Sparse Nonparametric Density Estimation in High Dimensions Using the Rodeo

Han Liu[†‡] **John Lafferty**[*‡] **Larry Wasserman**[†‡]

[†]Statistics Department
[*]Computer Science Department
[‡]Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA

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 component and a nonparametric component which depends on an unknown subset of the variables. Using a modification of a recently developed nonparametric regression framework called *rodeo* (regularization of derivative expectation operator), we propose a method to greedily select bandwidths in a kernel density estimate. It is shown empirically that the density rodeo works well even for very high dimensional problems. When the unknown density function satisfies a suitably defined sparsity condition, and the parametric baseline density is smooth, the approach is shown to achieve near optimal minimax rates of convergence, and thus avoids the curse of 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. Nonparametric density estimation methods such as the kernel estimator [1, 2], or local likelihood [3, 4, 5], work 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 to the slow rates of convergence of the estimators. Density estimation in high dimensions is often carried out by the use of 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 correct. While nonparametric mixture models can adapt the number of components to the data, such estimators achieve, at best, the same rates as kernel estimators. In fact, the theoretical guarantees with mixtures are generally not as good as for kernel estimators; see [10] and [11]. Other methods for high dimensional density estimation include projection pursuit [12], log-spline models [13] and penalized likelihood [14].

In d-dimensions, minimax theory shows that the best convergence rate for the mean squared error under standard smoothness assumptions is $\mathcal{R}_{opt} = O(n^{-4/(4+d)})$; this exhibits the "curse of dimensionality" when d is large. In this paper we present a method that achieves faster rates of convergence when a certain sparsity assumption is satisfied. Moreover, the method is based on a greedy algorithm that is computationally efficient. The idea comes from the recently 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, where $r \ll d$, the rodeo simultaneously performs bandwidth selection and (implicitly) variable selection to achieve a better minimax convergence rate of $\widetilde{O}(n^{-4/(4+r)})$, which is optimal up to a logarithmic factor, as if the r relevant variables were known and explicitly isolated in advance. The purpose of this paper is to extend this idea to the nonparametric density estimation setting.

It is first necessary to define an appropriate sparsity condition in the density estimation setting. Our key assumption is

$$f(x_1, \ldots, x_d) = g(x_R)\, b(x_1, \ldots, x_d)$$

where g is an unknown function, $x_R = (x_j : \ j \in R)$

is a subset of the variables, and b is a baseline density, which is either 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 where the baseline density b is uniform, so that $f(x) = g(x_R)$, and where b is normal, as in [3, 4]. We develop two versions of the rodeo for density estimation, a local version and a global version. The local version estimates f at a fixed point x and results in a local bandwidth selection algorithm. The global version estimates a single set of bandwidths that is used at each test point.

2 The Local Rodeo

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

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

where $\mathcal{K}$ is a symmetric kernel with $\int \mathcal{K}(u)du = 1$ and $\int u\mathcal{K}(u)du = 0_d$. We assume that $\mathcal{K}$ is a product kernel and $H = diag(h_1, ..., h_d)$ is diagonal, so that

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

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 derivatives $(Z_j : 1 \le j \le d)$ of the kernel density estimate with respect to each bandwidth h_j, and reduce bandwidth h_j if Z_j is large. The test statistic is

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

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

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

Density Estimation Rodeo

1. *Select* parameter $0 < \beta < 1$ and initial bandwidth $h_0 = c_0/\log\log n$ for some constant c_0. Also, let $c_n = O(\log n)$.
2. *Initialize* h_j, 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) Estimate the 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^* and estimator $\widehat{f}_{H^*}(x)$

Figure 1: The density rodeo algorithm.

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 resulting algorithm is given in Figure 1.

For a general kernel, we have that

$$\begin{aligned}Z_j &= -\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{h_j} + \frac{x_j - X_{ij}}{h_j^2}\widetilde{K}\left(\frac{x_j - X_{ij}}{h_j}\right)\right) \\ &\quad\times \prod_{k=1}^{d}\frac{1}{h_k}K\left(\frac{x_k - X_{ik}}{h_k}\right)\end{aligned}$$

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

$$\begin{aligned}Z_j &= \frac{\partial\widehat{f}_H(x)}{\partial h_j} \\ &= \frac{C}{n}\cdot\sum_{i=1}^{n}\left((x_j - X_{ij})^2 - h_j^2\right)\prod_{k=1}^{d}K\left(\frac{x_k - X_{ik}}{h_k}\right) \\ &\propto \frac{1}{n}\sum_{i=1}^{n}\left((x_j - X_{ij})^2 - h_j^2\right)\prod_{k=1}^{d}K\left(\frac{x_k - X_{ik}}{h_k}\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left((x_j - X_{ij})^2 - h_j^2\right)e^{-\sum_{k=1}^{d}\frac{(x_k - X_{ik})^2}{2h_k^2}}\end{aligned}$$

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

2.1 Local Likelihood Rodeo

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

problem $\max_\theta \ell(f,x)$ where

$$\begin{aligned}\ell(f,x) &= \sum_{i=1}^{n} \mathcal{K}\left(H^{-1}(X_i-x)\right)\log f(X_i;\theta) \\ &- n\int_{\mathcal{X}} \mathcal{K}\left(H^{-1}(u-x)\right) f(u;\theta)du\end{aligned}$$

is a localized version of the usual log-likelihood function for density estimation problems:

$$\ell(f) = \sum_{i=1}^{n} \log f(X_i;\theta) - n\left(\int_{\mathcal{X}} f(u;\theta)du - 1\right)$$

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 with respect to boundary effects [5]. When assuming a product Gaussian kernel, the closed form of the local likelihood estimator can be written as $\widetilde{f}_H(x) = \widehat{f}_H(x) \times e^B$, with

$$B = -\frac{1}{2}\sum_{k=1}^{d} h_k^2 \left(\frac{\sum_{i=1}^{n}\prod_{j=1}^{d} K\left(\frac{X_{ij}-x_j}{h_j}\right)\left(\frac{X_{ik}-x_k}{h_k^2}\right)}{\sum_{i=1}^{n}\prod_{j=1}^{d} K\left(\frac{X_{ij}-x_j}{h_j}\right)}\right)^2$$

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

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

Then

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

where $\frac{\partial}{\partial h_j}\log\widehat{f}_H(x) = \frac{\frac{\partial}{\partial h_j}\widehat{f}_H(x)}{\widehat{f}_H(x)}$ is calculated as in the previous section. The derivation of the second term, though quite involved, is straightforward. The same algorithm in Figure 1 applies.

2.2 Other Baseline Densities

When using a different baseline density, for example, a normal density, we use the semiparametric density estimate

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

where $\widehat{b}(x)$ is a parametric density estimate at the point x, with its parameters estimated by maximum likelihood. Since the parameters in this term are easy to estimate, we treat them as known. The motivation for this estimator comes from the local likelihood method; instead of using a polynomial $P(x)$ to approximate the log density $\log f(x)$, we use $\log b(x) + P(x)$. In this setting, if the true function is $b(x)$, the algorithm will tend to freeze all of the bandwidths for the estimator at their large initial values h_0.

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

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

where $(\widehat{\mu}, \widehat{\Sigma})$ is the M.L.E. for the 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_j is calculated in a similar manner. The variance of Z_j can be estimated using the bootstrap; see Section 4.1.

3 The Global Rodeo

Instead of using the local rodeo, which selects bandwidths for each evaluation point, the method can be modified to carry out global bandwidth selection, in which case each dimension uses a single bandwidth for all points. The idea is to average the test statistic over multiple evaluation points $x_1, ..., x_m$, which are sampled from the empirical distribution of the observed sample.

Averaging the Z_js directly leads to a statistic whose mean for relevant variables is asymptotically $\frac{1}{m}h_j\sum_{i=1}^{m} f_{jj}(x_i)$. However, as observed in [15], because of sign changes in $f_{jj}(x)$, cancellations can lead to an artificially small value for the statistic. To avoid this problem, the statistic is squared. Let $x_1, ..., x_m$ denote the evaluation points and let $Z_j(x_i)$ denote the derivative for the i-th evaluation point with respect to

the bandwidth h_j. Therefore,

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

Let $\gamma_{jk} = (Z_{j1}(x_k), Z_{j2}(x_k), ..., Z_{jm}(x_k))^T$, for $k = 1, ..., n$. 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, $\mathbf{Var}(Z_{j\cdot}) = \Sigma_j/n \equiv C_j$. Based on this, we define the test statistic

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

where

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

with $\widehat{\mu} = \frac{1}{m}\sum_{i=1}^{m} Z_j(x_i)$. For an irrelevant dimension $j \in R^c$, it can be shown that $\mathbf{E}Z_j(x_i) = o_P(h_j)$, 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

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

Several examples of this algorithm are given in Section 5.

4 Extensions

In this section we briefly discuss extensions of the density estimation rodeo, one involving bootstrap estimation of the variance, and another that runs the algorithm in reverse, with bandwidths starting small and gradually becoming larger.

4.1 Bootstrap Version

As we have seen, an explicit expression for the derivatives Z_j and variance s_j^2 can be derived when the underlying density estimate has an explicit form. In some cases, however, the density estimate itself may not have a closed form, and evaluation of the derivatives becomes problematic. When explicit forms can not be derived, the derivatives can still be practically approximated using finite differences:

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

where $H + \triangle h_j$ means adding a small value $\triangle h_j$ to the j-th diagonal element of H. The variance of Z_j can then be estimated using the bootstrap; the algorithm is detailed in Figure 2.

Bootstrap Variance Estimation

1. *Draw* a sample $X_1^*, ..., X_n^*$ of size n.
2. *Compute* Z_j^* from data $X_1^*, ..., X_n^*$.
3. *Repeat* steps 1 and 2, B times, to obtain $Z_j^{*(b)}$ for $b = 1, \ldots, B$.
4. *Output* estimated variance

$$s_j^2 = \frac{1}{B}\sum_{b=1}^{B}\left(Z_j^{*(b)} - \frac{1}{B}\sum_{r=1}^{B} Z_j^{*(r)}\right)^2$$

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

The bootstrap applies to both the local and global rodeo algorithms, and thus provides a general tool when explicit formulas for the variance are not available. Such cases include the local likelihood rodeo and the rodeo applied in the semiparametric case. We expect that a theoretical analysis can be derived similar to the results we discuss in Section 6 below. The drawback of this approach is that the bootstrap is computationally intensive.

4.2 Reverse Rodeo

Thus far, the rodeo algorithms presented have employed a sequence of *decreasing* bandwidths, estimating the optimal value by a sequence of hypothesis tests. As an alternative, it is possible to begin with very small bandwidths, and test whether the estimator changes significantly as a bandwidth is increased. This reverse rodeo can be helpful when many dimensions are expected to need small bandwidths; an illustration of this is given in the following section with image data.

5 Examples

In this section, we demonstrate rodeo density estimation on both synthetic and real data, including one-dimensional, two-dimensional, and high dimensional examples that illustrate the behavior of the algorithm in various conditions. For the purpose of evaluating the algorithm's performance quantitatively, we use the Hellinger distance. Assuming we have m evaluation points, this distance is approximated as

$$\begin{aligned} d(\widehat{f}, f) &= \int\left(\sqrt{\widehat{f}(x)} - \sqrt{f(x)}\right)^2 dx \\ &= 2 - 2\int\sqrt{\frac{\widehat{f}(x)}{f(x)}} f(x)\, dx \end{aligned}$$

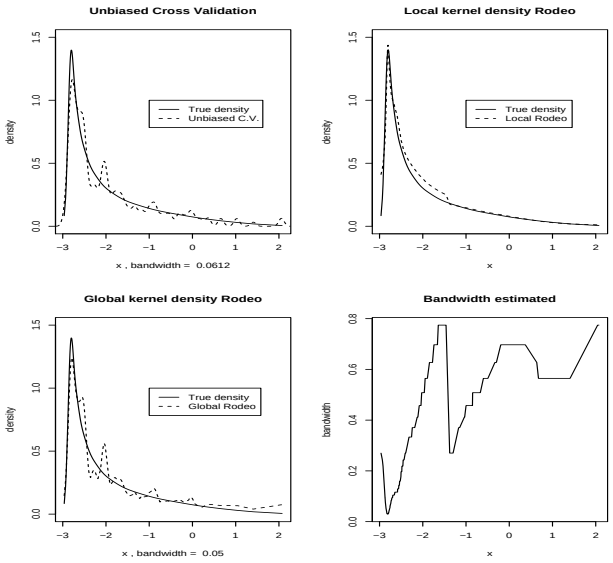


Figure 3: Different versions of the algorithms run on the highly skewed unimodal example. The first three plots are results for the different estimators; the last plot shows the bandwidths selected by the local rodeo.

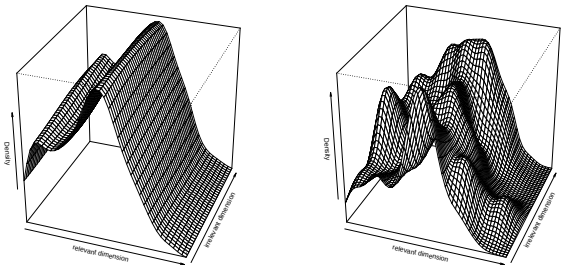
$$\approx \quad 2 - \frac{2}{m}\sum_{i=1}^{m}\sqrt{\frac{\widehat{f}(X_i)}{f(X_i)}}$$

This measure is more numerically stable than the commonly used Kullback-Leibler (KL) divergence for evaluating the discrepancy between two density functions. In the following, we first use simulated data to investigate the algorithm performance, where the true distribution is known. We then apply the rodeo to some real data. If not explicitly stated otherwise, the data are always rescaled to lie in a d-dimensional cube $[0,1]^d$, and a product Gaussian kernel is used. The default parameters are $c_0 = 1$, $c_n = \log d$, and $\beta = 0.9$.

5.1 One-Dimensional Examples

We first apply the density rodeo on one-dimensional examples. We conduct a comparative study on a list of 15 "test densities" proposed by Marron and Wand [16], which are all normal mixtures representing several different challenges for density estimation methods. Our approach achieves performance that is comparable to the built-in kernel density estimator with bandwidth selected by unbiased cross-validation (from the R base library). Due to space considerations, 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; the



Rodeo estimate KDE2d estimate

Figure 4: 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.

distribution is

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

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, where the sample size is $n = 200$. In these figures, the solid line is the true density function, and the dashed line illustrates the estimated densities by different methods. The local rodeo performs best. This is because the true density function is highly skewed, and a fixed bandwidth density estimator fails to fit the very smooth tail. The last subplot from Figure 3 shows the selected bandwidth for the local rodeo; it illustrates how smaller bandwidths are selected where the function varies more rapidly. Comparing the Hellinger distances of the estimates to the true density shows that the local rodeo works best, while the global rodeo and the unbiased cross-validation methods are comparable in this one-dimensional example.

5.2 Two-Dimensional Examples

Two-dimensional examples can also be easily visualized. We evaluate a synthetic dataset and a real dataset. The density rodeo's performance is compared with the built-in method `KDE2d` from the `MASS` package in R. The empirical results show that the density rodeo outperforms 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 results that are very similar to those of previous authors.

Example 2 (Mixture of Beta distributions, with the

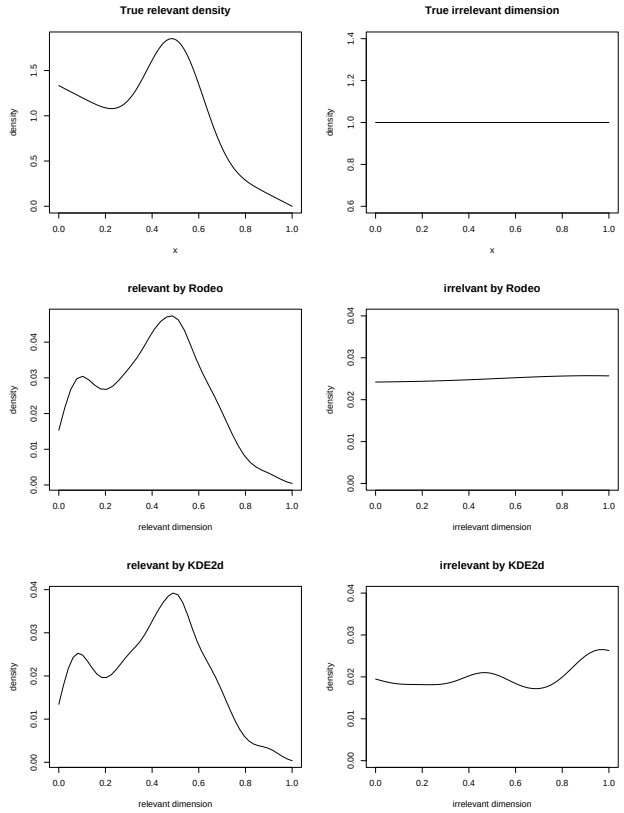


Figure 5: Marginal distributions of the relevant and the irrelevant dimensions for Example 2.

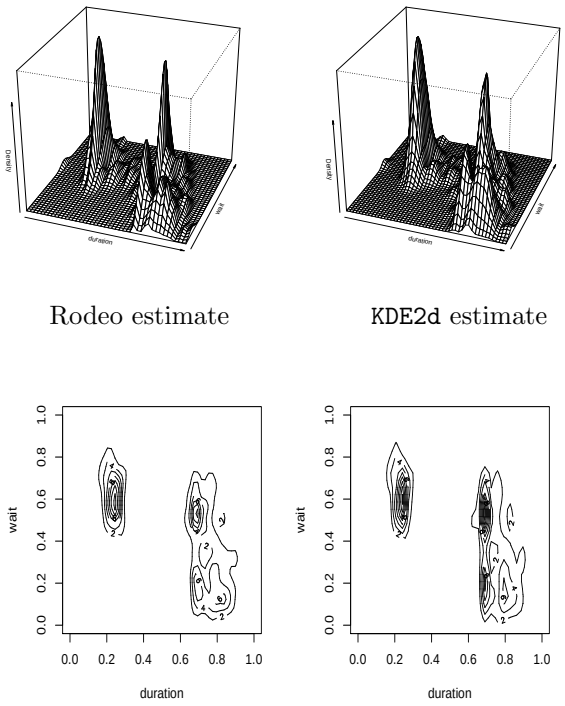


Figure 6: Top: Perspective plots of the estimated density functions by the global rodeo (left) and the R built-in method KDE2d (right) on the geyser data. Bottom: Contour plots of the result from the global rodeo (left) and KDE2d (right)

uniform distribution for an irrelevant dimension). In this example we simulate a 2-dimensional dataset with $n = 500$ points. The two dimensions are independently generated as

$$\begin{aligned} X_1 &\sim \tfrac{2}{3}\text{Beta}(1,2) + \tfrac{1}{3}\text{Beta}(10,10) \\ X_2 &\sim \text{Uniform}(0,1) \end{aligned}$$

Figure 4 shows perspective plots of the estimated density functions by the global rodeo and the built-in method KDE2d. 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 5 illustrates the numerically integrated marginal distributions of the two estimators (not normalized); it is seen that the rodeo fit is better than that of KDE2d, which is consistent with the previous observations.

Example 3 (Geyser data). This example uses a version of the eruptions data from the "Old Faithful" geyser in Yellowstone National Park [17]. The data consist of measurements taken between 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, "wait," represents the waiting time between eruptions. We apply the global rodeo algorithm on this dataset. The estimated densities using the rodeo and the built-in KDE2d method (used by the original authors) are provided in the upper plot of Figure 6. The two lower plots of Figure 6 show the corresponding contour plots. Based on visual inspection, the two estimates are very similar.

5.3 High Dimensional Examples

Example 4 (High dimensional synthetic data). Figure 7 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

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

while the irrelevant dimensions are generated as

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

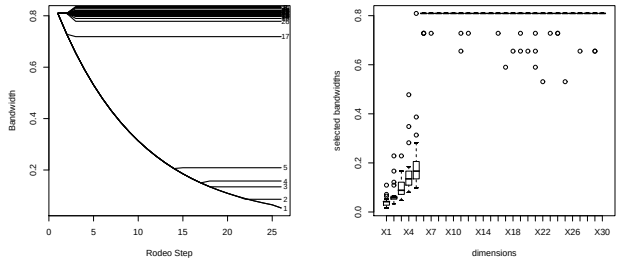


Figure 7: The bandwidth output by the local density rodeo for a 30-dimensional synthetic dataset (left) and its boxplot for 30 trials (right).

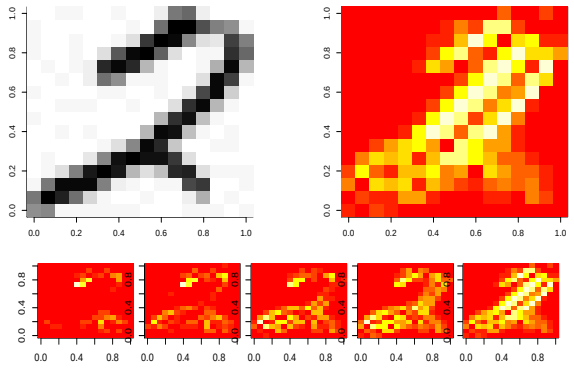


Figure 8: An image processing example: Evaluation digit (top left) and the bandwidths output by the reverse rodeo (top right). Each pixel corresponds to a dimension, and is assigned a bandwidth. The lower subplots illustrate a series of bandwidths sampled at different reverse rodeo steps: 10, 20, 40, 60, and 100. Darker colors correspond to smaller bandwidths.

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

Example 5: (Scanned digits). Here we apply the reverse local rodeo on image data. The results are shown in Figure 8. The algorithm was run on 2000 grayscale images of handwritten 1s and 2s. Each scanned handwritten digit has $256 = 16 \times 16$ pixels with some unknown background noise. Each pixel is considered a variable; this is therefore a 256-dimensional density estimation problem. An evaluation point is shown in the upper left subplot of Figure 8; the bandwidths output by the reverse rodeo algorithm are shown in the upper right subplot. The estimated bandwidth plots in different rodeo steps (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 marginal density close to a point mass. This requires a small bandwidth since the marginal density is highly peaked. The reverse rodeo starts from a small bandwidth, which is more efficient than the original rodeo and is more stable numerically. Figure 8 shows the evolution of the bandwidths, which can be viewed as a dynamic feature selection process—-the earlier a dimension's bandwidth increases, the more informative it is, and the greater variation there is in the local density. The reverse rodeo is quite efficient for this extremely high dimensional problem. It is interesting to note how the early stages of the rodeo reveal that some of the 2s in the data have looped bottoms, while others have straight bottoms; the evaluation point does not have such a loop.

In addition to these examples, we also conducted experiments using the Gaussian density as the baseline distribution, the results are similar to the above. Details will be available in the long version of the paper.

6 Asymptotic Properties

We conclude with a brief discussion of the asymptotic properties of the density rodeo, assuming that the baseline density $b(x)$ is a very smooth function. At a high level, our theoretical analysis shows that, with probability approaching one with sample size, when $f(x) = g(x_R)\, b(x)$, the bandwidths for the variables appearing in the nonparametric factor g have bandwidths that shrink, while the bandwidths for the remaining variables remain large. Note that, intuitively, the algorithm eventually halts before the bandwidths become too small, since the variance of the derivatives increases as the bandwidths shrink; thus, all derivatives are eventually below threshold. In fact, we are able to show that the variables in R have bandwidths that shrink all the way down to size $h_j \approx n^{-1/(4+|R|)}$. Together with asymptotic expansions of the bias and variance of the kernel or local linear estimators, this implies that the risk of the density rodeo estimator is of order $\widetilde{O}_P(n^{-4/(4+|R|)})$. Thus, the rate of convergence is as if we were carrying out density estimation

in $|R|$ dimensions.

In a bit more detail, 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 in R correspond to $1 \leq j \leq r$ and the irrelevant variables x_j in the complement R^c correspond to $r+1 \leq j \leq d$. We make standard assumptions on the kernel, and assume that $d = O(1)$; the remaining technical assumptions are omitted for clarity and lack of space.

Theorem. *The density rodeo algorithm outputs bandwidths $H^* = diag(h_1^*, ..., h_d^*)$ that satisfy*

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

Furthermore, for all $j \leq r$

$$\mathbf{P}\left(h_j^{(0)}(nb_n)^{\frac{-1}{(4+r)}} \leq h_j^* \leq h_j^{(0)}(na_n)^{\frac{-1}{(4+r)}}\right) \to 1$$

for certain constants a_n and b_n that are logarithmic in n. Moreover, the risk $\mathcal{R}_{H^*}$ of the rodeo density estimator satisfies

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

The proof is structured in a way that parallels the proof of the analogous statement for the regression rodeo [15]. However, there are modifications required; for example, we use the Berry-Esseen method to obtain uniform bounds on the deviation of the estimator from its mean over a set of bandwidths. Details are provided in the full version of the paper.

References

[1] E. Parzen. On the estimation of a probability density function and the mode. *The Annals 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 Bayesian analysis of mixtures 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 mixture 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. Full version: `http://arxiv.org/abs/math/0506342`.

[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.