that the analytic form of the variance is hard to evaluate, like the local likelihood rodeo and the semiparametric rodeo, this method applies.

4.2 Reverse Rodeo

The previous rodeo algorithms use a sequence of decreasing bandwidths and estimates the optimal value by a sequence of hypothesis testing. On the contrary, we could begin from a very small bandwidth, and use a sequence of increasing bandwidths to estimate the optimal value. This reversed version does not share the same theoretical property as before, but it's useful in some special cases (i.e. many dimensions need a small bandwidths, while only a few need large bandwidths). More details will be given in an image processing experiment in the next section.

5 Examples

In this section, we applied the rodeo algorithm on both synthetic and real data, including one-dimensional, two-dimensional, high-dimensional and very high-dimensional examples to investigate how it performs in various conditions. For the purpose of evaluating the algorithm performance quantitatively, we need some criterion to measure the distance between the estimated density function with the true density. For this, we use the Hellinger distance, defined as

$$D(\widehat{f}\|f) = \int \left(\sqrt{\widehat{f}(x)} - \sqrt{f(x)} \right)^2 dx = 2 - 2 \int f(x) \sqrt{\frac{\widehat{f}(x)}{f(x)}} dx \tag{25}$$

Assuming we have m evaluation points, the hellinger distance could be numerically calculated by the Monte Carlo integration

$$D(\widehat{f}\|f) \approx 2 - \frac{2}{m} \sum_{i=1}^{m} \sqrt{\frac{\widehat{f}_H(X_i)}{f(X_i)}} \tag{26}$$

Since this measure utilizes the property that $f(x)$ is a density function, it's expected to be numerically more stable than the commonly used Kullback-Leibler (KL) divergence as a loss function for evaluating the discrepancy between two density functions. In the following, we first use the simulated data, about which we have known the true distribution function, to investigate the algorithm performance. Then our algorithm is also applied on some real data for analysis and comparison. In the following experiments, if not state explicitly, the data is always rescaled into a d-dimensional cube $[0,1]^d$, a product Gaussian kernels are used, the default parameters are $c_0 = 1$, $c_n = d \log n/n$, and $\beta = 0.9$.

5.1 One-dimensional Examples

First, we apply the rodeo algorithm on one dimensional examples. We have conducted a series of comparative study on a list of 15 "test densities" proposed by Marron and Wand [16], which are all normal mixtures representing many different types of challenges to density estimation methods. Our approach achieves a comparable performance as the built-in kernel density estimator with bandwidth selected by unbiased cross-validation (from the base library of R). Due to the space consideration, only the strongly skewed example is reported here, since it demonstrates the advantage of adaptive bandwidth selection for the local rodeo algorithm.

Example 1 (Strongly skewed density): This density is chosen to resemble the lognormal distribution, it distributes as

$$X \sim \sum_{i=0}^{7} \frac{1}{8} \mathcal{N} \left(3 \left((\frac{2}{3})^i - 1 \right), \left(\frac{2}{3} \right)^{2i} \right). \tag{27}$$

200 samples were generated from this distribution, The estimated density functions by the local rodeo, the global rodeo, and the built-in kernel density estimator with bandwidth chosen by unbiased cross validation are shown in figure 3. In which, the solid line is the true density function, the dashed line illustrates the estimated densities by different methods. The local rodeo works the best, this is because the true density function is highly skewed, the fixed bandwidth density estimator fails to fit the very smooth tail. The last subplot from firgure 3 illustrates the selected bandwidth for the local rodeo, it illustrates how smaller bandwidths are selected where the function is more rapidly varying. Figure 4 shows the distribution of the empirical Hellinger distances based on 100 simulations. The boxplots show that the local rodeo works the best, while the global rodeo and the unbiased cross-validation methods are comparable in this one dimensional example.

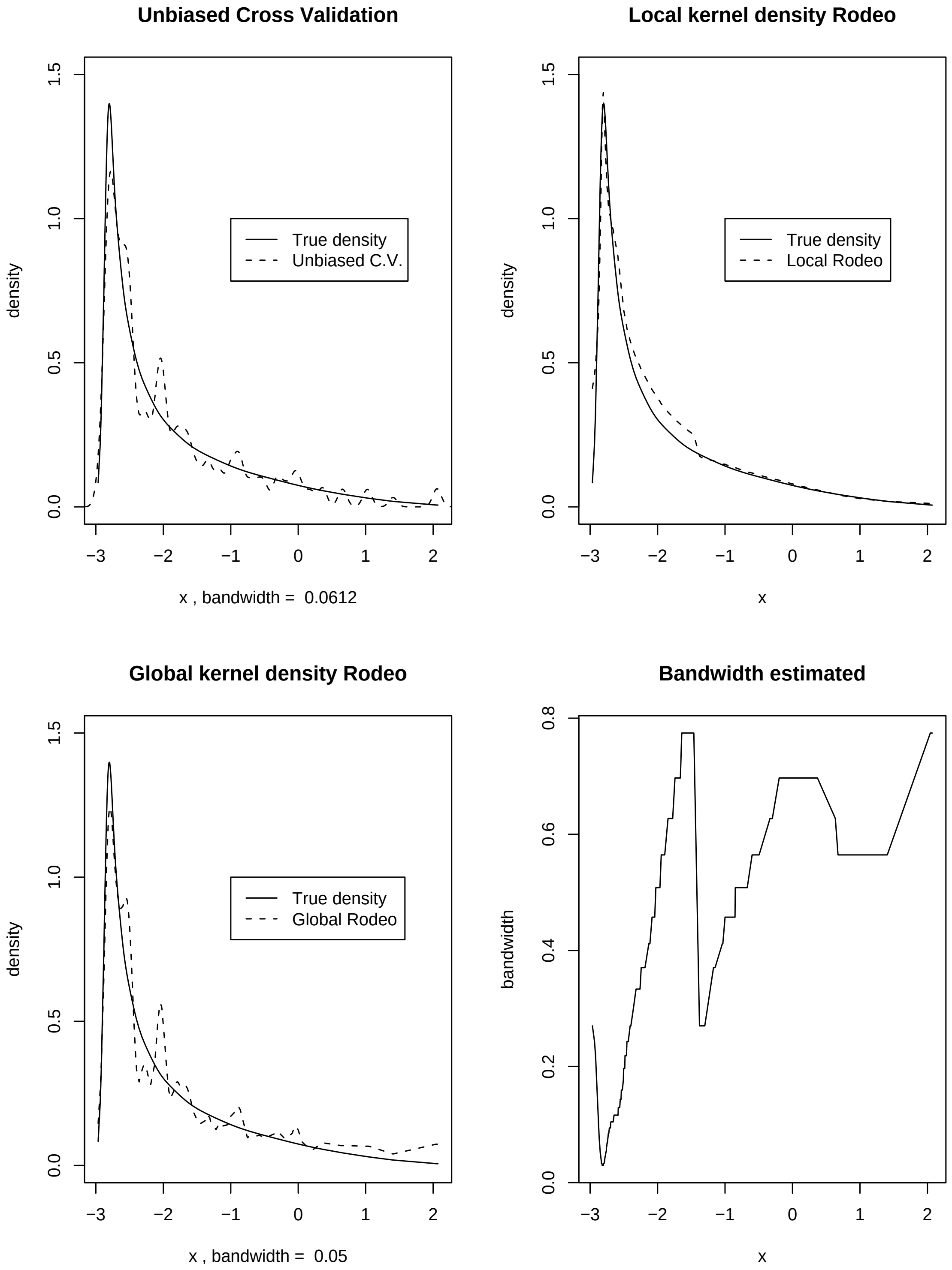


Figure 3: *Different versions of the algorithms run on the highly skewed unimodal example. The first three plots are results for the different estimators, the last one is the fitted bandwidths for the local rodeo.*

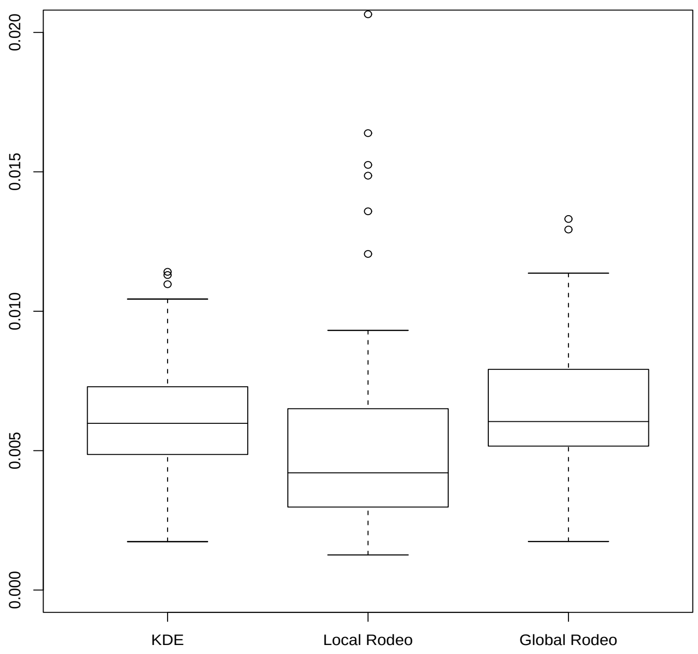


Figure 4: *Highly skewed unimodal distribution: The boxplots of the empirical Heillinger's losses on test samples of estimated densities by the three methods based on 100 simulations.*

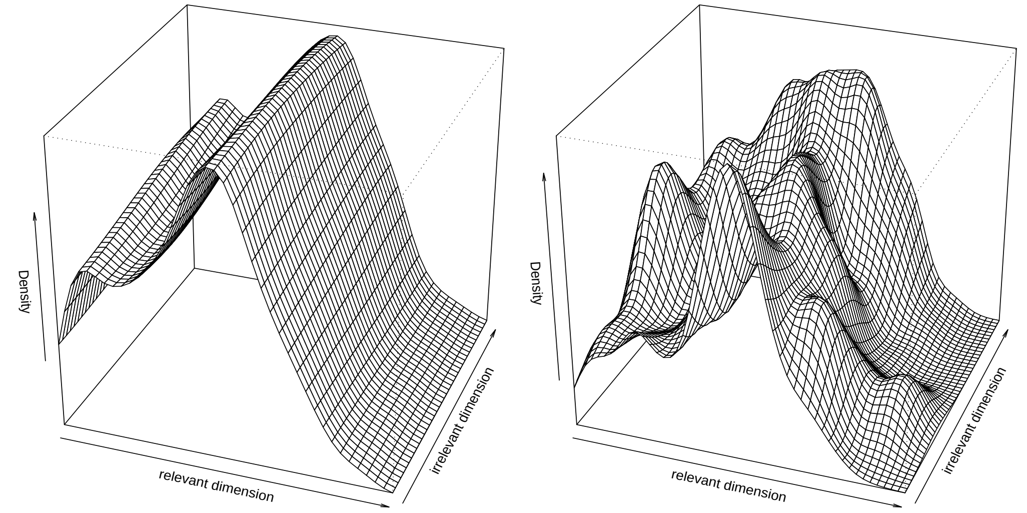
5.2 Two dimensional Examples

We also show some 2-dimensional examples, since they are easy to visualize. One uses a synthetic dataset, the other one uses some real data analyzed by the other authors. The density rodeo's performance is compared with a built-in method named KDE2d (from MASS package in R). The empirical results show that the rodeo algorithm works better than the built-in method on the synthetic data, where we know the ground truth. For the real-world dataset, where we do not know the underling density, our method achieves a very similar result as those of the previous authors.
Example 2:(Combined Beta distribution with the uniform distribution as irrelevant). We simulate a 2-dimensional dataset with $n = 500$ points. The two dimensions are independently generated as

$$X_1 \sim \frac{2}{3}\mathrm{Beta}(1,2) + \frac{1}{3}\mathrm{Beta}(10,10) \tag{28}$$

$$X_2 \sim \mathrm{Uniform}(0,1) \tag{29}$$

Figure 5 illustrates the perspective plots of the estimated density functions by the global rodeo and the built-in method KDE2d. From which, we see that the global rodeo fits the irrelevant uniform dimension perfectly, while KDE2d fails. For a quantitative comparison, we evaluated the empirical Hellinger distance between the estimated density and the true density, the global rodeo algorithm outperforms KDE2d uniformly on this example. For a qualitative comparison, figure 6 illustrates



(a) the rodeo estimation

(b) the KDE2d estimation

Figure 5: *Perspective plots of the estimated density functions by the global rodeo (left) and the R built-in method KDE2d (right) on a 2-dimensional synthetic data.*

the numerically integrated marginal distributions of the two estimators (not normalized). Even with an eye examination, we see that the rodeo's result is better than that of KDE2d, which is consistent with the previous observations.

Example 3:(Geyser data). For this example, a real dataset is used. Which is a version of the eruptions data from the "Old Faithful" geyser in Yellowstone National Park, Wyoming. This version comes from Azzalini and Bowman [17] and is of continuous measurement from August 1 to August 15, 1985. There are two variables with 299 observations altogether. The first variable ,"Duration", represents the numeric eruption time in minutes. The second variable, "waiting", represents the waiting time to next eruption. We apply the global rodeo algorithm on this dataset. The estimated density functions of the rodeo algorithm and the built-in KDE2d method (used by the original authors) are provided in the upper of figure 7. And lower two plots of figure 7 illustrates the corresponding contour plots. Based on a visual examination, our method achieves a very similar estimation as those provided by the previous authors who analyzed this data before.

5.3 High Dimensional Examples

Example 4: (High dimensional case) Figure 8 illustrates the output bandwidths from the local rodeo for a 30-dimensional synthetic dataset with $r = 5$ relevant dimensions ($n = 100$, with 30 trials). The relevant dimensions are generated as

$$X_i \sim \mathcal{N}(0.5, (0.02i)^2), \quad \text{for} \quad i = 1, ..., 5. \tag{30}$$

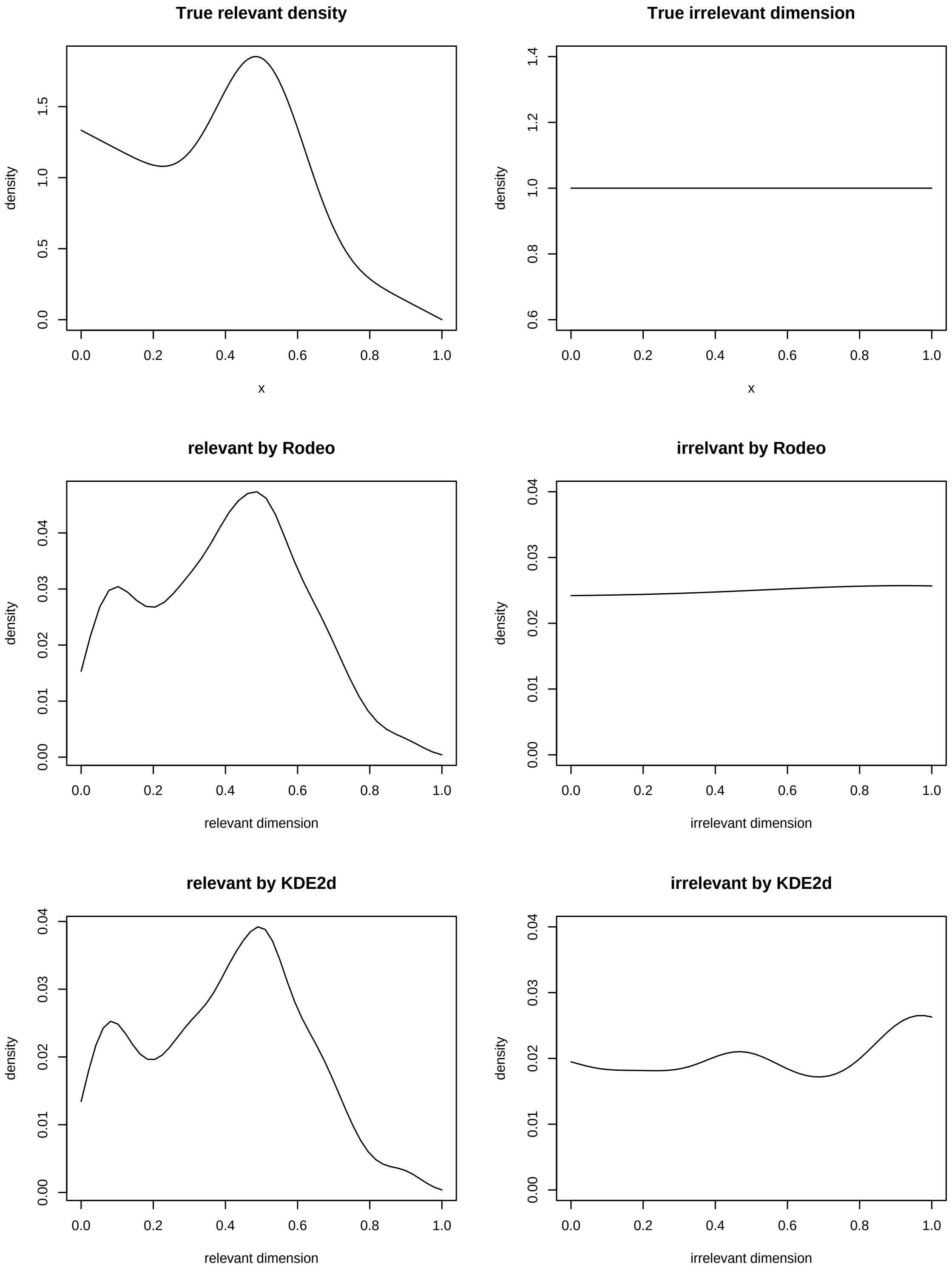
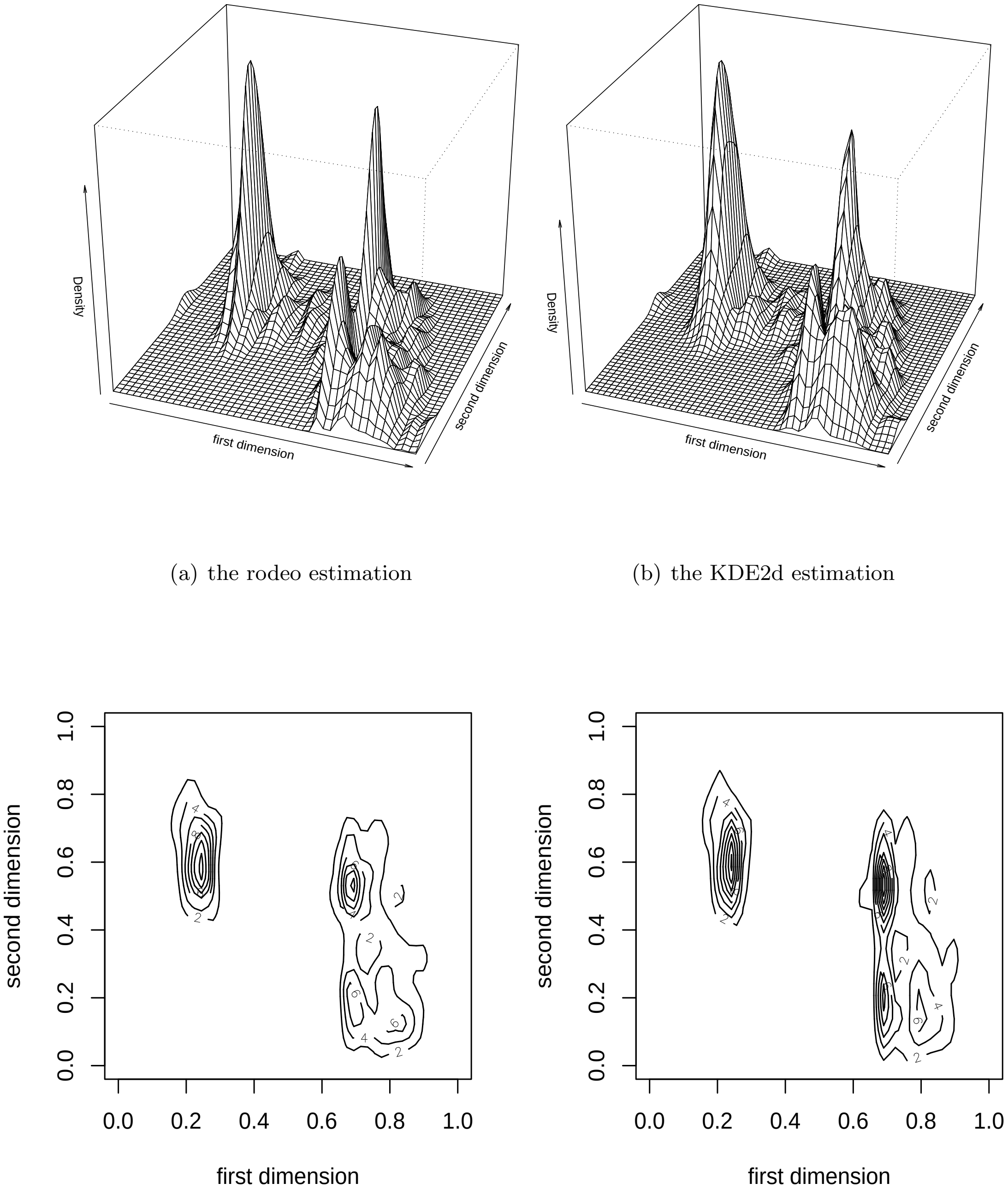


Figure 6: *Marginal distributions of the relevant and the irrelevant dimensions for example 2*



(a) the rodeo estimation

(b) the KDE2d estimation

Figure 7: *Upper: Perspective plots of the estimated density functions by the global rodeo (left) and the R built-in method KDE2d (right) on the geyser data. Lower: Contour plots of the result from the global rodeo (left) and KDE2d (right)*

while the irrelevant dimensions are generated as

$$X_i \sim \text{Uniform}(0,1), \quad \text{for} \quad i = 6, ..., 30. \tag{31}$$

The evaluation point is $x = (\frac{1}{2}, ..., \frac{1}{2})$. The boxplot illustrates the selected bandwidths out of 30 trials. The plot shows that the bandwidths of the relevant dimensions shrink towards zero, while the bandwidths of the irrelevant dimensions remain large, indicates that the algorithm's performance is consistent with our analysis. Also, from the bandwidth plot, we see that, for the relevant dimensions, the smaller the variance is, the smaller the estimated bandwidth will be.

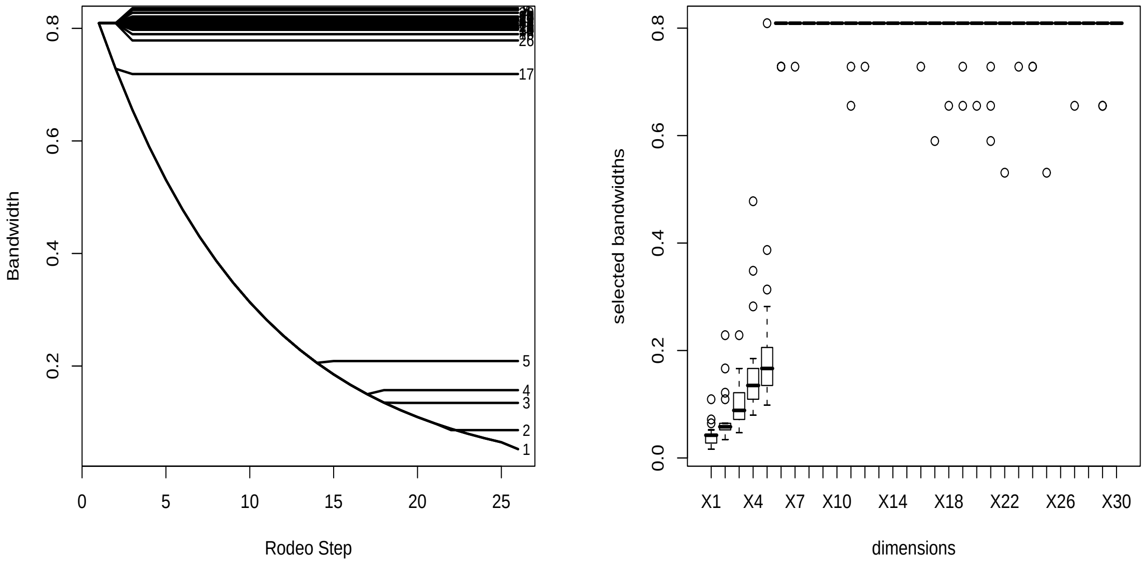


Figure 8: *the bandwidth output by the local density rodeo for a 30-dimensional synthetic dataset (Left) and its boxplot for 30 trials. (Right)*

Example 5: (Image processing). Here we apply the reverse local rodeo on image data. The results are shown in figure 9. The algorithm was run on 1100 grayscale images of digital letter 2s, each with $256 = 16 \times 16$ pixels with some unknown background noise; thus this is a 256-dimensional density estimation problem. An evaluation point is shown in the upper left subplot of figure 9, and the bandwidths output by the rodeo algorithm is shown in the upper right subplot. The estimated bandwidth plots in different rodeo steps (step 10,20,40,60,and 100) are shown in the lower series of plots$--$ smaller bandwidths have darker colors, the pixels with larger bandwidth are more informative than those with smaller bandwidths. This is a good example to illustrate the usefulness of the reverse rodeo. For the image data, many background pixels have a density close to point mass, which will pin down the bandwidth to a very small value. The reverse rodeo starts from a small bandwidths, which is more efficient than the original rodeo and is expected to be numerically more stable. Figure 9 visualizes the evolution of the bandwidths and could be viewed as a dynamic process for feature selection $--$ the earlier a dimension's bandwidth increases, the more informative it is. The reverse rodeo algorithm is quite efficient for this extremely high-dimensional problem. One interesting thing to note is, the early stages of the rodeo reveal that some of the 2s in the data have looped bottoms, while some have straight bottoms; the evaluation point does not have such a loop. This might because in the original dataset, some 2s have this loop while the others not. The density rodeo algorithm could discover these kind of characteristics automatically.

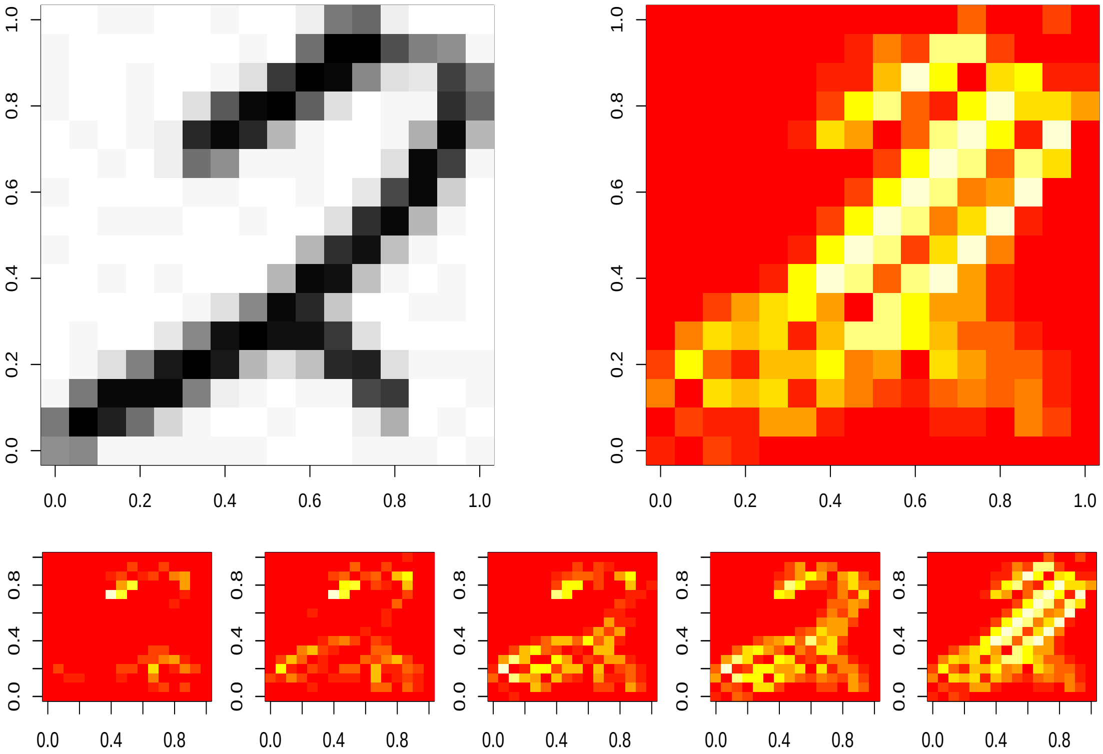


Figure 9: *the image processing example: the upper plots are the evaluation digit and the bandwidths output by the reverse rodeo. The lower subplots illustrate a series of bandwidth plots sampled at different rodeo steps: 10, 20, 40, 60, and 100*

5.4 Using Other Baseline Densities

Example 6: (Using normal distributions as the irrelevant dimensions) Figure 10 illustrates the output bandwidths from the semiparametric rodeo (developed in section 4) for both 15-dimensional and 20-dimensional synthetic datasets with $r = 5$ relevant dimensions ($n = 1000$). When using normal distributions as irrelevant dimensions, the relevant dimensions are generated as

$$X_i \sim \text{Uniform}(0, 1), \quad \text{for} \quad i = 1, ..., 5. \tag{32}$$

while the irrelevant dimensions are generated as

$$X_i \sim \mathcal{N}(0.5, (0.05i)^2), \quad \text{for} \quad i = 6, ..., d. \tag{33}$$

The evaluation point is $x = (\frac{1}{2}, ..., \frac{1}{2})$. Even when normal distributions are used as irrelevant dimensions, the result is similar as before, showing that the bandwidths of the relevant dimensions shrink toward zero, while the bandwidths of the irrelevant dimensions remain large, this is just what we expected.

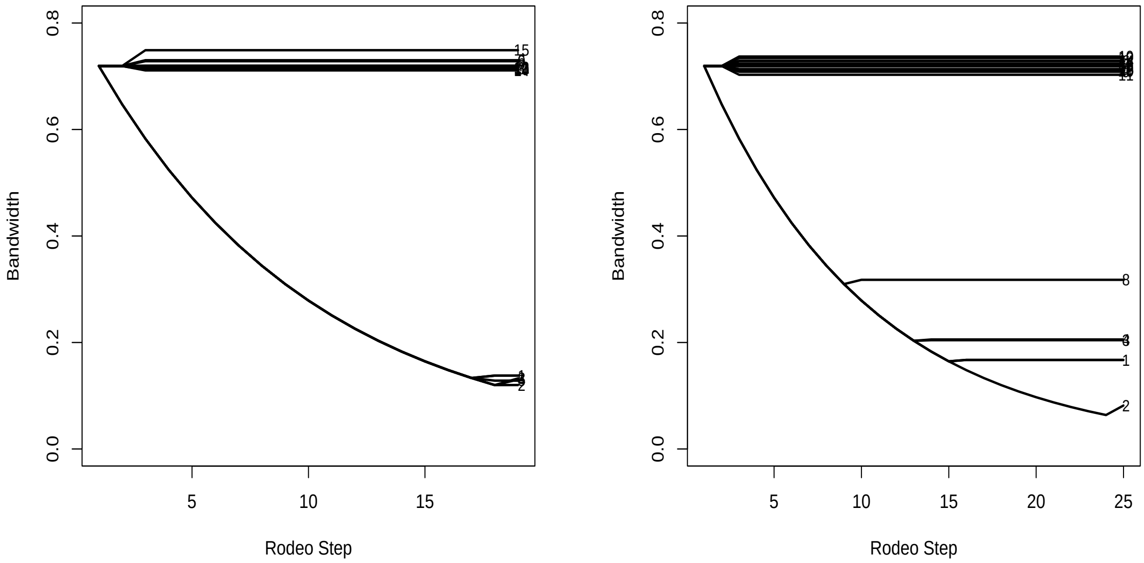


Figure 10: *the bandwidth output by the local semiparametric rodeo for a 15-dimensional synthetic dataset (Left) and a 20-dimensional synthetic dataset (Right). Using Gaussian distribution as the irrelevant dimensions*

Example 7: (The semiparametric density estimator for one dimensional problem) For the illustration purpose, we also applied the semiparametric rodeo algorithm on a dimensional example. We simulated 1000 one-dimensional data points with $X_i \sim \text{Uniform}(0, 1)$. With $\beta = 0.9$, the results of the semiparametric rodeo algorithm are shown in figure 11. The first plot shows the true density function, the second plot is the estimated density function, the lower left plot illustrates the estimated bandwidths at different evaluation points, the last one is the estimated density function by the kernel density estimator with bandwidth selected by unbiased cross validation. Based on a visual examination of the results, we see that the density function estimated by the semiparametric rodeo is quite similar to that estimated by the kernel density estimator with unbiased cross validation. However, the selected bandwidths are quite small in this case (≈ 0.015). Since the true density is uniform, smaller bandwidths are needed to correct the assumed normal density.

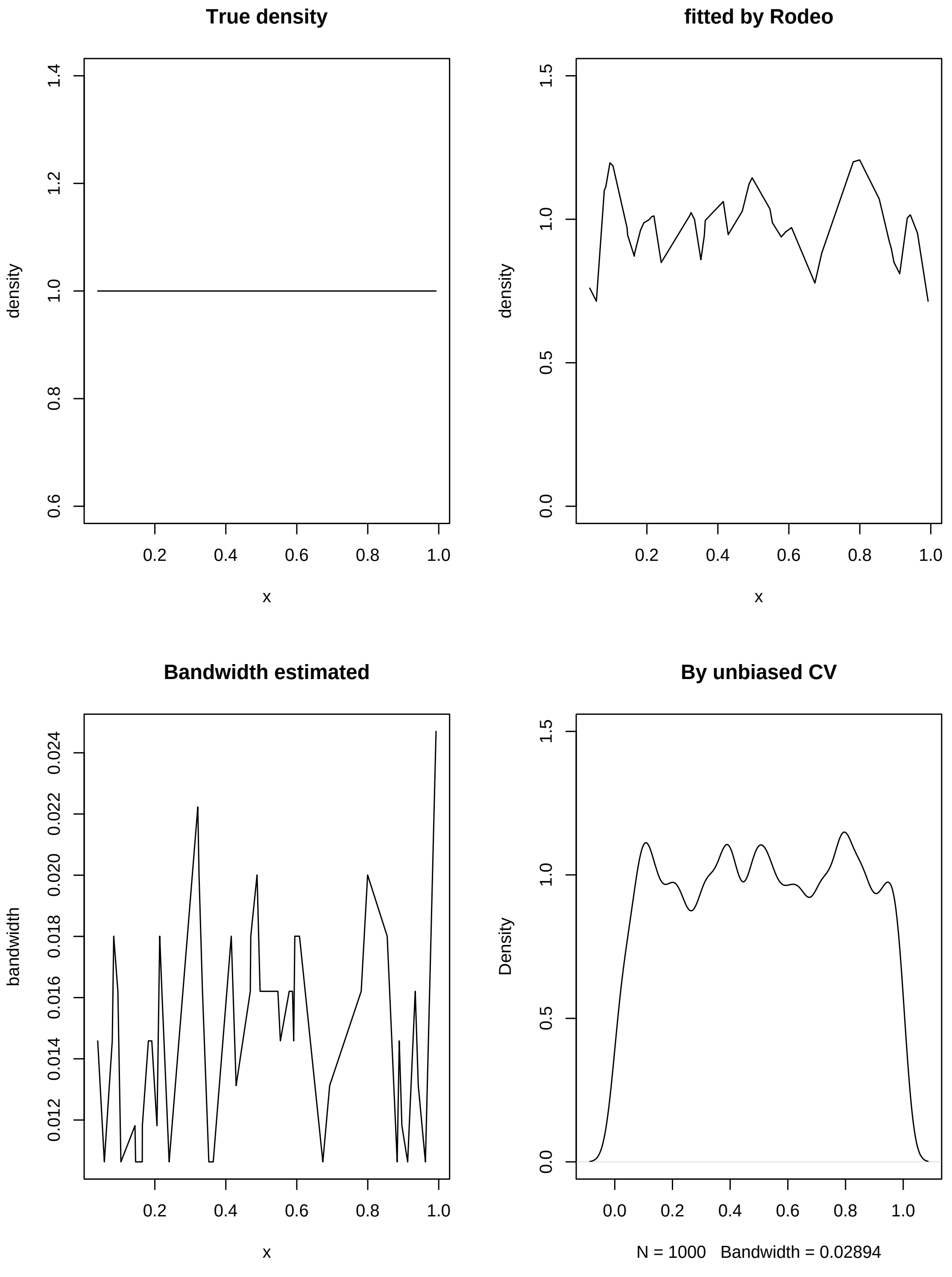


Figure 11: *Results for fitting the uniform distribution with the semiparametric rodeo. The first plot shows the true density , the second plot is the estimated density, the lower left plot illustrates the estimated bandwidths at different evaluation points, the last one is the estimated density function by the KDE with bandwidth selected by cross validation*

6 Asymptotic Properties

Here we show the asymptotic properties of the resulting estimator when assuming the baseline component $b(x)$ is a very smooth function. Our main theoretical results characterize the asymptotic running time, selected bandwidths and the risk of the resulting estimator. We assume that the underlying density function f has continuous second order derivatives in a neighborhood of x. For convenience of notation, the dimensions 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 $Y_n = O(b_n a_n)$ where b_n is logarithmic in n. As noted earlier, we write $a_n = \Omega(b_n)$ if $\liminf_n \left|\frac{a_n}{b_n}\right| > 0$; similarly $a_n = \widetilde{\Omega}(b_n)$ if $a_n = \Omega(b_n c_n)$ where c_n is logarithmic in n. Also, let $\mathcal{H}_f(x)$ denote the Hessian matrix of $f(x)$, let $h_j^{(s)}$ denote the j^{th} bandwidth at step s and denote the bandwidth matrix by $H^{(s)} = \mathrm{diag}(h_1^{(s)}, ..., h_d^{(s)})$. In the following, we assume that the data lines in a unit cube $[0,1]^d$.

We list the assumptions needed to establish the main result.

Assumption 1 (**A1**) *Kernel assumption: assuming that $\mathcal{K}$ is a bounded symmetric kernel, s.t. $\int \mathcal{K}(u)du = 1$, $\int u\mathcal{K}(u)du = 0_d$ while $\mathcal{K}_H(\cdot) = \frac{1}{\det(H)}\mathcal{K}(H^{-1}\cdot)$ represents the kernel with bandwidth matrix $H = diag(h_1, ..., h_d)$. then*

$$\int uu^T\mathcal{K}(u)du = v_2 I_d \text{ and } v_2 < \infty \tag{34}$$

$$\int \mathcal{K}^2(u)du = R(\mathcal{K}) < \infty \tag{35}$$

$$\textit{There exists some } C_{\mathcal{K}} \textit{ and } C_d < \infty \;\; \textit{s.t.} \;\; \sup_u |\mathcal{K}(u)| < C_{\mathcal{K}} \textit{ and } \sup_u \left|\frac{d\log K(u)}{du}\right| < C_d \tag{36}$$

Assumption 2 (**A2**) *Initial bandwdith assumption: Let $h_j^{(0)}$ denotes the initial bandwidth for the j-th dimension. Then,*

$$h_j^{(0)} = \frac{c_0}{\log\log n} \text{ for } (j = 1, ..., d). \tag{37}$$

Assumption 3 (**A3**) *Sparsity assumption: Assuming that $f(x)$ could be factorized into two components, $f(x) \propto g(x_1, ..., x_r)b(x)$, where $b_{jj}(x) = 0$ for $j = 1, ..., d$.*

Assumption 4 (**A4**) *Hessian assumption: Let $\mathcal{H}_R(x)$ denotes the Hessian matrix of all the relevant dimensions $j \leq r$. $\mathrm{diag}(\mathcal{H}_R(x))$ is a continuous vector and*

$$\int tr(\mathcal{H}_R^T(u)\mathcal{H}_R(u))du < \infty \tag{38}$$

$$\liminf_n \min_{1\leq j\leq r} |f_{jj}(x)| > 0 \tag{39}$$

Lemma 1 *Assume* A1 − A4*. Suppose that x is interior to the support of f and let $\mathcal{H}_R(x)$ denote the Hessian matrix of all the relevant dimensions $j \leq r$. Then, over different steps in the algorithm and over j, we have*

$$\mathbf{E}\widehat{f}_{H^{(s)}}(x) = f(x) + \frac{1}{2} v_2 tr((H^{(s)})^T \mathcal{H}_R^{(s)}(x) H) + o_P(tr((H^{(s)})^T H^{(s)})) \tag{40}$$

and

$$\mathbf{Var}(\widehat{f}_{H^{(s)}}(x)) = \frac{1}{n \det(H^{(s)})} R(\mathcal{K}) f(x) + o_P\left(\frac{1}{n \det(H^{(s)})}\right). \tag{41}$$

where v_2 and $R(\mathcal{K})$ are as defined in A1.

Lemma 2 *Suppose the kernel $\mathcal{K}_H$ is defined as in A1. Given a positive constant $\beta < 1$ and an increasing sequence of constants $t_n = \frac{1}{4+r} \log_{1/\beta}(nb_n)$, where $b_n = \widetilde{O}(1)$. Define the sets of bandwidth matrices*

$$\mathcal{H}_n = \{H^{(s)} : H^{(s)} = H^{(0)}\beta^s \text{ for all the nonnegative integer } s \text{ such that } s \leq t_n\} \tag{42}$$

Define

$$M_n(x) = \frac{\left(\widehat{f}_H(x) - \mathbf{E}\widehat{f}_H(x)\right)}{\sqrt{\mathbf{Var}(\widehat{f}_H(x))}} \tag{43}$$

Then

$$\sup_{H \in \mathcal{H}_n} \sup_z |\mathbf{P}\left(M_n(x) \leq z\right) - \Phi(z)| \longrightarrow 0. \tag{44}$$

Lemma 3 *Under assumptions* A1 − A4*, suppose that x is interior to the support of f and K is a product kernel with bandwidth matrix $H^{(s)} = diag(h_1^{(s)}, ..., h_d^{(s)})$. Then*

$$\mu_j^{(s)} = \frac{\partial}{\partial h_j^{(s)}} \mathbf{E}[\widehat{f}_{H^{(s)}}(x) - f(x)] = o_P(h_j^{(s)}) \text{ for all } j \in R^c \tag{45}$$

For $j \in R$ we have

$$\mu_j^{(s)} = \frac{\partial}{\partial h_j^{(s)}} \mathbf{E}[\widehat{f}_{H^{(s)}}(x) - f(x)] = h_j^{(s)} v_2 f_{jj}(x) + o_P(h_j^{(s)}). \tag{46}$$

Thus, for any integer $s > 0$, $h_s = h_0\beta^s$, each $j > r$ satisfies $\mu_j^{(s)} = o_P(h_j^{(s)}) = o_P(h_j^{(0)})$.

Lemma 4 *Define*

$$C = \frac{R(\mathcal{K}) f(x)}{4} \tag{47}$$

then, if $h_j^{(0)}$ is defined as in A2*.*

$$(s_j^{(s)})^2 = \mathbf{Var}(Z_j^{(s)}) = \frac{C}{n(h_j^{(s)})^2} \left(\prod_{k=1}^{d} \frac{1}{h_k^{(s)}}\right)(1 + o_P(1)) \tag{48}$$

Lemma 5 *Under assumptions* A1 − A4. *Assume* $Z_j = \frac{1}{n}\sum_{i=1}^{n} Z_{ji}$ *is defined as in equation(4), given a positive constant* $\beta < 1$ *and an increasing sequence of constants* $t_n = \frac{1}{4+r}\log_{1/\beta}(nb_n)$*, where* $b_n = \widetilde{O}(1)$*. Define the sets of bandwidth matrices*

$$\mathcal{H}_n = \{H^{(s)} : H^{(s)} = H^{(0)}\beta^s \textit{ for all the nonnegative integer } s \textit{ such that } s \le t_n\} \tag{49}$$

Then

$$\sup_{H\in\mathcal{H}_n} \sup_{z} \left| \mathbf{P}\left(\frac{Z_j - \mathbf{E}Z_j}{\sqrt{\mathbf{Var}(Z_j)}} \le z \right) - \Phi(z) \right| \longrightarrow 0. \tag{50}$$

Lemma 6 *Let* $Z \sim \mathcal{N}(\mu, \sigma^2)$*. If* $\lambda > 2\mu$ *and* $\lambda^2 > 2\sigma^2$ *then*

$$\mathbf{P}(|Z| > \lambda) \le \frac{5\lambda}{\sigma} \exp\left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \tag{51}$$

Moreover, if $\lambda \ge 5\sigma$ *then*

$$\mathbf{P}(|Z| > \lambda) \le \exp\left\{ -\frac{\lambda^2}{16\sigma^2} \right\}. \tag{52}$$

The proof of this Lemma could be found in Lafferty & Wasserman 2006. [18].

Theorem 1 *Suppose* (A1) –(A4) *hold. In addition, suppose that* $A_{min} = \min_{j\le r} |f_{jj}(x)| = \widetilde{\Omega}(1)$ *and* $A_{max} = \max_{j\le r} |f_{jj}(x)| = \widetilde{O}(1)$*. Then, the number of iterations* T_n *until the Rodeo stops satisfies*

$$\mathbf{P}\left(\frac{1}{4+r}\log_{1/\beta}(na_n) \le T_n \le \frac{1}{4+r}\log_{1/\beta}(nb_n) \right) \longrightarrow 1 \tag{53}$$

where $a_n = \widetilde{\Omega}(1)$ *and* $b_n = \widetilde{O}(1)$*. Moreover, the algorithm outputs bandwidths* $H^* = diag(h_1^*, ..., h_d^*)$ *that satisfies*

$$\mathbf{P}\left(h_j^* = h_j^{(0)} \text{ for all } j > r \right) \longrightarrow 1 \tag{54}$$

Also, we have

$$\mathbf{P}\left(h_j^{(0)}(nb_n)^{-1/(4+r)} \le h_j^* \le h_j^{(0)}(na_n)^{-1/(4+r)} \text{ for all } j \le r \right) \longrightarrow 1 \tag{55}$$

assuming that $h_j^{(0)}$ *is defined as in* A2.

Theorem 2 *Under the same condition of theorem 1, the risk* $\mathcal{R}_{h^*}$ *of the rodeo density estimator satisfies*

$$\mathcal{R}_{H^*} = \mathbf{E}\int \left(\widehat{f}_{H*}(x) - f(x) \right)^2 dx = \widetilde{O}_P\left(n^{-4/(4+r)} \right) \tag{56}$$

Proof: Since the integrand is nonnegative, the order of integration and expectation can be reversed, so that

$$\mathcal{R}_{H^*} = \mathbf{E}\int \left(\widehat{f}_{H*}(x) - f(x)\right)^2 dx = \int \mathbf{E}\left(\widehat{f}_{H*}(x) - f(x)\right)^2 dx \tag{57}$$

$$= \int \mathbf{Bias}^2\left(\widehat{f}_{H^*}(x)\right) dx + \int \mathbf{Var}\left(\widehat{f}_{H^*}(x)\right) dx \tag{58}$$

Given the bandwidths in expression(54) and expression(55), we have that the squared bias is given by

$$\int \mathbf{Bias}^2\left(\widehat{f}_{H^*}(x)\right) dx = \int \left(\sum_{j\le r} v_2 f_{jj}(x) h_j^{*2}\right)^2 dx + o_P(tr(H^{*T}H^*)) \tag{59}$$

$$= \int \sum_{i,j\le r} v_2^2 f_{ii}(x) f_{jj}(x) h_i^{*2} h_j^{*2} dx + o_P(tr(H^{*T}H^*)) \tag{60}$$

$$= \widetilde{O}_P(n^{-4/(4+r)}) \tag{61}$$

by Theorem 1. Similarly, by lemma 4, we calculate the variance as

$$\int \mathbf{Var}\left(\widehat{f}_{H^*}(x)\right) dx = \int \frac{1}{n}\prod_i \frac{1}{h_i^*} R(\mathcal{K}) f(x)(1 + o_P(1))dx \tag{62}$$

$$= \widetilde{O}_P(n^{-1+r/(4+r)}) \tag{63}$$

$$= \widetilde{O}_P(n^{-4/(4+r)}) \tag{64}$$

The result follows from the bias-variance decomposition.■

This result shows that the optimal rates of convergence is obtained up to a logarithmic factor.

7 Conclusions

This work is mainly purposed to illustrate the generality of the rodeo framework. Under some suitably-defined sparsity condition, the previously developed nonparametric regression framework is easily adapted to perform high-dimensional density estimation. The resulting method is both computationally efficient and theoretically soundable. Empirical results show that our method is better than the built-in methods in many cases.

Current assumption requires the underlying density to be factorized into two components. Another interesting assumption is to assume that the observed high-dimensional data are lying on a low-dimensional smooth manifold. A recent result from Bickel et al. [19] shows that local polynomial regression can adapt to the local manifold structure in the sense that it achieves the optimal convergence rate. When assuming all dimensions use the same bandwidth h, they formalize an asymptotic irrelevance condition as

$$\exists\epsilon(0<\epsilon<1), \text{s.t.}\mathbf{E}\left[\mathcal{K}^{\gamma}\left(\frac{X-x}{h}\right) w(X)\mathbb{1}\left(X \in \left(\mathcal{B}^D_{x,h^{1-\epsilon}} \cap \mathcal{X}\right)^c\right)\right] = o(h^{d+2}) \tag{65}$$

for $\gamma = 1, 2$ and $|w(x)| \le M(1 + |x|^2)$. Under this kind of assumptions, it's interesting to design a blockwised rodeo algorithm(i.e. only on a local portion of the data, we estimate the bandwidth) which can also adapt to the local manifold structure and achieves a better risk.

A Derivation of the Semiparametric Rodeo Estimator

When assuming a Gaussian kernel and a Gaussian baseline distribution, the semiparametric estimator is defined as

$$\bar{f}_H(x) = \frac{\widehat{b}(x) \sum_{i=1}^{n} \mathcal{K}_H(X_i - x)}{n \int \mathcal{K}_H(u-x)\widehat{b}(u)du} \tag{66}$$

Here, we ignore the hat notation for both μ and Σ. Assuming that

$$\begin{aligned} K_h(u) &\sim \mathcal{N}(0, H), \quad H = diag(h_1^2, h_2^2, ..., h_d^2) && (67) \\ \widehat{b}(u) &= \mathcal{N}(\mu, \Sigma) && (68) \end{aligned}$$

We get that the denominator is $\propto \mathcal{N}(\mu, H+\Sigma)$, therefore

$$n\int_{-\infty}^{\infty} K_h(u-x)\widehat{b}(u)du = \frac{n}{\sqrt{2\pi|H+\Sigma|}} \exp\left\{-\frac{(x-\mu)^T(H+\Sigma)^{-1}(x-\mu)}{2}\right\} \tag{69}$$

Thus

$$\begin{aligned} \frac{\widehat{b}(x)}{n\int_{-\infty}^{\infty} K_h(u-x)\widehat{b}(u)du} &= \frac{\frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left\{-\frac{(x-\mu)^T(\Sigma)^{-1}(x-\mu)}{2}\right\}}{\frac{n}{\sqrt{2\pi|H+\Sigma|}} \exp\left\{-\frac{(x-\mu)^T(H+\Sigma)^{-1}(x-\mu)}{2}\right\}} && (70) \\ &= \frac{1}{n}\sqrt{\frac{|H+\Sigma|}{|\Sigma|}} \exp\left\{-\frac{(x-\mu)^T\left(\Sigma^{-1}-(H+\Sigma)^{-1}\right)(x-\mu)}{2}\right\} && (71) \end{aligned}$$

The final estimator is

$$\bar{f}_H(x) = \widehat{f}_H(x)\sqrt{\frac{|H+\Sigma|}{|\Sigma|}} \exp\left\{-\frac{(x-\mu)^T\left(\Sigma^{-1}-(H+\Sigma)^{-1}\right)(x-\mu)}{2}\right\} \tag{72}$$

where $\widehat{f}_H(x)$ is the standard kernel density estimator defined in equation (3).

B Proofs

For the convenience of notation, we suppress the superscripts unless necessary.

Proof of Lemma 1. For the expectation term,

$$
\begin{aligned}
\mathbf{E}\widehat{f}_H(x) &= \mathbf{E}\frac{1}{n\det(H)}\sum_{i=1}^{n}\mathcal{K}(H^{-1}(x-X_i)) && (73)\\
&= \frac{1}{\det(H)}\int\mathcal{K}(H^{-1}(u-x))f(u)du && (74)\\
&= \int\mathcal{K}(u)f(x+Hu)du && (75)\\
&= \int\mathcal{K}(u)\{f(x)+u^TH^T\nabla f(x)+\frac{1}{2}u^TH^T\mathcal{H}_f(x)Hu+o_P(tr(u^TH^THu))\}du && (76)\\
&= f(x)+\frac{1}{2}\upsilon_2 tr(H^T\mathcal{H}_f(x)H)+o_P(tr(H^TH)) && (77)\\
&= f(x)+\frac{1}{2}\upsilon_2 tr(H^T\mathcal{H}_R(x)H)+o_P(tr(H^TH)) && (78)
\end{aligned}
$$

Equation (78) follows from A3. While

$$
\begin{aligned}
\mathbf{Var}(\widehat{f}_H(x)) &= \mathbf{Var}\left(\frac{1}{n\det(H)}\sum_{i=1}^{n}\mathcal{K}(H^{-1}(x-X_i))\right) && (79)\\
&= \frac{1}{n\det(H)^2}\mathbf{Var}\left(\mathcal{K}(H^{-1}(x-X_i))\right) && (80)\\
&= \frac{1}{n\det(H)^2}\mathbf{E}\{\mathcal{K}^2(H^{-1}(x-X_i))\}-\frac{1}{n\det(H)^2}\mathbf{E}^2\{\mathcal{K}(H^{-1}(x-X_i))\} && (81)\\
&= \frac{1}{n\det(H)^2}\int\{\mathcal{K}(H^{-1}(u-x))\}^2f(u)du-\frac{1}{n}\mathbf{E}^2\{\widehat{f}_H(x)\} && (82)\\
&= \frac{1}{n\det(H)}\int\{\mathcal{K}(u)\}^2f(u+Hu)du-\frac{1}{n}\mathbf{E}^2\{\widehat{f}_H(x)\} && (83)\\
&= \frac{1}{n\det(H)}R(\mathcal{K})f(x)+o_P\left(\frac{1}{n\det(H)}\right) && (84)
\end{aligned}
$$

The first equality follows from the fact that all the X_i's are i.i.d. The last equality follows by a Taylor's expansion. ■

Proof of Lemma 2. Define $\widehat{f}_H(x)=\frac{1}{n}\sum_{i=1}^{n}J_i(x)$, where $J_1(x),J_2(x),...,J_n(x)$ are i.i.d distributed, and

$$
M_n(x)=\frac{\left(\widehat{f}_H(x)-\mathbf{E}\widehat{f}_H(x)\right)}{\sqrt{\mathbf{Var}(\widehat{f}_H(x))}} \qquad (85)
$$

From Berry-Esseen bound, for each fixed H, we get that

$$\begin{aligned}
\sup_z |\mathbf{P}\left(M_n(x) \le z\right) - \Phi(z)| &\le \frac{33}{4} \frac{\mathbf{E}|J_1(x) - \mathbf{E}J_1(x)|^3}{\sqrt{n}\mathbf{Var}^{3/2}(J_1(x))} && (86)\\
&\le \frac{33}{4} \frac{\mathbf{E}\left(|J_1(x)| + |\mathbf{E}J_1(x)|\right)^3}{\sqrt{n}\mathbf{Var}^{3/2}(J_1(x))} && (87)\\
&\le \frac{66|J_1(x)|^3}{\sqrt{n}\mathbf{Var}^{3/2}(J_1(x))} && (88)\\
&\le \frac{66\prod_{k=1}^d \frac{1}{h_k^3} C_{\mathcal{K}}^3}{\sqrt{n}\prod_{k=1}^d \frac{1}{h_k^{3/2}}(4C)^3} && (89)\\
&= \frac{66(C_{\mathcal{K}}/4C)^3}{\sqrt{n}\prod_{k=1}^d h_k^{3/2}} && (90)
\end{aligned}$$

Where $C_{\mathcal{K}}$ and C are defined in assumption A4 and lemma1 respectively. Based on this, the supreme over all the bandwidths $H \in \mathcal{H}_n$ satisfies

$$\begin{aligned}
\sup_{H\in\mathcal{H}_n} \sup_z |\mathbf{P}\left(M_n(x) \le z\right) - \Phi(z)| &\le \frac{66(C_{\mathcal{K}}/4C)^3}{\sqrt{n}\left(\prod_{k=1}^d h_k^{(0)}\beta^{t_n}\right)^{3/2}} && (91)\\
&= O\left(\frac{b_n^{1/(4+r)}\left(\log\log n\right)^{3d/2}}{n^{(r-2)/(8r+2)}}\right) \longrightarrow 0 && (92)
\end{aligned}$$

The result follows directly. ■

Proof of Lemma 3. For $j \in R$, from lemma 2

$$\mathbf{E}\widehat{f_H}(x) - f(x) = \frac{1}{2}v_2 tr(H^T\mathcal{H}_R(x)H) + o_P(tr(H^TH)) \tag{93}$$

under some regularity conditions

$$\mu_j = \frac{\partial}{\partial h_j}\mathbf{E}[\widehat{f}_H(x) - f(x)] = h_j v_2 f_{jj}(x) + o_P(h_j). \tag{94}$$

For $j \in R^c$, the proof proceeds by equation (93), when $j \in R^c$, the corresponding elements in the Heissen $\mathcal{H}_f(x)$ will be 0, the result follows directly. ■

Proof of Lemma 4. Assuming that $\xi \sim \mathcal{N}(0,1)$. From lemma1, we could represents the kernel density estimator $\widehat{f}_H(x)$ as

$$\begin{aligned}
\widehat{f}_H(x) &= \mathbf{E}\widehat{f}_H(x) + \sqrt{\mathbf{Var}(\widehat{f}_H(x))} \times \xi && (95)\\
&= f(x) + \frac{1}{2}v_2 tr(H^T\mathcal{H}_f(x)H) + o_P(tr(H^TH)) + \sqrt{\mathbf{Var}(\widehat{f}_H(x))} \times \xi && (96)
\end{aligned}$$

Thus,

$$Z_j = \frac{\partial \widehat{f}_H(x)}{\partial h_j} + \frac{\partial}{\partial h_j}\left(\frac{1}{2}v_2 tr(H^T\mathcal{H}_R(x)H)\right) + \frac{\partial}{\partial h_j}\left(\sqrt{\mathbf{Var}(\widehat{f}_H(x))} \times \xi\right) \tag{97}$$

Since

$$
\begin{aligned}
\frac{\partial}{\partial h_j}\left(\sqrt{\mathbf{Var}(\widehat{f}_H(x))}\right) &= \frac{1}{2}\frac{1}{\sqrt{\mathbf{Var}(\widehat{f}_H(x))}}\frac{\partial}{\partial h_j}\left(\mathbf{Var}\left(\widehat{f}_H(x)\right)\right) & (98)\\
&= -\frac{1}{2}\frac{1}{\sqrt{\mathbf{Var}(\widehat{f}_H(x))}}\left(\frac{R(\mathcal{K})f(x)}{h_j n\det(H)}\right)\left(1+o_P\left(\frac{1}{n\det(H)h_j}\right)\right) & (99)\\
&= -\frac{1}{2}\sqrt{\frac{R(\mathcal{K})f(x)}{h_j^2 n\det(H)}}(1+o_P(\sqrt{h_j})) & (100)
\end{aligned}
$$

The second equality follows from lemma1, therefore

$$
s_j^2 = \mathbf{Var}(Z_j) = \frac{C}{nh_j^2}\left(\prod_{k=1}^{d}\frac{1}{h_k}\right)(1+o_P(1)).\blacksquare \qquad (101)
$$

Proof of Lemma 5. Since $Z_j = \frac{1}{n}\sum_{i=1}^{n} Z_{ji}$, and $Z_{j1}, Z_{j2}, ..., Z_{jn}$ are i.i.d distributed. Similar as in the proof of lemma 2 , from Berry-Esseen bound, for each fixed H, we get that

$$
\begin{aligned}
\sup_z\left|\mathbf{P}\left(\frac{Z_j-\mathbf{E}Z_j}{\sqrt{\mathbf{Var}(Z_j)}}\le z\right)-\Phi(z)\right| &\le \frac{33}{4}\frac{\mathbf{E}|Z_{j1}-\mathbf{E}Z_{j1}|^3}{\sqrt{n}\mathbf{Var}^{3/2}(Z_{j1})} & (102)\\
&\le \frac{66|Z_{j1}|^3}{\sqrt{n}\mathbf{Var}^{3/2}(Z_{j1})} & (103)\\
&\le \frac{66\frac{1}{h_j^9}\prod_{k=1}^{d}\frac{1}{h_k^3}C_{\mathcal{M}}^3}{\sqrt{n}\frac{1}{h_j^3}\prod_{k=1}^{d}\frac{1}{h_k^{3/2}}(C^{1/2})^3} & (104)\\
&= \frac{66(C_{\mathcal{M}}/\sqrt{C})^3}{\sqrt{n}h_j^6\prod_{k=1}^{d}h_k^{3/2}} & (105)
\end{aligned}
$$

Where $C_{\mathcal{M}}$ are evaluated from $C_{\mathcal{K}}$ and C_d in assumption A4 and C is defined in lemma 1. Based on the same reasoning as in lemma 2, the supreme over all the bandwidths $H \in \mathcal{H}_n$ satisfies

$$
\sup_{H\in\mathcal{H}_n}\sup_z\left|\mathbf{P}\left(\frac{Z_j-\mathbf{E}Z_j}{\sqrt{\mathbf{Var}(Z_j)}}\le z\right)-\Phi(z)\right| \longrightarrow 0. \qquad (106)
$$

$\blacksquare$

Proof of Lemma 6. Without loss of generality, assume $\mu > 0$. Then,

$$
\begin{aligned}
\mathbf{P}(|Z|>\lambda) &\le 2\mathbf{P}(Z>\lambda) & (107)\\
&= 2\mathbf{P}\left(\frac{Z-\mu}{\sigma}>\frac{\lambda-\mu}{\sigma}\right) & (108)\\
&\le \frac{2\sigma}{\lambda-\mu}\exp\left\{-\frac{(\lambda-\mu)^2}{2\sigma^2}\right\} \equiv B(\mu) & (109)
\end{aligned}
$$

Now $B(\mu) = B(0) + \mu B'(\widetilde{\mu})$ for some $0 \leq \widetilde{\mu} \leq \mu$ and

$$B'(\mu) = \frac{2\sigma}{\lambda - \mu} \exp\left\{ -\frac{(\lambda-\mu)^2}{2\sigma^2} \right\} \left(\frac{\lambda - \mu}{\sigma^2} + \frac{1}{\lambda - \mu} \right) \tag{110}$$

Hence

$$B'(\mu) \leq \frac{2\sigma}{\lambda - \mu} \exp\left\{ -\frac{(\lambda-\mu)^2}{2\sigma^2} \right\} \left(\frac{\lambda}{\sigma^2} + \frac{1}{\lambda - \mu} \right) \tag{111}$$

When $\lambda \geq 2\mu$, $1/(\lambda - \mu) \leq 2/\lambda$ and $(\lambda - \mu)^2 \geq \lambda^2/4$ so that if $\lambda^2 \geq 2\sigma^2$ then

$$B'(\mu) \leq \frac{4\sigma}{\lambda} \exp\left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \left(\frac{\lambda}{\sigma^2} + \frac{2}{\lambda} \right) \leq \frac{8}{\sigma} \exp\left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \tag{112}$$

Thus,

$$\mathbf{P}(|Z| > \lambda) \leq \frac{2\sigma}{\lambda} \exp\left\{ -\frac{\lambda^2}{2\sigma^2} \right\} + \frac{8\mu}{\sigma} \exp\left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \leq \frac{5\lambda}{\sigma} \exp\left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \tag{113}$$

The last statement follows since $5xe^{-x^2/8} \leq e^{-x^2/16}$ for all $x \geq 5$. ■

Proof of theorem 1. First, consider $j > r$. Let $V_t = \{j > r : h_j = h_0\beta^t\}$ be the set of irrelevant dimensions that are active at stage $t > 1$ of the algorithm. Define $v_j = \mathbf{Var}(Z_j)$, from lemma 3 and the algorithm in figure 1, for sufficiently large n, it's obvious that $\lambda_j \geq 2\mu_j$, $\lambda_j^2 \geq 2s_j^2$, and $\lambda \geq 5s_j$, and $v_j^2/s_j^2 = 1 + o(1)$ with probability tending to 1. Assuming $\widetilde{Z_j}$ is a normal random variable with the same mean and variance as Z_j. Then

$$\begin{aligned}
\mathbf{P}(|Z_j| > \lambda_j, \text{ for some } j \in V_t) &\leq \sum_{j \in V_t} \mathbf{P}(|Z_j| > \lambda_j) + o(1) &(114)\\
&= \sum_{j \in V_t} \Big(\mathbf{P}(|\widetilde{Z_j}| > \lambda_j) + \mathbf{P}(|Z_j| > \lambda_j) - \mathbf{P}(|\widetilde{Z_j}| > \lambda_j) \Big) + o(1) &(115)\\
&\leq d \exp\left\{ -\lambda_j^2/16v_j^2 \right\} + o(1) &(116)\\
&= d \exp\left\{ -\lambda_j^2(1 + o(1))/16s_j^2 \right\} + o(1) \longrightarrow 0 &(117)
\end{aligned}$$

Therefore, with probability tending to 1, $h_j = h_0$ for each $j > r$, meaning that the bandwidth for each irrelevant dimension is frozen in the first step in the algorithm.

Now consider $j \leq r$. By assumption A4 and lemma 3, for sufficiently large n, $\mu_j \geq ch_j|f_{jj}(x)|$ for some $c > 0$. Without loss of generality, assume that $ch_jf_{jj} > 0$. We claim that in iteration t of the algorithm, if

$$t \leq \frac{1}{4+r} \log_{1/\beta} \left(\frac{c^2 n A_{min}^2 h_0^{4+d}}{8C \log(nc_n)} \right) \tag{118}$$

then

$$\mathbf{P}(h_j = h_0\beta^t, \text{ for all } j \leq r) \longrightarrow 1. \tag{119}$$

To show this, first note that inequality (118) can be written as

$$\left(\frac{1}{\beta}\right)^{t(4+r)} \leq \frac{c^2 n A_{min}^2 h_0^{4+d}}{8C\log(nc_n)} \tag{120}$$

Except on an event of vanishing probability, we have shown above that

$$\prod_{j>r} \frac{1}{h_j} \leq \left(\frac{1}{h_0}\right)^{d-r} \tag{121}$$

So on the complement of this event, if each relevant dimension is active at step $s \leq t$, we have

$$\begin{aligned}
\frac{\lambda_j^2}{h_j^2} &= \frac{2s_j^2\log(nc_n)}{h_j^2} && (122)\\
&= \frac{2C\log(nc_n)}{nh_j^4}\prod_i \frac{1}{h_i} && (123)\\
&\leq \frac{2C\log(nc_n)}{nh_0^{4+d}}\left(\frac{1}{\beta}\right)^{(4+r)t} && (124)\\
&\leq \frac{c^2A_{min}^2}{4} && (125)\\
&\leq \frac{c^2f_{jj}(x)^2}{4} && (126)
\end{aligned}$$

which implies that

$$cf_{jj}(x)h_j \geq 2\lambda_j \tag{127}$$

and hence

$$\frac{cf_{jj}(x)h_j - \lambda_j}{s_j} \geq \frac{\lambda_j}{s_j} = \sqrt{2\log(nc_n)} \tag{128}$$

for each $j \le r$. Now,

$$\begin{aligned}
\mathbf{P}(\text{ rodeo halts }) &= \mathbf{P}(|Z_j| < \lambda_j \text{ for all } j \le r) && (129)\\
&\le \mathbf{P}(|Z_j| < \lambda_j \text{ for some } j \le r) && (130)\\
&\le \sum_{j \le r} \mathbf{P}(|Z_j| < \lambda_j) && (131)\\
&\le \sum_{j \le r} \mathbf{P}(Z_j < \lambda_j) && (132)\\
&\le \sum_{j \le r} \mathbf{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{\mu_j - \lambda_j}{s_j}\right) && (133)\\
&\le \sum_{j \le r} \mathbf{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{c f_{jj}(x) h_j - \lambda_j}{s_j}\right) && (134)\\
&\le \sum_{j \le r} \mathbf{P}\left(|\frac{Z_j - \mu_j}{s_j}| > \frac{c f_{jj}(x) h_j - \lambda_j}{s_j}\right) && (135)\\
&\approx \sum_{j \le r} \mathbf{P}\left(|\frac{\widetilde{Z_j} - \mu_j}{s_j}| > \frac{c f_{jj}(x) h_j - \lambda_j}{s_j}\right) && (136)\\
&\le \frac{r}{n c_n \sqrt{2 \log(n c_n)}} && (137)
\end{aligned}$$

Where, equation (136) follows the same idea as in equation (115). The last inequality follows from the standard Miller's inequality. Finally, summing over all iterations $s \le t$ gives

$$\begin{aligned}
\mathbf{P}\left(\bigcup_{s \le t} \bigcup_{j \le r} \left\{|Z_j^{(s)}| < \lambda_j^{(s)}\right\}\right) &\le \frac{tr}{n c_n \sqrt{2 \log(n c_n)}} && (138)\\
&\le \frac{\frac{r}{4+r} \log_{1/\beta}\left(\frac{c^2 n A_{min}^2 h_0^{4+d}}{8C \log(n c_n)}\right)}{n c_n \sqrt{2 \log(n c_n)}} \longrightarrow 0 && (139)
\end{aligned}$$

by the density Rodeo's algorithm. Thus, the bandwidths h_j for $j \le r$ satisfy, with high probability,

$$\begin{aligned}
h_j = h_0 \beta^t &\le h_0 \left(\frac{8C \log(n c_n)}{c^2 A_{min}^2 n h_0^{4+d}}\right)^{1/(4+r)} && (140)\\
&= n^{-1/(4+r)} \left(\frac{8C \log(n c_n)}{c^2 A_{min}^2 h_0^{d-r}}\right)^{1/(4+r)} && (141)
\end{aligned}$$

In particular, with probability approaching one, the algorithm runs for a number of iterations T_n bounded from below by

$$T_n \ge \frac{1}{4+r} \log_{1/\beta}(n a_n) \qquad (142)$$

where

$$a_n = \frac{c^2 A_{min}^2 h_0^{d-r}}{8C \log(n c_n)} = \widetilde{\Omega}(1). \qquad (143)$$

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

$$s \geq \frac{1}{4+r} \log_{1/\beta}(nb_n) \tag{144}$$

where $b_n = \widetilde{O}(1)$ will be defined below. From the argument above, we know that except on an event of vanishing probability, each relevant dimension $j \leq r$ has bandwidth no larger than

$$h_j \leq h_0 \beta^{\left(\log_{1/\beta}(na_n)\right)/(4+r)} = \frac{h_0}{(na_n)^{1/(4+r)}} \tag{145}$$

Thus, if relevant dimension j has bandwidth $h_j \leq h_0\beta^s$, then from lemma 4 we have that

$$\begin{aligned}
\frac{s_j^2}{\mu_j^2} &= \frac{s_j^2}{v_2^2 f_{jj}^2(x) h_j^2} && (146)\\
&\geq \frac{C}{v_2^2 f_{jj}^2(x) n h_j^4 \det(H)} && (147)\\
&\geq \frac{C}{v_2^2 f_{jj}^2(x) n h_0^4 \beta^{4s}} \frac{n^{r/(4+r)} a_n^{r/(4+r)}}{h_0^r} \frac{1}{h_0^{d-r}} && (148)\\
&= \frac{C}{v_2^2 f_{jj}^2(x) n^{4/(4+r)}} \frac{a_n^{r/(4+r)}}{h_0^{4+d}} \frac{1}{\beta^{4s}} && (149)\\
&\geq \frac{C}{A_{max}^2 n^{4/(4+r)}} \frac{a_n^{r/(4+r)}}{h_0^{4+d}} \frac{1}{\beta^{4s}} && (150)
\end{aligned}$$

Therefore

$$\frac{s_j^2}{\mu_j^2} \geq \log\log n \tag{151}$$

in case

$$\left(\frac{1}{\beta}\right)^s \;\geq\; (nb_n)^{1/(4+r)} \geq n^{1/(4+r)} \left(\frac{A_{max}^2 h_0^{4+d} \log\log n}{C a_n^{r/(4+r)}} \right)^{1/4} \tag{152}$$

which defines $b_n = \widetilde{O}(1)$. It follows that in iteration $s \geq \frac{1}{4+r}\log_{1/\beta}(nb_n)$, the probability of a relevant dimension having estimated derivative Z_j above threshold is bounded by

$$\begin{aligned}
\mathbf{P}(|Z_j| > \lambda_j, \text{ for some } j \leq r) &\leq \sum_{j \leq r} \mathbf{P}(|Z_j| > \lambda_j) && (153)\\
&= \sum_{j \leq r} \mathbf{P}\left(\frac{|Z_j|}{s_j} > \frac{\lambda_j}{s_j}\right) && (154)\\
&\approx \sum_{j \leq r} \mathbf{P}\left(\frac{|\widetilde{Z_j}|}{s_j} > \frac{\lambda_j}{s_j}\right) && (155)\\
&\leq \sum_{j \leq r} \mathbf{P}\left(\frac{s_j}{\lambda_j} e^{-\lambda_j^2/(2s_j^2)} + \frac{1}{4}\frac{\mu_j^2}{s_j^2}\right) && (156)\\
&\leq \frac{r}{nc_n\sqrt{2\log(nc_n)}} + \frac{1}{4}\sum_{j \in V_t} \frac{\mu_j^2}{s_j^2} && (157)\\
&\leq \frac{r}{nc_n\sqrt{2\log(nc_n)}} + \frac{r}{4\log\log n} && (158)\\
&= O\left(\frac{1}{\log\log n}\right) && (159)
\end{aligned}$$

which gives the statement of the theorem. ■

References

[1] E. Parzen. On the estimation of a probability density function and the mode. *The Annuals of Mathematical Statistics*, 33:832–837, 1962.

[2] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27:642–669, 1956.

[3] N.L. Hjort and M.C.Jones. Locally parametric nonparametric density estimation. *The Annals of Statistics*, 24:1619–1647, 1996.

[4] N.L. Hjort and I.K. Glad. Nonparametric density estimation with a parametric start. *The Annals of Statistics*, 23(3):882–904, 1995.

[5] C.R. Loader. Local likelihood density estimation. *The Annals of Statistics*, 24:1602–1618, 1996.

[6] A.P. Dempster, N.M. Laord, and D.B. Rubin. Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[7] N.M. Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73:805–811, 1978.

[8] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1994.

[9] S. Richardson and P.J. Green. On bayesin analysis of mxitures with an unknown number of components. *Journal of the Royal Statistical Society, Series B*, 59(4):731–792, 1997.

[10] C.R. Genovese and L.A. Wasserman. Rates of convergence for the gaussian mxiture sieve. *The Annals of Statistics*, 28(4):1105–1127, 2000.

[11] S. Ghosal, J.K. Ghosh, and A.W. van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 2000.

[12] J. Friedman, W. Stuetzele, and A. Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[13] C.J. Stone. Large sample inference for log-spline models. *The Annals of Statistics*, 18:717–741, 1990.

[14] B.W. Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, 10:795–810, 1982.

[15] J.D.Lafferty and L.A.Wasserman. Rodeo: Sparse nonparametric regression in high dimensions. *Advances in Neural Information Processing Systems (NIPS)*, 18, 2005.

[16] J.S.Marron and M.P. Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20:712–736, 1992.

[17] A.Azzalini and A.W.Bowman. A look at some data on the old faithful geyser. *Applied Statistics*, 39:357–365, 1990.

[18] J.D.Lafferty and L.A.Wasserman. Rodeo: Sparse nonparametric regression in high dimensions. *Preprint*, 2006.

[19] P. Bickel and B. Li. Local polynomial regression on unknown manifolds. *Technical Report*, 2006.