An Efficient Algorithm for Nonparametric Online Prediction

Haijie Gu
School of Computer Science
Carnegie Mellon University

Advisor: John Lafferty

April 29, 2011

Abstract

A new approach to online prediction using nonparametric statistics is described and analyzed. This approach is based on a sequential local polynomial regression procedure that has linear runtime complexity. We provide a theoretical analysis showing that the algorithm achieves the optimal minimax rate of convergence. Using the mixing experts framework, we also develop a method that adapts to the unknown global smoothness of the true regression function.

1 Introduction

Bandwidth selection is a crucial issue in nonparametric statistics. It is well known that the optimal bandwidth for regression depends on the sample size. Considering the one dimensional case, let

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

where the X_is are identical and independently generated under some distribution, $m : \mathbb{R} \to \mathbb{R}$ is the unknown function to estimate, and $\epsilon_i \sim N(0, \sigma^2)$. Assuming that m'' is absolutely continuous and $\int (m''(x))^2 dx < \infty$, the risk of a kernel regression with bandwidth h has the form

$$R(\hat{m}_n, m) = c_1 h^4 + \frac{c_2}{nh} + O(\frac{1}{n}) + O(h^6) \tag{2}$$

where

$$R(\hat{m}_n, m) = \mathbb{E}_m[\int (\hat{m}_n(x) - m(x))^2 dx] \tag{3}$$

is defined as the risk of the estimate $\hat{m}_n$ on a sample of size n, and where c_1 and c_2 are constants that depend on the distribution of X [12]. The optimal bandwidth that minimizes (2) is $h^* = O(n^{-4/5})$, which leads to the optimal minimax rate of convergence $O(n^{-4/5})$; see [7]. More generally, if we assume further smoothness of m, so that the d-th derivative of m exists and is bounded, the minimax rate becomes $n^{-2d/(2d+1)}$, which can be achieved by performing local polynomial regression of order $d-1$ with the optimal bandwidth $h^* = O(n^{-1/(2d+1)})$ [5].

These classical results assume that a training data set of size n is given, and formally lets the size n of the data to increase to infinity. In the online setting, however, the data arrive sequentially, and the size of the data set is changing. In this case the bandwidth needs to adapt to the changing sample size. A naive variation of the classical methods would carry out batch regression with variable bandwidth each time a new data point arrives. However, this would require quadratic complexity $O(T^2)$ to compute the estimates after T points are observed. This is prohibitive for large sample sizes.

This motivates the problem studied in this thesis, to develop an efficient algorithm for adaptive sequential regression. Significant previous work in the literature has been devoted to the related problems. For adaptive bandwidth selection in the batch setting, Fan et al. [4] considered using Residual Squares Criteria (RCS) for performing data-driven bandwidth selection in local polynomial regression; Ruppert et al. [10] proposed the plug-in bandwidth selection for local linear kernel estimators. These are effective methods for adaptive estimation, however, they did not take into account the computational cost for online updating. Among recent work, Steland [11] investigated a cross-validation scheme for sequential data and established theoretical results. Again, however, this does not consider the cost of re-computing the entire model for each new bandwidth; furthermore, performing the leave-one-out cross-validation adds extra computation and would be impractical for many applications.

In the online setting, Kivinen and Smola developed computationally efficient algorithms for online learning in a reproducing kernel Hilbret space (RKHS). However, their RKHS analysis does not consider adaptation to the unknown smoothness of the regression function. The mixing expert framework has been a popular strategy for online prediction, and there is a rich literature on this topic. Cesa-Bianchi and Lugosi [2] derive regret bounds under different assumptions. Yang [13] establishes risk bounds under mild

conditions on the loss function. These results bound the performance of the combined estimator relative to that of the best expert. Advanced results by Bunea and Nobel [1] show a risk bound in terms of the generalized simplex linear combination of a set of fixed estimators. However, this work does not allow the online situation where the experts change over time. We refer below to the work of Cesa-Bianchi et al. [2], and Yang [13] when using the mixing expert framework.

In this paper we propose a new algorithm for sequential regression that requires linear computational cost. Moreover, we prove that the algorithm achieves the optimal minimax rate of convergence (see Theorem 3.3 and Theorem 3.5). The essential idea is to avoid recomputation by shrinking the bandwidth for each new observed data point. A similar algorithm appears in Kristan et al. [8]. They approach online density estimation using a Gaussian mixture model, which is updated by adding a new Gaussian component with adapted bandwidth. The strength of this approach was illustrated with examples in density estimation and computer vision problems. Although their examples are encouraging, there are no theoretical results. Our sequential algorithm is based on the local polynomial regression which is more general and flexible that kernel density estimation. In addition, we take a different approach from [8] in selecting the right smoothing parameter in each trial. Instead of directly picking the bandwidth, we show how the weighted expert framework can be used to adapt to unknown smoothness. The experimental results confirm our theoretical analysis and show the approach is practical for sequential regression.

The organization of this paper is as follows. In §2 we present the algorithm for sequential local polynomial regression. In §3 we present a theoretical risk analysis results for both sequential density estimation (Theorem 3.3) and regression (Theorem 3.5). Full proofs of the results are postponed to the Appendix. In §4 we introduce the mixing expert framework and show various bounds used to guarantee adaptation to unknown smoothness. In §5 we present experimental results which show that our algorithm (1) is quite comparable to the batch algorithm but much more efficient (2) adapts to the global smoothness of the true regression function.

2 Sequential Local Polynomial Smoothing

The efficient sequential estimator is extended from the standard local polynomial regression. We choose this particular smoother for two reasons. First among the various nonparametric regression methods, local polynomial regression enjoys desirable minimax properties as well as other features such

as the automatic boundary treatment [6]. Second, for its generality in application and analysis: the sequential version of local constant, local linear estimation can viewed as a special case of the sequential local polynomial regression.

Let $Z = \{(X_1, Y_1), (X_2, Y_2), \dots\}$ be a sequence of observation independent and identically distributed random variables from some unknown function m and independent noise ϵ_i with mean zero and variance σ_i.

At time $T+1$, an order-d local polynomial regression at the prediction point $x_0 = X_{T+1}$ is computed by minimizing

$$\sum_{t=1}^{T}(Y_t - \sum_{j=0}^{d}\beta_j(x_0)(X_t - x_0)^j)^2 K(\frac{X_t - x_0}{h_T}) \tag{4}$$

where $K(\cdot)$ is a symmetric weight function or kernel, and h_T is the smoothing parameter or bandwidth. Denote by $\hat{\beta}_j(x_0), j = 0, \dots, d$ the solution of the weighted least squares loss function (4).

Let $\mathbf{X}$ be the design matrix

$$\mathbf{X} = \begin{pmatrix} 1 & (X_1 - x_0) & \cdots & (X_1 - x_0)^d \\ \vdots & \vdots & \vdots & \\ 1 & (X_T - x_0) & \cdots & (X_T - x_0)^d \end{pmatrix}$$

and put

$$\mathbf{y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_T \end{pmatrix} \quad \hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \vdots \\ \hat{\beta}_d \end{pmatrix}$$

and

$$\mathbf{W} = diag\{K_{h_T}(X_t, x_0)\}_{1 \le t \le T}$$

the $n \times n$ diagonal matrix of weights, the solution that minimizes (4) is

$$\hat{\beta}(x_0) = (\mathbf{X^T W X})^{-1}\mathbf{X^T W y} \tag{5}$$

It follows from Taylor's formula that $\nu!\hat{\beta}_\nu(x_0)$ is an estimator for $m^\nu(x_0)$. Therefore the estimation for $\hat{Y}_{T+1} = \hat{\beta}_0(X_{T+1})$.

From the (5) above, it is clear that if we would adapt h_{T+1} to the increased size $T+1$, we would have to recompute the entire $X^T W X$ matrix. To save computation, we choose to allow variable bandwidth in W. In other words, the effect of new bandwidth h_{T+1} only applies to Z_{T+1}. Based on this

motivation, we have constructed the Sequential Local Polynomial Regression (SLPR) as follows. Let

$$\mathbf{W_T} = diag\{K_{h_t}(X_t, x_0)\}_{1 \le t \le T}$$

where $h_t = c \cdot t^{-1/(2d+1)}$ is the bandwidth with respect to the sample size t, and c is some constant.

The estimator at time $t = n$ is

$$\hat{\beta}(x_0) = (\mathbf{X^T W_n X})^{-1} \mathbf{X^T W_n y} \tag{6}$$

Denote the $d+1 \times d+1$ matrix $\mathbf{X^T W_n X}$ as $\mathbf{S_n}$, with the (i,j) entry

$$S_n(i,j) = \sum_{t=1}^{n} K_{h_t}(X_t, x_0)(X_t - x_0)^{i+j} \tag{7}$$

where $K_{h_t}(X_t, x) = \frac{1}{h_t} K(\frac{X_t - x}{h_t})$.

To update the model after we observed (X_{n+1}, Y_{n+1}), we discover that

$$S_{n+1} = S_n + K_{h_{n+1}}(X_{n+1}, x_0) \mathbf{x_{n+1}} \mathbf{x_{n+1}^T} \tag{8}$$

where $\mathbf{x_{n+1}}$ is the length $d+1$ vector: $[(X_{n+1} - x_0)^i]_{i=0,\ldots,d}$.

By Woodbury matrix inverse lemma:

$$(A + vv^T)^{-1} = A^{-1} - A^{-1} v (1 + v^T A^{-1} v)^{-1} v^T A^{-1} \tag{9}$$

and (8), updating S^{-1} takes $O(d^2)$.

Similarly, updating

$$\mathbf{X_{n+1}^T W_{n+1} y_{n+1}} = \mathbf{X_n^T W_n y_n} + K_{h_{n+1}} \mathbf{x_{n+1}} Y_{n+1}$$

costs $O(d)$.

Therefore, the total complexity of adapting SLPR to the increased sample size is $O(d^2)$, which is independent of T.

If we were to apply h_{n+1} to all $Z_1, \ldots, Z_n$, as doing a batch local polynomial, the cost for updating the model would be $O(n^2 d^2)$.

3 Risk Analysis

In this section, we present our main technical results: the risk analysis of the sequential density estimation and regression. Our goal is to show that the asymptotic risk of SPLR has the rate of convergence of $n^{-2d/(2d+1)}$ assuming

m in C^d, the existence of d continuous derivatives of m. This rate equals the minimax optimal rate of Local Polynomial Regression.

The main techniques involved in the sequential risk analysis are bias-variance decomposition and integral approximation.

We first analyze the risk of sequential kernel density estimation (SKDE) and sequential kernel regression (SKR). These two examples can be viewed as special cases of the local polynomial regression but they are simpler and thus more illustrative. Afterwards, we generalized the results to order $d-1$ $SLPR$ for m in C^d.

We assume that the true density function f and the true regression function m have $d \geq 2$ continuous derivatives. The kernel K satisfies the following properties:

$$\int K(u)du = 1, \int K(u)udu = 0, \int K(u)u^2du > 0$$

We also restrict our bandwidth $\{h_t | t = 1, 2, 3 \dots\}$ to satisfy

$$\lim_{t\to\infty} h_t = 0 \quad \lim_{n\to\infty} \frac{1}{n^2} \sum_{t=1}^{n} \frac{1}{h_t} = \infty$$

To simplify the notation, we put the following definitions:

- $K_{h_t}(x, X_t) = \frac{1}{h_t} K(\frac{x - X_t}{h_t})$
- $\sigma_K^2 = \int x^2 K(x)dx$

3.1 Optimal Risk for Sequential Kernel Density Estimation

The sequential kernel density estimator $\hat{f}$ is given by

$$\hat{f}_n(x) = \frac{1}{n} \sum_{t=1}^{n} \frac{1}{h_t} K\left(\frac{x - X_t}{h_t}\right) \tag{10}$$

The bias-variance decomposition of $\hat{f}_n$ at x is:

$$R_x(f(x), \hat{f}_n(x)) = Bias^2(\hat{f}_n(x)) + Var(\hat{f}_n(x)) \tag{11}$$

Lemma 3.1. *The point-wise risk of SKDE at time $t = n$ and input x is*

$$\begin{aligned} & Risk(f(x), \hat{f}_n(x)) \\ =& \frac{1}{4}(f''(x))^2(\sigma_k^2)^2 \frac{(\sum_{t=1}^{n} h_t^2)^2}{n^2} \\ +& \frac{f(x)}{n^2} \int k^2(u)du \cdot (\sum_{t=1}^{n} \frac{1}{h_n}) + o(\frac{\sum_{t=1}^{n} \frac{1}{h_t}}{n^2}) + o\left(\frac{(\sum_{t=1}^{n} h_t^2)^2}{n^2}\right) \end{aligned} \tag{12}$$

The proof of Lemma 3.1 is based on the classical bias variance analysis and is postponed to Appendix §A.1.

Taking integral of the above equation over input x, the risk SKDE estimator at time $t = n$ is as follows:

$$\begin{aligned} & Risk(f, \hat{f}_n) \\ =& \frac{1}{4}\left(\int (f''(x))^2 dx\right)(\sigma_K^2)^2 \frac{(\sum_{t=1}^n h_t^2)^2}{n^2} \\ +& \frac{1}{n^2}\int k^2(u)du \cdot (\sum_{t=1}^n \frac{1}{h_t}) + o(\frac{\sum_{t=1}^n \frac{1}{h_t}}{n^2}) + o\left(\frac{(\sum_{t=1}^n h_t^2)^2}{n^2}\right) \end{aligned} \tag{13}$$

Using Lemma 3.1, we can prove Theorem 3.2 by integral approximation on the sum.

Theorem 3.2. *Let $h_t = t^{-1/5}, t = 1, 2, \ldots, n$, the risk of online kernel density estimator is $O(n^{-4/5})$.*

Proof. Let $c_1 = \frac{1}{4}(\int (f''(x))^2 dx)(\sigma_K^2)^2$, and $c_2 = \int K^2(u)du$:

$$Risk(f, \hat{f}_n) = \frac{1}{n^2}\left[c_1(\sum_{t=1}^n h_t^2)^2 + c_2 \sum_{t=1}^n (\frac{1}{h_t})\right] \tag{14}$$

Let $h_t = t^{-k}, k < 0$, and plug it in (14).

$$\begin{aligned} \hat{Risk} &= \frac{1}{n^2}\left[c_1(\sum_{t=1}^n t^{-2k})^2 + c_2 \sum_{t=1}^n t^k\right] \\ &\leq \frac{1}{n^2}\left[c_1(\int_0^n t^{-2k} dt)^2 + c_2 n^{k+1}\right] \\ &= \frac{1}{n^2}\left[c_1 \frac{1}{(1-2k)^2} n^{2-4k} + c_2 n^{k+1}\right] \\ &= c_1 \frac{1}{(1-2k)^2} n^{-4k} + c_2 n^{k-1} \end{aligned} \tag{15}$$

Differentiate (15) with respect to k and set it equal to 0, we have that: when $k^* = -\frac{1}{5}, h_t^* = t^{k^*}$, the optimal risk of SKDE $Risk^* = O(n^{-4/5})$. □

The above analysis uses the assumption that the density function f has continuous second derivative. In general, this can be extended to the case where f is in C^d, having d continuous derivatives.

Theorem 3.3. *If the density function f has up to d continuous derivatives, the optimal risk of SKDE is $Risk^* = O(n^{-2d/2d+1})$, with $h_t^* = O(t^{-1/2d+1})$.*

Proof. Suppose the true density function f has up to d continuous derivatives, we take Taylor's formula of f to order d. By a similar calculation, we have:

$$\begin{aligned}\mathbb{E}[[]\hat{f}_n(x)] &= \frac{1}{n}\sum_{t=1}^{n}\int K(u)f(x-h_t u)du \\ &= \frac{1}{n}\sum_{t=1}^{n}[f(x)+\frac{h_t^2}{2}f''(x)\sigma_K^2+\cdots+\frac{h_t^d}{d!}f^d(x)\sigma_K^d+o(h_t^d)] \\ &= f(x)+\frac{1}{2}f''(x)\sigma_K^2(\frac{\sum_{t=1}^{n}h_t^2}{n})+\cdots+\frac{1}{d!}f^d(x)\sigma_K^d(\frac{\sum_{t=1}^{n}h_t^d}{n})+o(\frac{\sum_{t=1}^{n}h_t^d}{n})\end{aligned} \tag{16}$$

By selecting a higher order kernel, we can make $\sigma_K^j = \int u^j K(u)du = 0$ for all $j < d$. Hence the resulting bias remains as follows:

$$Bias = \frac{1}{d!}f^d(x)\sigma_K^d(\frac{\sum_{t=1}^{n}h_i^d}{t})+o(\frac{\sum_{t=1}^{n}h_i^d}{t}) \tag{17}$$

and the result of variance in (33) still holds.

Plugging the bias and variance into (11), we derive the general form of risk

$$c_1\frac{(\sum_{t=1}^{n}h_t^d)^2}{n^2}+c_2\frac{\sum_{t=1}^{n}\frac{1}{h_t}}{n^2} \tag{18}$$

Assume $h_t = t^k, k < 0$, take the derivative with respect to k and set it equal to 0, we obtain $k^* = -\frac{1}{2d+1}$, $h^* = n^{-1/2d+1}$, and $Risk = O(n^{-2d/2d+1})$. □

3.2 Optimal Risk for Sequential Kernel Regression

Following the risk analysis for SKDE, similar results can be derived for the sequential kernel regression (SKR). As a simple but illustrative example in the regression setting, for $d = 2$, we based our analysis on the Nadaraya-Waston estimator.

The Sequential Nadaraya-Waston estimator trained on sample size n is

$$\hat{m}_t(x) = \frac{\frac{1}{n}\sum_{t=1}^{n}K_{h_t}(x,X_t)Y_t}{\hat{f}_n(x)} \tag{19}$$

where

$$\hat{f}_n(x) = \frac{1}{n}\sum_{t=1}^{n}K_{h_t}(x,X_t) \tag{20}$$

It remains to show that the risk of SKR has the same form as (18).

Lemma 3.4. *The risk of SKR at time t is:*

$$\begin{aligned} R(\hat{m}_n, m) =& \frac{1}{4}\Big(\int x^2 K(x)dx\Big)^2 \int (m''(x) + 2m'(x)\frac{f'(x)}{f(x)})^2 dx \Big(\frac{(\sum_{t=1}^n h_t^2)^2}{n^2}\Big) \\ &+ \sigma^2 \int K^2(x)dx \int \frac{1}{f(x)}dx \Big(\frac{\sum_{t=1}^n \frac{1}{h_t}}{n^2}\Big) \\ &+ \sigma^2 + o\Big(\frac{\sum_{t=1}^n \frac{1}{h_t}}{n^2}\Big) + o\Big(\frac{(\sum_{t=1}^n h_t^2)^2}{n^2}\Big) \end{aligned} \tag{21}$$

The proof of Lemma 3.4 is shown in Appendix (A.2) using similar bias-variance risk analysis.

3.3 Risk Analysis for Sequential Local Polynomial Regression

The same generalization of Theorem 3.3 in density estimation can be applied in regression setting. Instead of choosing a special kernel to cancel out the lower order terms in the Taylor's series, we leverage the minimax optimality of local polynomial regression as introduced in §2.

Theorem 3.5. *If the true function m has d continuous derivatives and let the bandwidth $h_t = O(t^{-1/2d+1})$, at $t = n$, the order $d-1$ SLPR has the optimal risk $R^* = O(n^{-2d/2d+1})$.*

The proof of Theorem 3.5 is shown in Appendix §A.3.

4 Adapting to Unknown Smoothness

The performance of the sequential estimator (SKDE or SLPR) relies on selecting the right order d of local polynomial (also the constant c). Traditional statistical model selection methods, e.g. the AIC and cross validation, are less desirable in an online scenario due to their computational cost.

In order to maintain a reasonable computational cost, we combine estimators that use different parameters (order d and constant c) through an exponential weighting strategy. Various of useful bounds in terms of the regret are summarized in [2]. The risk bound is also proposed by Yang [13]. Furthermore, according to Yang, combining forecasts with proper weights can outperform the model selection, which often yields an estimator with large variance.

Based on these results, we combine the sequential estimators to adapt to unknown smoothness while keeping the computational cost linear in T and also in the number of expert. Although there exists applications that require large size of experts, unlike other model selection methods, with mixing experts as the top level framework, we can easily leverage on the rising power of paralleling computing and distributed architecture and therefore making this approach practical for those applications.

4.1 Procedure

The Mixing Online Expert (MOE) is the exponential weighting of a set of Sequential Local Polynomial Regression with different orders and constants. The exponential weighting procedure is defined as follows:

Let $\mathbf{C} = \{c_i\}_{1\le i\le m}$, and $\mathbf{D} = \{d_j\}_{1\le j\le n}$.

Define $K = mn$ SLPRs $\mathbf{y} = \{\hat{y}_{i,j}\}_{1\le i\le m, 1\le j\le n}$. At time t, the bandwidth of $\hat{y}_{i,j}$, $h_{i,j}^t = c_i \cdot t^{-1/2d_j+1}$.

The double index i, j is for illustrating the construction of the expert set. In what follows we will use single index k to index the K expert estimators, and t to specify the time steps.

Put $L_{k,t} = \sum_{t=1}^{T}(Y_i - \hat{y}_k)^2$ to be the cumulative loss of estimator k at time t. Let $w_{k,0} = K^{-1}$, and for $t \ge 1$ let

$$w_{k,t} = \frac{\exp(-\eta L_{k,t-1})}{\sum_{k'=1}^{K}\exp(-\eta L_{k',t-1})} \tag{22}$$

where η is a positive constant called the learning rate to be chosen later.

Then the combined estimator at time t is the convex combinations of candidate estimators at time t using the weights at $t-1$:

$$\hat{y}_{,t}^{*} = \sum_{k=1}^{K} w_{k,t-1}\hat{y}_{k,t} \tag{23}$$

Note that the weight at time t can be updated in linear time $O(K)$ using

$$w_{k,t} = \frac{w_{k,t-1}\exp(-\eta(Y_t - \hat{y}_{k,t})^2)}{\sum_{k'=1}^{K} w_{k',t-1}\exp(-\eta(Y_t - \hat{y}_{k',t})^2)} \tag{24}$$

4.2 Performance bound for general convex loss

The performance of the combined estimator is evaluated in comparison to the best individual estimator. The literature of combining experts focuses on deriving bounds on the cumulative loss between the best expert and the combined expert. Existing results (see, e.g. [2] page 14–45) showed that for general convex loss function l, the optimal bound on the cumulative loss is

$$\hat{L}_n \leq \min_{k=1,\ldots,K} L_{k,n} + \frac{\ln K}{\eta} + \frac{\eta}{8} n \tag{25}$$

It follows that when $\eta = \sqrt{\frac{8 \ln K}{n}}$, the additive penalty of the combined expert is $\sqrt{1/2 n \ln K}$.

4.3 Time uniform bounds

Note that the optimal learning rate parameter η in the results above depends on the number of iteration n. When n is very large, a fixed η leads to poor performance for in the earlier predictions. There are several ways to address this issue. First, in [2], they consider a so-called "doubling trick", which partitions time into exponentially increasing lengths. And in each partition, they choose the optimal η for the length of that partition. This solution results in an additive penalty of $\frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{\frac{t}{2}\ln K}$ for all $t = 1, \ldots, T$. Second, [2] also shows that choosing a time-varying $\eta_t = \sqrt{\frac{8 \ln K}{t}}$ yields an additive penalty of $2\sqrt{\frac{t}{2}\ln K} + \sqrt{\frac{\ln K}{8}}$.

A different approach proposed by Freund and Hsu [3] is called Normal Hedging that does not require choosing a learning rate η. Here, we briefly introduce this novel approach:
Define the regret of expert k at time t to be $R_{k,t} = \hat{L}_t - \hat{L}_{k,t}$.
Initially: Set $R_{k,0} = 0, w_{t,0} = \frac{1}{K}$, for $k = 1, \ldots, K$.
For $t = 1, 2, \ldots$:

1. Make prediction: $\hat{y}_t^* = \sum_{k=1}^K w_{k,t-1}\hat{y}_{k,t}$.
2. Update regret: $R_{k,t} = R_{k,t-1} + (l_t - l_{k,t})$, where $l_{k,t} = (Y_t - \hat{y}_{k,t})^2$,and $l_t = (Y_t - \hat{y}_t^*)^2$.
3. Find $c_t > 0$ satisfying $\frac{1}{K}\sum_{k=1}^K \exp\left(\frac{([R_{k,t}]_+)^2}{2c_t}\right) = e$.
4. Update weights: $w_{k,t} \propto \frac{([R_{k,t}]_+)^2}{c_t} \exp\left(\frac{([R_{k,t}]_+)^2}{2c_t}\right)$.

They prove that the cumulative loss of the combined estimator using Normal Hedge is $O(\sqrt{t \ln \frac{1}{\epsilon}} + \ln^2 K)$ worse than that of the ϵKth best expert. In the special case when $\epsilon = \frac{1}{K}$, it is $O(\sqrt{t \ln K} + \ln^2 K)$ worse than the best expert. Another noteworthy advantage of the Normal Hedging is that it assigns zero weight to the expert that has larger cumulative loss than the combined estimator, which reduces the old candidate set to an small subset of effective experts, and thus achieves better performance.

4.4 Tighter bound for squared loss function

When the assumption on the loss function is more restrictive than convexity, tighter bound can be derived. For example, a loss function is $exp-concave$ for a certain $\eta > 0$ if the function $F(z) = e^{-\eta l(z,y)}$ for all y in the outcome space. Then, for $exp-concave$ loss function,

$$\hat{L}_n \leq \min_{k=1,\ldots,K} L_{k,n} + \frac{\ln K}{\eta} \tag{26}$$

To qualify the squared loss as $exp-concave$, we need to pick η to ensure that $e^{-\eta(Y_t - \hat{y}_{k,t})^2}$ is concave in Y_t for all $t = 1, \ldots, T$ and $k = 1, \ldots, K$. Additional assumption needs to be made to bound the output space.

4.5 Risk bound for squared loss function

The cumulative bounds above hold for any realization of data. However, to fit the risk analysis of an individual online kernel estimator, we are more interested in the statistical risk bound. In other words, we seek an oracle type inequality in the form of

$$\mathbb{E}[||\hat{y}^* - f||^2] \leq \min_{k=1,\ldots,K} \mathbb{E}[||\hat{y}_k - f||^2] + \delta(n) \tag{27}$$

where $\delta(n)$ is the additive we receive for the combining strategy. Such risk bound reflects a more comprehensive view of our combined estimator.

Here, we borrow the result from Yang [13], theorem 5, to illustrate the performance of our combined estimator.

First we have to make some assumptions on the output of the estimator and the true outcome. (1) The prediction of individual estimator $\hat{y}_{k,t}$ is bounded between some constant $[-A, A]$; (2) there exists $0 < L < \infty$ such that $\mathbb{E}[\exp|\hat{y}_k - Y|] \leq L$.

Under these two assumptions, according to Theorem 5 of Yang [13], we have the following risk bound for our algorithm using the squared loss function:

$$\frac{1}{n}\sum_{t=1}^{T}\mathbb{E}[(Y_t-\hat{y}_t^*)^2] \le \min_{k=1,\ldots,K}\frac{1}{n}\sum_{t=1}^{T}\mathbb{E}[(Y_t-\hat{y}_t)^2]+\frac{\ln K}{\eta} \tag{28}$$

The (28) above applies to non-stationary sequences. If assuming that the data $(X_i,Y_i)_{i=1,2,\ldots}$ arrives $i.i.d$, as our setting, (28) can be rewritten as:

$$\mathbb{E}[||\hat{y}^*-f||^2] \le \min_{k=1,\ldots,K}\mathbb{E}[||\hat{y}_k-f||^2]+\frac{\ln K}{n\eta} \tag{29}$$

where η is picked to be a small constant depending on A.

Note that the assumption is made only for bounding the estimation output $\hat{y}_k$. As for assumptions on the distribution of Y, other than the strong moment generating function condition (2), no addition assumption is needed.

The risk bound in (29) shows that the risk of our combined estimator is within a constant factor 1 of the estimator using the optimal model plus an error term of $O(n^{-1}\ln K)$. Combining the result of the previous section that the optimal online kernel estimator has the minimax risk, consequently, our combined estimator adapts to that optimal rate at $O(n^{-1}\ln K)$.

5 Experiments

To illustrate the performance of the Mixing Online Experts, we have carried out simulation experiments. In order to demonstrate the adaptation to the unknown smoothness, we have chosen three regression functions to estimate:

$$m_p(x)=20(2x-1)^{p+1}sin(1/(2x-1)), p=2,4,6$$

where m_p has p continuous derivatives, or in C^p. The sample size is 1200 and the noise has variance 0.25. The input space are normalized to $[0,1]$.

In the estimation we have used the Gaussian kernel $K_h(u)=(2\pi)^{-1/2}e^{-u^2/(2h^2)}$. In this setup, the family of order $\mathbf{D}=\{1,2,\ldots,8\}$, and the family of constant contains 20 numbers uniformly chosen from the interval $[0,02,2.5]$:

$$\mathbf{C}=\{c_i=0.02+(2.5-0.02)(i-1)/19|i=1,\ldots,20\}$$

The bandwidth of $\hat{h}(t)$ is defined as

$$\hat{h}_{i,j}(t)=c_i\cdot t^{\frac{1}{2d_j+1}}$$

	Loss	Risk
MOE	202.63	0.004
MBE	200.18	0.0036
Expert (min loss)	201.41	0.00445
Expert (min risk)	201.82	0.00404

Table 1: Loss and risk of MOE, MBE, experts with min loss and min risk

The MOE contains 160 SQLR experts, with degree varying from 1 to 8 and constant varying from 0.02 to 2.5. These experts are indexed by ascending order, and within the same order, are indexed by ascending constants.

For the purpose of performance benchmark, we compare the MOE against the expert with minimum loss, the expert with minimum risk, and also the Mixing Batch Experts (MBE), which employees the batch local polynomial estimates as the experts. Considering that the MBE is too expensive to compute, for experiments involving MBE, we reduce the number of experts to 40.

The performance is summarized in Table 1. It shows that the loss and the risk of the MOE are quite comparable to those of the best experts and the BME.

Figure 1 shows the cumulative loss of the MOE (blue), the MBE (red), the best (green). The loss of the MOE is very close to that of the best online expert; From Table 1, its loss is higher than that of the MBE which is the cost for only updating the bandwidth for the latest data point.

To contrast the runtime of MOE and MBE, Figure 2 shows the linear runtime of the MOE and the quadratic runtime for the MBE.

Figure 3 shows the weights of the experts under three functions with different smoothness. We run experiments on each of the regression functions 10 times and box plot the weights versus expert index. For m_2 in figure 3(a), the weights are centered in the experts with index 21 to 40, which corresponds to $d = 2$. Similarly, for m_4 in figure 3(b) and m_6 in figure 3(c), the experts who have the right order get higher weights. This nicely demonstrates the fact that the MOE adapts to the unknown smoothness.

Hence, the experiments show that while MOE makes trade off between performance and runtime, its performance is quite comparable to that of the MBE and the best experts.

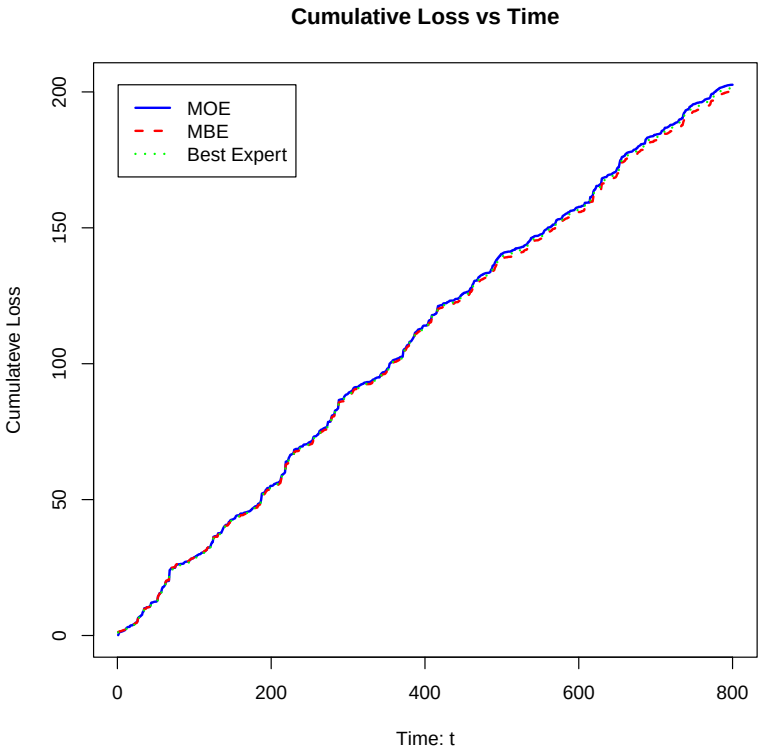


Figure 1: Cumulative loss of the Mixing Online Experts, the Mixing Batch Experts, the best and the worst experts

6 Summary and Conclusions

We proposed the Mixing Online Experts (MOE) as an efficient approach to online prediction. The MOE consists an efficient sequential local polynomial regression procedure, and the mixing expert framework. Our work contains three main contributions. The first contribution is the efficient online estimator, especially the sequential local polynomial regression (SLPR). An order-d SLPR has the desirable runtime complexity of $O(nd^2)$. Similar approach has been investigated by Kristan [8]. But we differ in the way of choosing the variable smoothing parameter. Also they only extended the case of kernel density estimation and does not involve theoretical risk analysis. The second contribution is that we carry out a risk analysis for the sequential estimates, including the sequential local polynomial estimate. The result show the sequential estimates achieve minimax optimal risk $n^{-2d/2d+1}$ when the bandwidth adapts to the optimal bandwidth $O(t^{-1/2d+1})$ at each time step. The third contribution is the smoothness adaptation achieved by the mixing

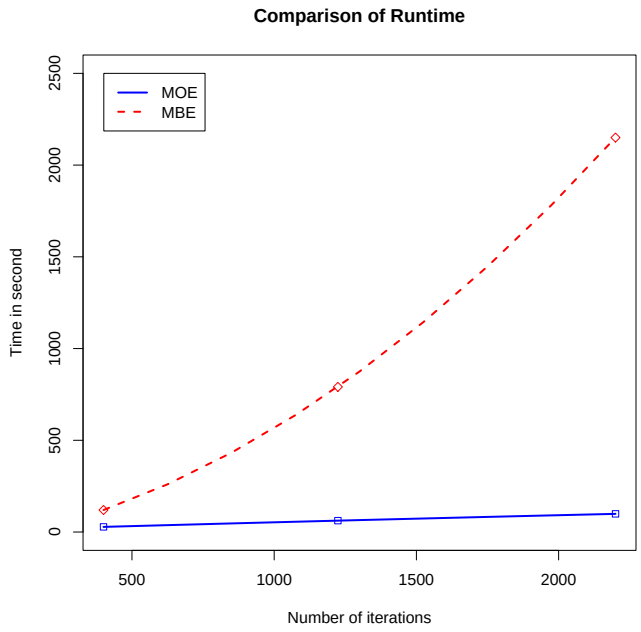


Figure 2: Runtime of MOE and MBE vs Sample size

expert framework. We investigated existing regret bound [2] and risk bound [13] with their restrictions. Under mild assumptions on the loss function, the risk bound by Yang [13] shows that we can adapt to the optimal model at the rate of $O(n^{-1} \ln K)$. The experimental results confirm our theoretical analysis in 2 aspects. First, both the cumulative loss and risk of our Mixing Online Experts (MOE) are comparable to those best sequential model, and the batch model. Second, the MOE assigns high weights on the experts that have the correct order of smoothness. Hence, the MOE adapts to both the data size and the unknown smoothness of the true function. Last but not least, the linear runtime makes MOE a practical approach for sequential regression.

One important future work could be adapting MOE to spatially inhomogeneous smoothness. Spatial adaptation is a powerful feature for estimation in practice, where the true function is often spatial inhomogeneous. Lepski et al. [9] proposed a variable bandwidth selector for kernel estimation that achieve optimal rates of convergence over Besov Classes. But for sequential regression, the problem remains unsolved. For the MOE, although the mixing expert framework allows the estimate to adapt to arbitrary optimal model, it is unpractical to have experts on every point of the input space. Not only is it computationally infeasible, but it also requires too many data

to train the experts.

More sophisticated online learning problems can be investigated in the presence of this online estimator which is both efficient for practical experiments and amenable for theoretical analysis. In particular, for the next step, we are interested in multi-task learning with constrained resources. The goal is to optimally allocate limited samples for multiple MOEs with different learning tasks such that the overall loss is minimized.

A Proofs of the results

A.1 Proof of Lemma 3.1

Proof. We first compute the bias of $\hat{f}_n$ at a point x.

$$\begin{aligned}
\mathbb{E}[[]\hat{f}_n(x)] &= \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}[[]K_{h_t}(x, X_t)] \\
&= \frac{1}{n}\sum_{t=1}^{n}\int K(u)f(x-h_t u)du \\
&= \frac{1}{n}\sum_{t=1}^{n}[f(x) + \frac{h_t^2}{2}f''(x)\int K(u)u^2 du + o(h_t^2)] \\
&= f(x) + \frac{1}{2}f''(x)\sigma_K^2 \cdot \left(\frac{\sum_{t=1}^{n} h_t^2}{n}\right) + o\left(\frac{\sum_{t=1}^{n} h_t^2}{n}\right)
\end{aligned} \tag{30}$$

Thus, the bias of $\hat{f}_n(x)$ is

$$\frac{1}{2}f''(x)\sigma_K^2 \cdot \left(\frac{\sum_{t=1}^{n} h_t^2}{n}\right) + o\left(\frac{\sum_{t=1}^{n} h_t^2}{n}\right)$$

where $\sigma_K = \int x^2 K(x)dx$.

To compute the variance of $\hat{f}_n(x)$, we need to compute $Var[K_{h_t}(x, X_t)]$:

$$\begin{aligned}
\mathbb{E}[]\,[K_{h_t}^2(x, X_t)] &= \frac{1}{h_t^2}\int K^2\left(\frac{x-v}{h_t}\right)f(v)dv \\
&= \frac{1}{h_t}\left(\int K^2(u)[f(x) + O(h_t^2)]du\right) \\
&= \frac{1}{h_t}f(x)\int K^2(u)du + o(\frac{1}{h_t})
\end{aligned} \tag{31}$$

Since the term $\mathbb{E}[][K_{h_t}(x, X_t)]$ computed above is $f(x) + O(h_t^2)$, we can include it in $O(1)$.

$$\begin{aligned} Var[K_{h_t}(x, X_t)] &= \mathbb{E}[][K_{h_t}^2(x, X_t)] - \mathbb{E}[][K_{h_t}^2(x, X_t)]^2 \\ &= \frac{1}{h_t} f(x) \int K^2(u)du + o(\frac{1}{h_t}) - O(1) \\ &= \frac{1}{h_t} f(x) \int K^2(u)du + o(\frac{1}{h_t}) \end{aligned} \tag{32}$$

By rule of independence, the variance of $\hat{f}_n(x)$ is:

$$\begin{aligned} Var(\hat{f}_n(x)) &= \frac{1}{t^2} \sum_{t=1}^{n} Var(K_t(x, X_t)) \\ &= \frac{f(x) \int K^2(u)du}{n^2} \cdot (\sum_{t=1}^{n} \frac{1}{h_t}) + o(\frac{\sum_{t=1}^{n} \frac{1}{h_t}}{n^2}) \end{aligned} \tag{33}$$

Hence, plugging the bias and variance into (11), we have proved Lemma 3.1. □

A.2 Proof of Lemma 3.4

Proof. Note that the denominator in (19) is the SKDE for x, which approximates to the true density function $f(x)$.

Thus, using bias variance decomposition we can write the risk of the regression function as:

$$R(\hat{m}_t(x), m(x)) = Bias^2(\hat{m}_t(x)) + Var(\hat{m}_t(x)) + \sigma^2 \tag{34}$$

where $Y_t = m(X_t) + \epsilon_t$, $\epsilon_t \ N(0, \sigma^2)$. In the context of kernel regression, the bias and variance are conditional on the observation of X.

First, we compute the bias.

$$\begin{aligned} \mathbb{E}[]K_h(x, X_t)Y_t] =& \mathbb{E}[]K_h(x, X_t)m(X_t)] + \mathbb{E}[]\epsilon_t K_h(x, X_t)] \\ =& \int K(u)m(x - h_t u)f(x - h_t u)du \\ =& \int K(u)[m(x) + h_t u m'(x) + \frac{h_t^2 u^2}{2} m''(x) + O(h_t^2)] \\ & \cdot [f(x) - h_t u f'(x) + o(h_t^2)]du \\ =& m(x)f(x) + \frac{h_t^2}{2}\sigma_K^2 \cdot [m''(x)f(x) + 2m'(x)f'(x)] + o(h_t^2) \end{aligned} \tag{35}$$

$$\mathbb{E}[[]\hat{m}_t(x)] = m(x) + \frac{\sum_{t=1}^{n} h_t^2}{2t}\sigma_K^2 \cdot [m''(x) + 2m'(x)\frac{f'(x)}{f(x)}] + o\left(\frac{\sum_{t=1}^{n} h_t^2}{t}\right) \quad (36)$$

Subtracting $m(x)$ from (36), we showed the bias of $\hat{m}(x)$ is at $t = n$ is

$$\frac{\sum_{t=1}^{n} h_t^2}{2n}\int u^2 K(u)du \cdot [m''(x) + 2m'(x)\frac{f'(x)}{f(x)}] + o\left(\frac{\sum_{t=1}^{n} h_t^2}{n}\right) \quad (37)$$

By a similar calculation, the variance of the online kernel estimator is:

$$\begin{aligned} Var[\hat{m}_t(x)] =& \frac{1}{n^2 f^2(x)}\sum_{t=1}^{n} Var[K_{h_t}(x, X_t)Y_t] \\ =& \frac{\sigma^2}{n^2 f^2(x)}\sum_{t=1}^{n}[K_{h_t}(x, X_t)]^2 \\ =& \frac{\sigma^2}{n^2 f^2(x)}\int \frac{1}{h_t^2}K^2\left(\frac{x-u}{h_t}\right)du \\ =& \frac{\sigma^2 \int K^2(u)du}{f(x)}\frac{\sum_{t=1}^{n}\frac{1}{h_t}}{n^2} + o(\frac{\sum_{t=1}^{n}\frac{1}{h_t})}{n^2}) \end{aligned} \quad (38)$$

Hence, plugging the bias and variance into (34), we proved Lemma 3.4. □

A.3 Proof of Theorem 3.5

Notice that the only difference between batch local polynomial regression (5) and the SLPS (6) is in the diagonal bandwidth matrix W and W_n.

The general idea of extending the existing result for batch to sequential local polynomial regression is to bound the bias and variance of $(\mathbf{X^T W_n X})^{-1}$ with $(\mathbf{X^T W X})^{-1}$ and $\mathbf{X^T W_n y}$ with $\mathbf{X^T W y}$.

It follows from the solution (6) that the conditional bias and variance of $\hat{\beta}$ are

$$\mathbb{E}[(]\hat{\beta}|\mathbb{X}) = \beta + (\mathbf{X^T W_n X})^{-1}\mathbf{X^T W_n}(\mathbf{m} - \mathbf{X}\beta) \quad (39)$$

and

$$Var(\hat{\beta}|\mathbb{X}) = (\mathbf{X^T W_n X})^{-1}(\mathbf{X^T \Sigma_n X})(\mathbf{X^T W_n X})^{-1} \quad (40)$$

where $\mathbf{m} = \{m(X_1), \ldots, m(X_n)\}^T, \beta = \{m(x_0), \ldots, m^d(x_0)/(d-1)!\}^T$, and $\Sigma_n = diag\{K_{h_t}^2(X_t, x_o)\sigma_n^2\}_{1 \le t \le n}$.

In (7), we denote by S_n the $d \times d$ matrix $(\mathbf{X^T W_n X})$. Here we denote $(\mathbf{X^T W X})$ as $S_{n_{usual}}$, with respect to the usual local polynomial regression case with fixed bandwidth h.

Similarly, we denote S_n^* to be the $d \times d$ matrix $(\mathbf{X^T \Sigma_n X})$; and $S^*_{n_{usual}}$ to be $(\mathbf{X^T W X})$.

For the simplicity of the derivation, we need some more useful notations before diving in to the bias variance analysis. First, we denote the moments of K and K^2 respectively by

$$\mu_j = \int u^j K(u)du \quad \nu_j = \int u^j K^2(u)du$$

Second, let S and S^* denote the Hankel matrix with the kernel moments:

$$S = (\mu_{j+l})_{0\le j,l\le d-1} \quad S^* = (\nu_{j+l})_{0\le j,l\le d-1}$$

Following the bias-variance guideline, we need to show that if we choose $h_t = O(t^{-k})$ for some $k \in (0,1)$, then $Var(\hat{m}) = O(n^{k-1})$, and $Bias(\hat{m}) = O(n^{-dk})$ respectively, shown in §A.3.1 and §A.3.2. This concludes that the optimal $k^* = \frac{1}{2d+1}$ and the optimal risk is $O(n^{-2d/2d+1})$.

A.3.1 Variance bound for SLPR

The (i,j) entry of $S_{n_{usual}}$ can be approximated its mean plus additional error.

$$S_{n_{usual}}(i,j) = nh^{i+j}\mu_{i+j}\{f(x_0) + o_P(1)\} \tag{41}$$

It follows that the matrix form of $S_{n_{usual}}$ is:

$$S_{n_{usual}} = nf(x_0)HSH\{1 + o_P(1)\} \tag{42}$$

where S is defined in (A.3) and $H = diag\{h^j\}_{(0\le j\le d-1)}$, and h is the constant bandwidth.

Now, we will show that if the bandwidth for the usual case $S_{n_{usual}}$ is $h = n^{-k}$ for some $0 < k < 1$, and if we choose the online bandwidth $h_t = t^{-k}, 0 \le t \le n$, then S_n is bounded by $S_{n_{usual}}$ element-wise within a constant factor independent of n.

By integral approximation,

$$n \times n^{-(i+j)k} < \sum_{t=1}^{n} h_t^{i+j} < \int_0^n t^{-(i+j)k}dt = n\frac{1}{1-(i+j)k}n^{-(i+j)k} \tag{43}$$

Hence there exists $c_{i,j} \in (1, \frac{1}{1-(i+j)k})$ such that $c_{i,j} * S_n(i,j) = S_{n_{usual}}(i,j)$, provided that $0 < 1-(i+j)k < 1$.

Using the above fact and (42), we let C_1 be the $d \times d$ scalar matrix where $C_{1_{i,j}}$ is some constant bounded within $(1, \frac{1}{1-(i+j)k})$. Hence, $S_n = X^T W_n X$ can be rewritten as:

$$S_n = nf(x_0)H(C_1 \cdot S)H\{1 + o_P(1)\} \tag{44}$$

where $\cdot$ is the element-wise multiplication.

Similarly for $S_n^* = X^T \Sigma_n X$, we have $S^*_{n_{usual}}(i,j) = nh^{i+j-1}\nu_{i+j} f(x_0)\sigma^2 H S^* H\{1+ o_P(1)\}$. By a similar calculation, we can bound S_n^*:

$$S_n^* = nh^{-1} f(x_0)\sigma^2 H(C_2 \cdot S^*)H\{1 + o_P(1)\} \tag{45}$$

where C_2 is $d \times d$ scalar matrix with entry $C_{2_{i,j\,(1\le i+j<2d-2)}}$ bounded within $(1, \frac{1}{1-(i+j-1)k})$, and $C_{2_{0,0}}$ bounded within $(\frac{1}{1+k}, 1)$.

It follows immediately, that:

$$\begin{aligned} Var(\hat{\beta}|\mathbb{X}) &= S_n^{-1} S_n^* S_n^{-1} \\ &= \frac{\sigma^2}{nf(x_0)} H^{-1}(C_1 \cdot S)^{-1}(C_2 \cdot S^*)(C_1 \cdot S)^{-1} H^{-1}\{1 + o_P(1)\} \end{aligned} \tag{46}$$

$$\begin{aligned} Var(\hat{m}|\mathbb{X}) &= Var(e_0^T \hat{\beta}) \\ &= \frac{\sigma^2}{nf(x_0)h} e_0^T (C_1 \cdot S)^{-1}(C_2 \cdot S^*)(C_1 \cdot S)^{-1} e_0\{1 + o_P(1)\} \end{aligned} \tag{47}$$

Given that $h = n^{-k}$, we derived that

$$Var(\hat{m}|\mathbb{X}) = O(n^{k-1}) \tag{48}$$

A.3.2 Bias bound for SLPR

Assume d is even, by Taylor expansion, the conditional bias $S_n^{-1} X^T W(m - X\beta)$ of $\hat{\beta}$ can be written as

$$\begin{aligned} Bias(\beta|\mathbb{X}) &= S_n^{-1} X^T W_n \left[\beta_d (X_t - x_0)^d + o_P\{(X_t - x_0)^d\}\right]_{1\le t\le n} \\ &= \beta_d S_n^{-1} X^T W_n [(X_t - x_0)^d]_{1\le t\le n}\{1 + o_P(1)\} \\ &= \beta_d S_n^{-1} nf(x_0) h^{-d} H(C_{1[d]} \cdot [\mu_j]_{d-1\le j\le 2d-1})\{1 + o_P(1)\} \end{aligned} \tag{49}$$

where $\left[\mu_j\right]_{d-1\le j\le 2d-1} = [\mu_{d-1},\ldots,\mu_{2d-1}]^T$, $C_{1[d]}$ is the dth column of the scalar matrix C_1, and $\cdot$ denotes the point-wise multiplication.

Plug (44) in (49) we get:

$$Bias(\beta|\mathbb{X}) = \beta_d h^{-d} H^{-1}(C_1 \cdot S)^{-1} \cdot (C_{1[d]} \cdot [\mu_j]_{d-1\le j\le 2d-1})\{1 + o_P(1)\}$$

Hence, as we choose $h = n^{-k}$, the conditional bias of $\hat{m} = e_0^T\hat{\beta}$ is

$$\begin{aligned} Bias(\hat{m}|\mathbb{X}) &= \beta_d n^{-dk} e_0^T H^{-1}(C_1 \cdot S)^{-1}(C_{1[d]} \cdot [\mu_j]_{d-1\le j\le 2d-1})\{1 + o_P(1)\} \\ &= \beta_d n^{-dk} e_0^T((C_1 \cdot S)^{-1}(C_{1[d]} \cdot [\mu_j]_{d-1\le j\le 2d-1}))\{1 + o_P(1)\} \\ &= O(n^{-dk}) \end{aligned} \tag{50}$$

References

[1] Florentina Bunea and Andrew Nobel. Sequential procedures for aggregating arbitrary estimators of a conditional mean. *Information Theory, IEEE Transactions on*, 54:1725–1735, 2008.

[2] Nicoló Cesa-Bianchi and Gábor Lugosi. *PREDICTION LEARNING AND GAMES*. Cambridge University Press, 2006.

[3] Kamalika Chaudhuri, Yoav Freund, and Daniel Hsu. A parameter-free hedging algorithm. *NIPS*, 2009.

[4] Jianqing Fan and Irene Gijbels. Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation. *Journal of the Royal Statistical Society. Series B(Methodological)*, 57(2):371–394, 1995.

[5] Jianqing Fan and Irene Gijbels. *Local Polynomial Modelling and Its Applications*, pages 57–105. Chapman and Hall/CRC, 1996.

[6] Jianqing Fan, Irene Gijbels, Theo Gasser, Michael Brockmann, and Joachim Engel. Local polynomial fitting: A standard for nonparametric regression, 1993.

[7] Lásló Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, 2002.

[8] Matej Kristan, Danijel Skocaj, and Ales Leonardis. Online kernel density estimation for interactive learning. *Image and Vision Computing*, 28:1106–1116, 2010.

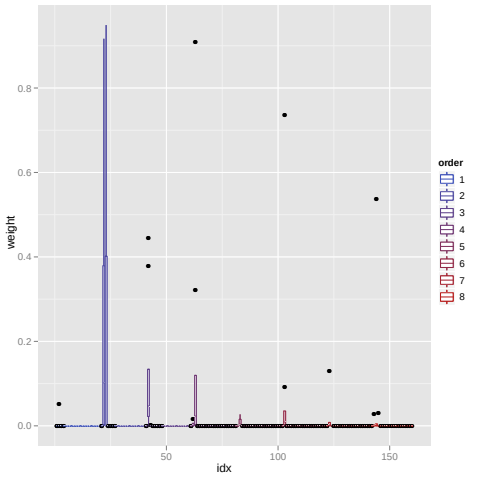
[9] Oleg Lepski, Enno Mammen, and Vladimir Spokoiny. Optimal spatial adaptation to inhomogeneous smoothness: An approach based on kernel estimates with variable bandwidth selectors. *The Annals of Statistics*, 25(3):929–947, 1997.

[10] David Ruppert, Simon J. Sheather, and Matthew P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432), 1995.

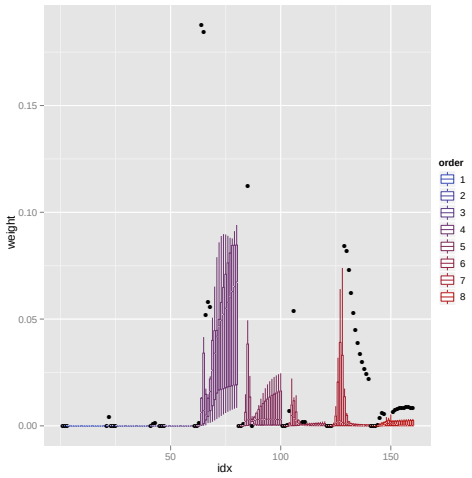
[11] Ansgar Steland. Sequential data-adaptive bandwidth selection by cross validation for nonparametric prediction, 2010.

[12] Larry Wasserman. *All of Nonparametric Statistics*, pages 54–57. Springer, 2005.

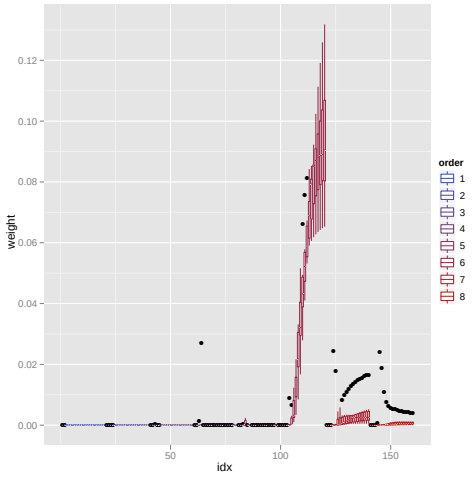
[13] Yuhong Yang. Combining forecasting procedures: Some theoretical results. *Econometric Theory*, 20:176–222, 2004.



(a) weights for m_2 estimation



(b) weights for m_4 estimation



(c) weights for m_6 estimation

Figure 3: Weights of the 160 experts on estimation for m_2, m_4, m_6