Fast Function to Function Regression

Junier B. Oliva[†] **Willie Neiswanger**[†] **Barnabás Póczos**[†] **Eric Xing**[†] **Hy Trac**[*] **Shirley Ho**[*] **Jeff Schneider**[‡]
[†]Machine Learning Department [‡]Robotics Institute [*]Department of Physics
Carnegie Mellon University

Abstract

We analyze the problem of regression when both input covariates and output responses are functions from a nonparametric function class. Function to function regression (FFR) covers a large range of interesting applications including time-series prediction problems, and also more general tasks like studying a mapping between two separate types of distributions. However, previous nonparametric estimators for FFR type problems scale badly computationally with the number of input/output pairs in a data-set. Given the complexity of a mapping between general functions it may be necessary to consider large data-sets in order to achieve a low estimation risk. To address this issue, we develop a novel scalable nonparametric estimator, the Triple-Basis Estimator (3BE), which is capable of operating over data-sets with many instances. To the best of our knowledge, the 3BE is the first nonparametric FFR estimator that can scale to massive data-sets. We analyze the 3BE's risk and derive an upper-bound rate. Furthermore, we show an improvement of several orders of magnitude in terms of prediction speed and a reduction in error over previous estimators in various real-world data-sets.

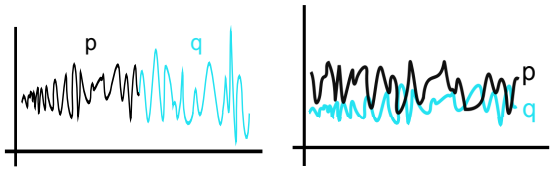
1 INTRODUCTION

Modern data-sets are growing not only in quantity of instances but the instances themselves are growing in complexity and dimensionality. The goal of this paper is to perform regression with data-sets that are massive in both the number of instances and also in the complexity of instances; specifically, we consider functional data. We study function to function regression (FFR) where one aims to learn a mapping f that takes in a general input functional covariate $p : \mathbb{R}^l \mapsto \mathbb{R}$ and outputs a functional response $q = f(p) : \mathbb{R}^k \mapsto \mathbb{R}$. In general, functions are infinite dimensional objects; hence, the problem of FFR is not immediately solvable by traditional regression methods on finite vectors. Furthermore, unlike with typical regression problems, neither the covariate nor the response will be directly observed (since it is infeasible to directly observe functions). Previous nonparametric estimators for FFR do not scale computationally to large data-sets. However, large data-sets are often needed to achieve a low risk; to mitigate this issue we introduce the Triple-Basis Estimator (3BE).

The FFR framework is quite general and includes many interesting problems. For instance, one may consider input/output functions that are probability distribution functions (pdfs). An example of a financial domain related FFR problem with density functions is learning the mapping that takes in the pdf of stock prices in a specific industry and outputs the pdf of stock prices in another industry. Additionally, in cosmology one may be interested in regressing a mapping that takes in the pdf of simulated particles from a computationally inexpensive but inaccurate simulation and outputs the corresponding pdf of particles from a computationally expensive but accurate simulation. In essence, one would be enhancing the inaccurate simulation using previously seen data from accurate simulations. There are also many non-distributional FFR problems. For example, one may view foreground/background segmentation as a FFR problem that maps an image's p function to a segmentation's q function, where $p(x, y)$ is a function that takes in a pixel's (x, y) position and outputs the corresponding pixel's intensity, and $q(x, y)$ is function that takes in a pixel's position and outputs 1 if the pixel is in the foreground and 0 otherwise.

Moreover, several time-series tasks may be posed in the FFR framework (see Figure 1). Suppose, for example, that one is interested in predicting the next unit interval of a time-series given the previous unit interval; then, one may frame this as a FFR problem by letting input functions $p : [0, 1] \mapsto \mathbb{R}$ be the function representing the time-series during the first unit interval and output func-

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.



(a) Forward Prediction (b) Co-occurring Series

Figure 1: (a) One may consider trying to predict a later portion of a time-series when given the prior portion of a time-series as a FFR problem. (b) One may try to predict one co-occurring time-series when given another.

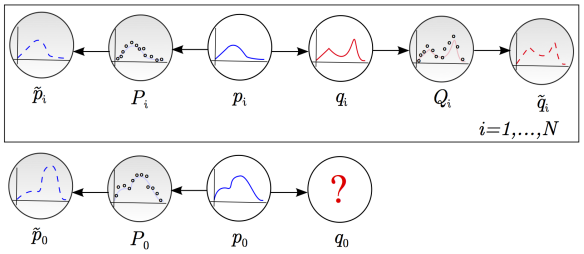


Figure 2: We observe a data-set of input/output functional observation pairs $\{(P_i, Q_i)\}_{i=1}^N$, where P_i, Q_i are some inexact observations of functions p_i and q_i such as a set of noisy function evaluations at uniformly distributed points. P_i, Q_i then are used to make function approximations $\tilde{p}_i, \tilde{q}_i$, which in turn are used to predict the response q_0 for a unseen query input function p_0.

tions $q : [0, 1] \mapsto \mathbb{R}$ be the function representing the time-series during the next unit interval (Figure 1(a)). A related problem is that of predicting co-occurring functions (Figure 1(b)). An interesting application of predicting co-occurring functions is with motion capture data, where one may be interested in predicting the movement of joints that are occluded given the movement of observed joints.

As stated previously, the problem of FFR boils down to the study of a mapping between infinite dimensional objects. Thus, the regression task would benefit greatly from learning on data-sets with a large number of input/output pairs. However, many nonparametric estimators for regression problems *do not scale well* in the number of instances in a data-set. Thus, if the number of instances is in the many thousands, millions, or even more, then it will be infeasible to use such estimators. This leads to a paradox: one wants many instances in a data-set in order to effectively learn the FFR mapping, but one also wants a low number of instances in order to avoid a high computational cost. We resolve this issue through the 3BE, which we will show can perform FFR in a scalable manner.

The data-sets we consider are as follows. Since general functions are infinite dimensional we cannot work over a data-set $\bar{\mathcal{D}} = \{(p_i, q_i)\}_{i=1}^N$ where $q_i = f(p_i)$. Instead we shall work with a data-set of instances that are (inexact) observation pairs from input/output functions $\mathcal{D} = \{(P_i, Q_i)\}_{i=1}^N$ where P_i, and Q_i are some form of empirical observations from p_i and q_i (see Figure 2). For example, one may consider the functional observations to be a set of n noisy function evaluations at uniformly distributed points, or a sample of n points drawn from p and q respectively (when p, q are distributions). Using $\mathcal{D}$ we will make an estimate of $\bar{\mathcal{D}}$ as $\tilde{\mathcal{D}} = \{(\tilde{p}_i, \tilde{q}_i)\}_{i=1}^N$ where $\tilde{p}_i, \tilde{q}_i$ are functional estimates created using P_i, Q_i respectively. The task then is to estimate $q_0 = f(p_0)$ as $\hat{q}_0 = \hat{f}(\tilde{p}_0)$ when given a functional observation, P_0, of an unseen function p_0.

Our approach will be as follows. First, we convert the infinite dimensional task of estimating the output function q_0 into a finite dimensional problem by projecting q_0 into a finite number of basis functions (focusing on the crucial characteristics of q_0, roughly speaking). Then, to estimate the projections onto the basis functions we embed the input functions into a nonlinear space where linear operations are approximately evaluations of a nonlinear mapping f in a broad function class. Finally, f is estimated by minimizing the empirical risk of a linear operation in the nonlinear embedding of input functions for predicting the basis projections of output functions in a data-set.

Our Contribution We develop the Triple-Basis Estimator (3BE), a novel nonparametric estimator for FFR that scales to large data-sets. The 3BE is the first estimator of its kind, allowing one to regress functional responses given functional covariates in massive data-sets. Furthermore, we analyze the L_2 risk of the 3BE under nonparametric assumptions. Lastly, we show an improvement of several orders of magnitude over existing estimators in terms of prediction time as well as a reduction in error in various real-world data-sets.

2 RELATED WORK

A previous nonparametric FFR estimator was proposed by Kadri et al. (2010). Kadri et al. (2010) attempt to perform FFR on a functional RKHS. That is, if we consider $\mathcal{F}$ as a functional Hilbert space, where $f \in \mathcal{F}$ is such that $f : \mathcal{G}_x \mapsto \mathcal{G}_y$, then f is estimated by $f^* = \arg\min_{\hat{f}} \sum_{i=1}^N \|q_i - \hat{f}(p_i)\|_{\mathcal{G}_y}^2 + \lambda \|f\|_{\mathcal{F}}^2$. However, when each function is observed though n noisy function evaluations this estimator will require the inversion of a $Nn \times Nn$ matrix, which will be computationally infeasible for data-sets of even a modest size.

In addition, Oliva, Póczos, and Schneider (2013) provide an estimator for doing FFR, and analyze its risk for the special case where both input and output functions are probability distribution functions. The estima-

tor, henceforth referred to as the linear smoother estimator (LSE), works as follows when given a training data-sets of $\mathcal{D} = \{(P_i, Q_i)\}_{i=1}^N$ of empirical functional observations and $\tilde{\mathcal{D}} = \{(\tilde{p}_i, \tilde{q}_i)\}_{i=1}^N$ of function estimates and a function estimate $\tilde{p}_0$ of a new query input function:

$$\hat{f}(\tilde{p}_0) = \sum_{i=1}^{N} W(\tilde{p}_i, \tilde{p}_0)\tilde{q}_i \text{ where} \tag{1}$$

$$W(\tilde{p}_i, \tilde{p}_0) = \begin{cases} \frac{K(D(\tilde{p}_i, \tilde{p}_0))}{\sum_{j=1}^{N} K(D(\tilde{p}_j, \tilde{p}_0))} & \text{if } \sum_j K(D(\tilde{p}_j, \tilde{p}_0)) > 0 \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

Here $K : \mathbb{R} \to [0, \infty)$ is taken to be a symmetric kernel with bounded support, and D is some metric over functions. However, while such an estimator is useful for smaller FFR problems, it may not be used in larger data-sets. Clearly, the LSE must perform a kernel evaluation with all input functions in one's data-set to produce a prediction, leading to a total computational cost of $\Omega(Nn)$ when considering the cost of computing metrics $D(\tilde{p}_j, \tilde{p}_0)$ when $|P_i| \asymp |Q_i| \asymp n$. This implies, for example, that obtaining estimates for each training instance scales as $\Omega(nN^2)$, which will be prohibitive for big data-sets.

Previous work for nonparametric estimation in large data-sets with functional inputs includes work by Oliva et al. (2014a). There an estimator was proposed for scalable learning of a distribution input covariate to real-value output response regression problem. We note however that it is not immediately clear how to achieve a scalable estimator for regression functional responses with functional covariates, nor how to analyze such an estimator's risk since general functional responses are infinite dimensional.

We note further that work has been done in linear models for FFR (Ramsay and Silverman 2006; Oliva et al. 2014b). However, such models work over a strong assumption on the linearity of the mapping f, and will not be able to capture non-linear mappings. Moreover, FFR is a specific case of general functional analysis (Ramsay and Silverman 2006; Ferraty and Vieu 2006; Ramsay and Silverman 2002).

3 MODEL

We expound upon our model of input/output functions and the mapping between them. Later, we introduce the 3BE and analyze its risk for the case when one has a data-set of pairs of input/output functional observations that are a set of noisy function evaluations at uniformly distributed points. However, the following is generalizable for the case where one observes function evaluations at a fixed grid of points or function observations of samples from distributions. In short, we assume smooth input/output functions that are well approximated by a finite number of basis functions. Further, we consider a nonparametric mapping between them where the projection of the output function onto each basis function may be written as an infinite linear combination of RBF kernel evaluations between the input function and unknown functions (see below).

We take our data-set to be input/output empirical function observation pairs:

$$\mathcal{D} = \{(P_i, Q_i)\}_{i=1}^N \text{ where} \tag{3}$$

$$P_i = \{p_i(u_{ij}) + \epsilon_{ij}\}_{j=1}^{n_i}, \; Q_i = \{q_i(v_{ij}) + \xi_{ij}\}_{j=1}^{m_i}, \tag{4}$$

with sample points $u_{ij} \overset{iid}{\sim} \text{Unif}([0,1]^l)$, $v_{ij} \overset{iid}{\sim} \text{Unif}([0,1]^k)$, and noise $\epsilon_{ij} \overset{iid}{\sim} \Xi_\epsilon$, $\xi_{ij} \overset{iid}{\sim} \Xi_\xi$. With error distributions Ξ_ξ, Ξ_ϵ, s.t. $\mathbb{E}[\epsilon_{ij}] = \mathbb{E}[\xi_{ij}] = 0$, $\text{Var}[\epsilon_{ij}], \text{Var}[\xi_{ij}] \leq \varsigma < \infty$. Furthermore, $p_i \in \mathcal{I}$, $p_i : [0,1]^l \mapsto \mathbb{R}$, $q_i \in \mathcal{O}$, $q_i : [0,1]^k \mapsto \mathbb{R}$, $q_j = f(p_j)^*$, and $p_i \overset{iid}{\sim} \Phi$ where $\mathcal{I}$ and $\mathcal{O}$ are some class of input/output functions and Φ is some measure over $\mathcal{I}$. Furthermore, we shall assume that $n_i \asymp n$ and $m_i \asymp m$. We shall use $\mathcal{D}$ to make estimates of the true input/output functions $\tilde{D} = \{(\tilde{p}_i, \tilde{q}_i)\}_{i=1}^N$, which will then be used to estimate the output function q_0 corresponding to an unseen input function p_0.

3.1 Basis Functions and Projections

Let $\{\varphi_i\}_{i\in\mathbb{Z}}$ be an orthonormal basis for $L_2([0,1])$. Then, the tensor product of $\{\varphi_i\}_{i\in\mathbb{Z}}$ serves as an orthonormal basis for $L_2([0,1]^d)$; that is, the following is an orthnormal basis for $L_2([0,1]^d)$:

$$\{\varphi_\alpha\}_{\alpha\in\mathbb{Z}^d} \quad \text{where} \quad \varphi_\alpha(x) = \prod_{i=1}^{d} \varphi_{\alpha_i}(x_i), \; x \in [0,1]^d.$$

So we have that $\forall \alpha, \rho \in \mathbb{Z}^d$, $\langle \varphi_\alpha, \varphi_\rho \rangle = I_{\{\alpha=\rho\}}$. Let $h \in L_2([0,1]^d)$, then

$$h(x) = \sum_{\alpha\in\mathbb{Z}^d} a_\alpha(h)\varphi_\alpha(x) \quad \text{where} \tag{5}$$

$$a_\alpha(h) = \langle \varphi_\alpha, h \rangle = \int_{[0,1]^d} \varphi_\alpha(z)h(z)\mathrm{d}z \in \mathbb{R}. \tag{6}$$

As previously mentioned, a data-set of estimated input/output function pairs, $\tilde{\mathcal{D}} = \{(\tilde{p}_i, \tilde{q}_i)\}_{i=1}^N$, will be constructed from the data-set of input/output function evaluation sets $\mathcal{D} = \{(P_i, Q_i)\}_{i=1}^N$. Suppose function h has a corresponding set of evaluations $H = \{y_j = h(u_j) + \epsilon_j\}_{j=1}^r$ where $u_j \overset{iid}{\sim} \text{Unif}([0,1]^d)$ and $\mathbb{E}[\epsilon_j] = 0$, $\mathbb{E}[\epsilon_j^2] < \infty$.

*Similarly, one may consider a model $q_i(x) = [f(p_i)](x) + \zeta w_i(x)$, where w_i is a standard Wiener process. This however will be akin to adding variance to our noisy function evaluations, hence we omit w_i for simplicity.

Then, $\tilde{h}$, the estimate of h, will be as follows:

$$\tilde{h}(x) = \sum_{\alpha \in M} a_\alpha(H)\varphi_\alpha(x) \quad \text{where} \tag{7}$$

$$a_\alpha(H) = \frac{1}{r}\sum_{j=1}^{r} y_j \varphi_\alpha(u_j), \tag{8}$$

and M is a finite set of indices for basis functions.

3.1.1 Cross-validation

In practice, one would choose indices M in (7) through cross-validation. The number of projection coefficients one chooses will depend on the smoothness of the function h as well as the number of points in H. Typically, a larger $|i|$ will correspond to a higher frequency 1-dimensional basis function φ_i; thus, a natural way of selecting M is to consider sets

$$M_t = \{\alpha \in \mathbb{Z}^d : \|\alpha\|_2 \leq t\} \tag{9}$$

with $t \in [0, \infty)$. One would then choose the value of t (setting $M = M_t$) that minimizes a loss, such as the mean squared error between $\tilde{h}(u_i)$ and y_i. We shall see below that considering M_t in this manner corresponds to a smoothness assumption on the class of input/output functions.

3.2 Function to Function Mapping

Let $p \sim \Phi$ and $q = f(p)$, we have that:

$$q(x) = [f(p)](x) = \sum_{\alpha \in \mathbb{Z}^k} a_\alpha(f(p))\varphi_\alpha(x) \tag{10}$$

$$= \sum_{\alpha \in \mathbb{Z}^k} f_\alpha(p)\varphi_\alpha(x), \tag{11}$$

where $f_\alpha(p) = a_\alpha(f(p))$. Hence, we may think of $f : \mathcal{I} \mapsto \mathcal{O}$ as consisting of countably many functions $\{f_\alpha \mid f_\alpha : \mathcal{I} \mapsto \mathbb{R},\ \alpha \in \mathbb{Z}^k\}$, where each f_α is responsible for the mapping of p to the projection of q on to φ_α. We take f_α functions to be a nonparametric linear smoother on a possibly infinite set of functions weighted by a kernel:

$$f_\alpha(p) = \sum_{i=1}^{\infty} \theta_{\alpha i} K_\sigma(g_{\alpha i}, p) \text{ where} \tag{12}$$

$$\theta_{\alpha i} \in \mathbb{R},\ g_{\alpha i} \in \mathcal{I}. \tag{13}$$

We shall consider the following class of functions:

$$\mathcal{F}_\sigma = \{f\ :\ \forall \alpha \in \mathbb{Z}^k\ \|\theta_\alpha\|_1 \leq B_\alpha, f_\alpha \text{ as in (12)}\}. \tag{14}$$

4 TRIPLE-BASIS ESTIMATOR

If the tail-frequency behavior of output functions are controlled, then we may effectively estimate output functions using a finite number of projection coefficients; thus, we only need to estimate a finite number of the f_α functions. The 3BE consists of two orthonormal bases for estimating input and output functions respectively, and a random basis to estimate the mapping between them. To efficiently estimate the f_α functions, we shall use random basis functions from Random Kitchen Sinks (RKS) (Rahimi and Recht 2007). We shall show that to approximate f_α, we need only estimate a linear mapping in the random RKS features. Rahimi and Recht (2007) show that if one has a shift-invariant kernel K (in particular we consider the RBF kernel $K(x) = \exp(-x^2/2)$), then for fixed $\omega_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_d)$, $b_i \overset{iid}{\sim} \text{Unif}([0, 2\pi])$, we have that for each $x, y \in \mathbb{R}^d$:

$$K(\|x - y\|_2 / \sigma) \approx z(x)^T z(y), \text{ where} \tag{15}$$

$$z(x) \equiv \sqrt{\tfrac{2}{D}} \left[\cos(\omega_1^T x + b_1) \cdots \cos(\omega_D^T x + b_D)\right]^T, \tag{16}$$

and D is the number of random basis functions (see (Rahimi and Recht 2007) for approximation quality) . Let U and V be a set of indices for basis functions to project input and output functions respectively:

$$U = \{\alpha_1, \ldots, \alpha_s\}, V = \{\beta_1, \ldots, \beta_r\}. \tag{17}$$

In practice one would choose U and V through cross-validation (see §3.1.1). First note that:

$$\langle \tilde{p}_i, \tilde{p}_j \rangle = \left\langle \sum_{\alpha \in U} a_\alpha(P_i)\varphi_\alpha, \sum_{\alpha \in U} a_\alpha(P_j)\varphi_\alpha \right\rangle \tag{18}$$

$$= \sum_{\alpha \in U}\sum_{\beta \in U} a_\alpha(P_i) a_\beta(P_j) \langle \varphi_\alpha, \varphi_\beta \rangle \tag{19}$$

$$= \sum_{\alpha \in U} a_\alpha(P_i) a_\alpha(P_j) = \langle \vec{a}_U(P_i), \vec{a}_U(P_j) \rangle, \tag{20}$$

where $\vec{a}_U(P_i) = (a_{\alpha_1}(P_i), \ldots, a_{\alpha_s}(P_i))^T$. Thus, $\|\tilde{p}_i - \tilde{p}_j\|_2 = \|\vec{a}_U(P_i) - \vec{a}_U(P_j)\|_2$, where the norm on the LHS is the L_2 norm and the ℓ_2 on the RHS.

Consider a fixed σ, and let $\omega_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_s)$, $b_i \overset{iid}{\sim} \text{Unif}[0, 2\pi]$, be fixed. Then

$$f_\alpha(p_0) = \sum_{i=1}^{\infty} \theta_{\alpha i} K_\sigma(\|g_{\alpha i} - p_0\|_2) \tag{21}$$

$$\approx \sum_{i=1}^{\infty} \theta_{\alpha i} K_\sigma(\|\vec{a}_U(g_{\alpha i}) - \vec{a}_U(P_0)\|_2) \tag{22}$$

$$\approx \sum_{i=1}^{\infty} \theta_{\alpha i} z(\vec{a}_U(g_{\alpha i}))^T z(\vec{a}_U(P_0)) \tag{23}$$

$$= \psi_\alpha^T z(\vec{a}_U(P_0)), \tag{24}$$

where $\psi_\alpha = \sum_{i=1}^{\infty} \theta_{\alpha i} z(\vec{a}_U(g_{\alpha i})) \in \mathbb{R}^s$. Hence, by (24) f_α is approximately linear in $z(\vec{a}_U(\cdot))$; so, we consider linear estimators in the non-linear space induced by $z(\vec{a}_U(\cdot))$.

In particular, we take the OLS estimator using the data-set $\{(z(\vec{a}_U(P_i)), a_\alpha(Q_i))\}_{i=1}^N$, and for each f_α we estimate :

$$\hat{f}_\alpha(P_0) \equiv \hat{\psi}_\alpha^T z(\vec{a}_U(P_0)) \quad \text{where} \tag{25}$$

$$\hat{\psi}_\alpha \equiv \underset{\psi \in \mathbb{R}^D}{\arg\min} \|\vec{A}_\alpha - \mathbf{Z}\psi\|_2^2 = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\vec{A}_\alpha \tag{26}$$

for $\vec{A}_\alpha = (a_\alpha(Q_1), \ldots, a_\alpha(Q_N))^T$, and $\mathbf{Z}$ the $N \times D$ matrix $\mathbf{Z} = [z(\vec{a}_U(P_1)) \cdots z(\vec{a}_U(P_N))]^T$. Suppose that the indices of basis functions we project output function onto is V (as in (17)), then the set of functions we estimate is $\{\hat{f}_\beta \ : \ \beta \in V\}$. Let $\hat{f}_{1:r}(P_0) = (\hat{f}_{\beta_1}(P_0), \ldots, \hat{f}_{\beta_r}(P_0))^T$, $\mathbf{A}_{1:r} = [\vec{A}_{\beta_1}, \ldots, \vec{A}_{\beta_r}] \in \mathbb{R}^{N \times r}$:

$$\hat{f}_{1:r}(P_0) = \hat{\Psi}^T z(\vec{a}_U(P_0)) \quad \text{where} \tag{27}$$

$$\hat{\Psi} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{A}_{1:r}. \tag{28}$$

4.1 Evaluation Computational Complexity

We see that after computing $\hat{\Psi}$, evaluating the estimated projection coefficients for a new function p_0 amounts to performing a matrix multiplication of a $r \times D$ matrix with a $D \times 1$ vector. Including the time required for computing $z(\vec{a}_U(P_0))$, the computation required for the evaluation, (27), is: 1) the time for evaluating the projection coefficients $\vec{a}_U(P_0)$, $O(sn)$; 2) the time to compute the RKS features $z(\cdot)$, $O(Ds)$; 3) the time to compute the matrix multiplication, $\hat{\Psi}^T z(\vec{a}_U(P_0))$, $O(rD)$. Hence, the total time is $O(rD + Ds + sn)$.

We'll see that we may choose $D = O(n\log(n))$, $s = O(n)$, and $r = O(m)$. If we assume further that $m \asymp n$, the total runtime for evaluating $\hat{f}(\tilde{p}_0)$ is $O(n^2\log(n))$. Since we are considering data-sets where the number of instances N far outnumbers the number of points per sample set n, $O(n^2\log(n))$ is a *substantial improvement* over $\Omega(Nn)$ for the LSE; indeed, the LSE requires a metric evaluation with every training-set input function (2) where the 3BE does not. Furthermore, the space complexity is much improved for the 3BE since we only need to store the $O(n^2\log(n))$ matrix Ψ and the $O(n^2\log(n))$ total space for the RKS basis functions $\{(\omega_i, b_i)\}$. Contrast this with the space required for the LSE, $\Omega(Nn)$, which is much larger for our case of $n \ll N$. Lastly, note that to evaluate $\hat{q}_0(x) = [\hat{f}(P_0)](x)$ once one has computed $\hat{f}_{1:r}(P_0)$, one only needs to compute $\hat{q}_0(x) = \langle \hat{f}_{1:r}(P_0), \vec{\varphi}_{1:r}(x) \rangle$ where $\vec{\varphi}_{1:r}(x) = (\varphi_{\beta_1}(x), \ldots, \varphi_{\beta_r}(x))$.

Triple-Basis Estimator We note that a straightforward extension to the 3BE is to use a ridge regression estimate on features $z(\vec{a}_t(\cdot))$ rather than a OLS estimate. That is, for $\lambda \geq 0$ let

$$\hat{\psi}_{\alpha\lambda} \equiv \underset{\psi \in \mathbb{R}^D}{\arg\min} \|\vec{A}_\alpha - \mathbf{Z}\psi\|_2^2 + \lambda\|\psi\|_2^2 \tag{29}$$

$$= (\mathbf{Z}^T\mathbf{Z} + \lambda I)^{-1}\mathbf{Z}^T\vec{A}_\alpha. \tag{30}$$

The Ridge-3BE is still evaluated via a matrix multiplication, and our complexity analysis holds.

4.2 Algorithm

We summarize the basic steps for training the 3BE in practice given a data-set of empirical functional observations $\mathcal{D} = \{(P_i, Q_i)\}_{i=1}^N$, parameters σ and D (which may be cross-validated), and an orthonormal basis $\{\varphi_i\}_{i \in \mathbb{Z}}$ for $L_2([0,1])$.

1. Determine the sets of basis functions U and V (17) for approximating p, and q respectively. For each j in a subset $J \subseteq \{1, \ldots, N\}$* one can select a set M_{t_j} (9) to estimate p_j by cross-validating a loss as described in § 3.1.1. One may then set $U = M_{\bar{t}}$ where $\bar{t} = \frac{1}{|J|}\sum_{j \in J} t_j$. Similarly, one may set $V = M_{\bar{c}}$ by cross-validating M_{c_j}'s for q_j's.

2. Let $s = |U|$, draw $\omega_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2}I_s)$, $b_i \overset{iid}{\sim} \text{Unif}[0, 2\pi]$ for $i \in \{1, \ldots, D\}$; keep the set $\{(\omega_i, b_i)\}_{i=1}^D$ fixed henceforth.

3. Let $\{\beta_1, \ldots, \beta_r\} = V$. Generate the data-set of random kitchen sink features, output projection coefficient vector pairs: $\{(z(\vec{a}_U(P_i)), \vec{a}_V(Q_i))\}_{i=1}^N$. Let $\hat{\Psi} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{A}_{1:r} \in \mathbb{R}^{D \times r}$ where $\mathbf{Z} = [z(\vec{a}_U(P_1)) \cdots z(\vec{a}_U(P_N))]^T \in \mathbb{R}^{N \times D}$, $\mathbf{A}_{1:r} = [\vec{A}_{\beta_1}, \ldots, \vec{A}_{\beta_r}] \in \mathbb{R}^{N \times r}$ as in (28). Note that $\mathbf{Z}^T\mathbf{A}_{1:r}$ and $\mathbf{Z}^T\mathbf{Z}$ can be computed efficiently using parallelism.

4. For all future query input functional observations P_0, estimate the projection coefficients of the corresponding output function as $\hat{f}_{1:r}(P_0) = \hat{\Psi}^T z(\vec{a}_U(P_0))$.

5 THEORY

We analyze the L_2 risk for the 3BE estimator below. We assume that input/output functions belong to a Sobolev Ellipsoid function class and that the mapping between input and output functions is in $\mathcal{F}_\sigma$ (14).

5.1 Assumptions

5.1.1 Sobolev Ellipsoid Function Classes

We shall make a Sobolev ellipsoid assumption for classes $\mathcal{I}$ and $\mathcal{O}$. Let $a(h) \equiv \{a_\alpha(h)\}_{\alpha \in \mathbb{Z}^d}$. Suppose that the projection coefficients $a(p) = \{a_\alpha(p)\}_{\alpha \in \mathbb{Z}^l}$ and $a(q) = \{a_\alpha(q)\}_{\alpha \in \mathbb{Z}^k}$ are as follows for $p \in \mathcal{I}$, $q \in \mathcal{O}$:

$$\mathcal{I} = \{p : a(p) \in \Theta_l(\nu_\mathcal{I}, \gamma_\mathcal{I}, A_\mathcal{I}), \|p\|_\infty \leq A_\mathcal{I}\} \tag{31}$$

$$\mathcal{O} = \{q : a(q) \in \Theta_k(\nu_\mathcal{O}, \gamma_\mathcal{O}, A_\mathcal{O}), \|q\|_\infty \leq A_\mathcal{O}\} \tag{32}$$

*Empirically it has been observed that $\bar{t}$ and $\bar{c}$ perform well even when $|J|$ is much smaller than N

where $\nu_{\mathcal{I}}, \gamma_{\mathcal{I}} \in \mathbb{R}^l_{++}$, $\nu_{\mathcal{O}}, \gamma_{\mathcal{O}} \in \mathbb{R}^k_{++}$, $A_{\mathcal{I}}, A_{\mathcal{O}} \in \mathbb{R}_{++}$, $\mathbb{R}_{++} = (0, \infty)$, and

$$\Theta_d(\nu, \gamma, A) = \left\{ \{a_\alpha\}_{\alpha \in \mathbb{Z}^d} : \sum_{\alpha \in \mathbb{Z}^d} a_\alpha^2 \kappa_\alpha^2(\nu, \gamma) < A \right\} \tag{33}$$

$$\kappa_\alpha^2(\nu, \gamma) = \sum_{i=1}^{d} (\nu_i |\alpha_i|)^{2\gamma_i} \text{ for } \nu_i, \gamma_i, A > 0. \tag{34}$$

See Ingster and Stepanova (2011) and Laurent (1996) for other work using similar Sobolev elipsoid assumptions. The assumptions in (31) and (32) will control the tail-behavior of projection coefficients and allow one to effectively estimate $p \in \mathcal{I}$ and $q \in \mathcal{O}$ using a finite number of projection coefficients on the empirical functional observation.

Suppose that function h is such that $a(h) \in \Theta_d(\nu, \gamma, A)$ has a corresponding set of evaluations $H = \{y_j = h(u_j) + \epsilon_j\}_{j=1}^r$ where $u_j \overset{iid}{\sim} \text{Unif}([0,1]^d)$ and $\mathbb{E}[\epsilon_j] = 0, \mathbb{E}[\epsilon_j^2] < \infty$. Then, $\tilde{h}$, the estimate of h, is:

$$\tilde{h}(x) = \sum_{\alpha \,:\, \kappa_\alpha(\nu,\gamma) \le t} a_\alpha(H) \varphi_\alpha(x) \quad \text{where} \tag{35}$$

$$a_\alpha(H) = \frac{1}{r} \sum_{j=1}^{r} y_j \varphi_\alpha(u_j). \tag{36}$$

Choosing t optimally[†] can be shown to lead to $\mathbb{E}[\|\tilde{h} - h\|_2^2] = O(r^{-\frac{2}{2+\gamma^{-1}}})$, where $\gamma^{-1} = \sum_{j=1}^d \gamma_j^{-1}$, $r \to \infty$. Thus, we can represent h using a finite number of projection coefficients $\vec{a}_t(H) = (a_\alpha(H) : \kappa_\alpha(\nu, \gamma) \le t)^T$; this allows one to approximate the FFR problem as a regression problem over finite vectors $\vec{a}_t(P)$ and $\vec{a}_{t'}(Q)$. Note that our choice of sets M_t (9) in §3.1.1 corresponds to the estimator in (35) with $\nu, \gamma = \vec{1}$. Varying t in this case will still be adaptive to the smoothness of h, and the number of points in H.

5.1.2 Function to Funcion Mapping

Recall that we take output functions to be:

$$q(x) = [f(p)](x) = \sum_{\alpha \in \mathbb{Z}^k} f_\alpha(p) \varphi_\alpha(x)$$

where $f_\alpha(p) = a_\alpha(f(p))$. Our assumption of the class of mappings is:

$$\mathcal{F}_\sigma = \{f \,:\, \forall \alpha \in \mathbb{Z}^k \; \|\theta_\alpha\|_1 \le B_\alpha, f_\alpha \text{ as in (12)}\}$$

Suppose further that:

$$\sum_{\alpha \in \mathbb{Z}^k} B_\alpha^2 \kappa_\alpha^2(\nu_{\mathcal{O}}, \gamma_{\mathcal{O}}) \le A_{\mathcal{O}}. \tag{37}$$

Hence, if $f \in \mathcal{F}_\sigma$ then $q = f(p) \implies q \in \mathcal{O}$ since $|f_\alpha(p)| \le \|\theta_\alpha\|_1 \le B_\alpha$ and (37) holds.

[†]See appendix for details.

5.2 Risk Upperbound

Below we state our main theorem, upperbounding the risk of the 3BE (with truncation).

Theorem 5.1. *Let a small constant $\delta > 0$ be fixed. Suppose that $\hat{q}_0(x) = \sum_{\alpha \in M_u^{\mathcal{O}}} T_{B_\alpha}(\hat{f}_\alpha(P_0)) \varphi_\alpha(x)$, $T_B(x) \equiv \text{sign}(x) \min(|x|, B)$, and $\hat{f}_\alpha(P_0)$ given by* (25)*. Furthermore, suppose that* (31) *and* (32) *holds, and $f \in \mathcal{F}_\sigma$ is as in* (37)*. Moreover, assume that* (4) *holds and $n_i, m_i \asymp n$. Also, assume that the number of RKS features D* (16) *is taken to be $D \asymp n \log(n)$. Then,*

$$\mathbb{E}\left[\|q_0 - \hat{q}_0\|_2^2\right] \tag{38}$$

$$\le O\left(\left(n^{-1/(2+\gamma_{\mathcal{I}}^{-1})} + \frac{n \log(n) \log(N)}{N} \right)^{2/(2+\gamma_{\mathcal{O}}^{-1})} \right) \tag{39}$$

with probability at least $1 - \delta$.

See appendix for proof. The rate (39) yields consistency for our estimator if $n \log(n) = o(N/\log(N))$; that is, so long as one is in the large data-set domain where the number of instances is larger than the number of points in function observations. Note that the first summand in (39) is similar to typical functional estimation rates, and it stems from our approximation with bases; the second summand is akin to a linear regression rate, and it stems from our OLS estimation (26).

6 EXPERIMENTS

Below we show the improvement of the 3BE over previous FFR approaches in several real-world data-sets. Empirically, the 3BE proves to be the most general, quick, and effective estimator. Unlike previous time-series FFR approaches, the 3BE easily lends itself to working over distributions. Moreover, unlike previous nonparametric FFR estimators the 3BE does not need to compute pairwise kernel evaluations, making it much more scalable. All differences in MSE were statistically significant ($p < 0.05$) using paired t-tests.

6.1 Rectifying 2LPT Simulations

Numerical simulations have become an essential tool to study cosmological structure formation. Astrophysics use N-body simulations to study the gravitational evolution of collisionless particles like dark matter particles (Trac and Pen 2006). Unfortunately, N-body simulations require forces among particles to be recomputed over multiple time intervals, leading to a large magnitude of time steps to complete a single simulation. In order to mitigate the large computational costs of running N-body simulations, often simulations based on Second Order Lagrange Perturbation Theory (2LPT) are used (Scoccimarro 1998). Although

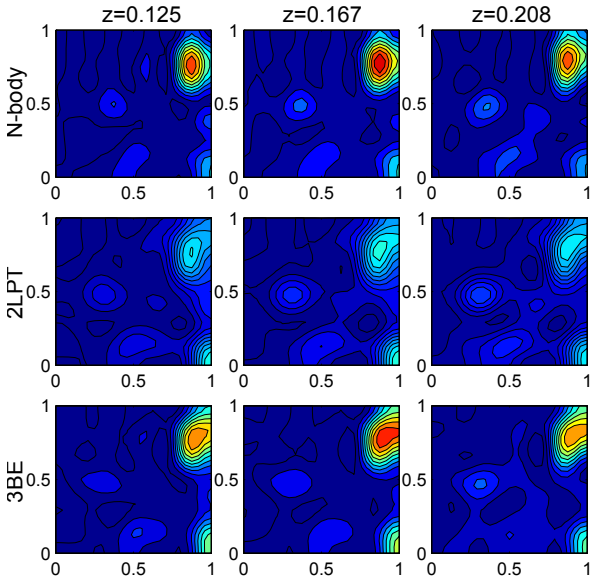


Figure 3: Slices of particle pdfs.

2LPT simulations are several orders of magnitude faster, they prove to be inaccurate, especially at smaller scales. In this experiment we bridge the gap between the speed of 2LPT simulations and the accuracy of N-body simulations using FFR and the 3BE. Namely, we regress the mapping between a distribution of particles in an area coming from a 2LPT simulation and the distribution of the particles in the same area under an equivalent N-body simulation.

We regress the distribution of 3d (spatial) N-body simulation particles in 16 Mpc^3 cubes when given the distribution of particles of the 2LPT simulation in the same cube (note that each distribution is estimated through the set of particles in each cube). A training-set of over 900K pairs of 2LPT cube sample-set/N-body cube sample-set instances was used, along with a test-set of 5K pairs. The number of projection coefficients used to represent input and output distributions was 365/401 respectively, chosen by cross-validating the density estimates. We chose the number of RKS features to be 15K based on rules-of-thumb. We cross-validated the σ and λ parameters of the ridge variant 3BE (30), and the smoothing parameter of the LSE and reported back the MSE and mean prediction time (MPT, in seconds) of our FFR estimates to the distributions truly coming directly through N-body simulation (Table 1); we also report the MSE of predicting the average output distribution (AD).

Method	MSE	MPT
3BE	**4.958**	**0.009**
LSE	6.816	4.977
2LPT	6.424	NA
AD	9.289	NA

Table 1: MSE and MPT(s) results.

We see that the 3BE is about $500\times$ faster than the LSE in terms of prediction time and achieved an improvement in R^2 of over 50% over using the distribution coming directly from the 2LPT simulation (2LPT, Table 1). Note also that the LSE does not achieve an improvement in MSE over 2LPT.

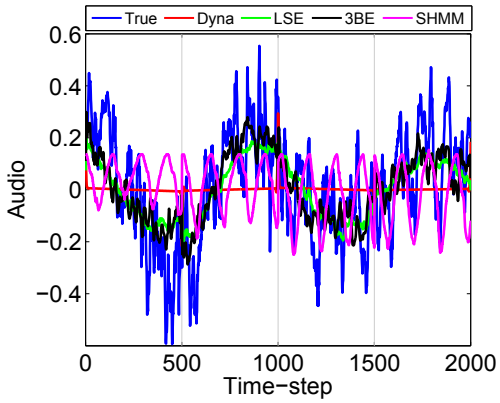


Figure 4: Example audio predictions; segments separated with vertical lines.

6.2 Time-series Data

We compared the performance of the 3BE in time-series prediction problems to using the LSE and widely used time-series prediction methods like Dynamics Mining with Missing values (DynaMMo) (Li et al. 2009) and Kernel Embedded HMMs (SHMM) (Song et al. 2010). DynaMMo is a latent-variable probabilistic model trained with EM aimed at predicting data that is missing in chunks and not just in a single time-step (as we also attempt with our functional responses).

6.2.1 Forward Prediction with Music Data

Music data presents a particularly interesting application of forward prediction for time-series. That is, given a short segment of audio data from a piece of music, can we predict the audio data in the short segment that follows? Uses for forward prediction with music include compression and music similarity.

In this experiment, we use a 30 second clip, sampled at 44.1 kHz from the song "I Turn To You" by the artist Melanie C. We extract a mono signal of the sound clip and use the first 85% for training and hold-out, and the final 15% for testing. To perform forward prediction in the test set, we take a 500 time-step segment of the (true) music time-series as input and use it to predict the following 500 time-steps. We repeat this sequentially over consecutive disjoint segments in the test set until we have made predictions for the entire test set. In total our data-set consisted of about 2200 training instances. For audio prediction with the 3BE we use the ridge variant (30). We use 150 trigonometric basis functions for both input and output functions, and 5000 RKS basis func-

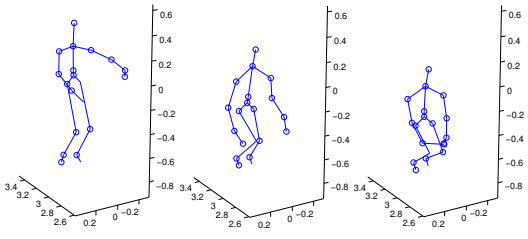


Figure 5: Example "duck" frames.

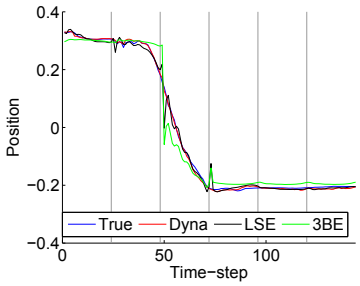


Figure 6: Occluded joint predictions.

tions (both quantities chosen via rules of thumb). We then cross-validate the bandwidth and λ penalty parameters.

We cross-validated the number of dimensions for hidden-states for DynaMMo, and the bandwidth parameter for the LSE. The mean squared error (MSE) on the test-set is reported in Table 2 for each method. The 3BE achieves the lowest estimation error. Furthermore, looking at Figure 4 it is apparent that the 3BE outperforms the other methods in terms of capturing the structure of the audio data. The quality of the audio predicted with the 3BE is also superior to the other methods (hear predicted sound clips in supplemental materials). Furthermore, DynaMMo takes over 4 hours to learn a model given a fixed hidden state dimensionality with no missing data (and even longer if also predicting missing data), where as the 3BE takes only about 2 minutes to cross-validate and perform predictions (a speed-up of over $7000\times$). Similarly the 3BE was over $5000\times$ faster than SHMM for predictions. Additionally, even though the data-set is of a smaller scale, the 3BE still enjoys a $3\times$ speedup over LSE for prediction time.

Method	MSE
3BE	**0.0327**
LSE	0.0351
Dyna	0.0492
SHMM	0.1082

Table 2: Audio MSE.

6.2.2 Co-occurring Predictions with Joint Motion Capture Data

Next, we explore predicting co-occurring time-series with motion capture (MoCap) data. We use the MSRC-12 Data-set (Fothergill et al. 2012). The 3d positions are provided for 20 total joints. We look to predict the time-series of the position of an unobserved joint over a T time-step segment given time-series data (one function for each joint's x, y, or z position) for R observed joints for the segment.

We performed co-occurring time-series prediction with MoCap data of a subject performing the gesture "duck" (Figure 5). We randomly chose 10 joints to designate as occluded, and used the other 10 as our non-occluded joints. We then solved 30 separate FFR problems, where each of the problems had one of the missing joints' time-series as the output response function (e.g. missing joint 1's y position or missing joint 4's x position). In each of the problems, the 30 functions corresponding to the time-series for non-occluded joint spatial positions were used as inputs (by concatenating the projection coefficients of each input function) . We considered segments of 24 time-steps for time-series functions. In total we used a training set of about 1100 instances. The number of projection coefficients for functions was taken to be 10 while the number of RKS features was 250. The same parameters for all estimators were cross validated as before.

DynaMMo performs the best (Table 3), which is perhaps not surprising given that MoCap occlusion prediction was a point of emphasis for DynaMMo. However, the differences in prediction qualities among the different methods is not as pronounced in this data-set (Figure 6). We again see a speed up of over 1000x using 3BE over DynaMMo, also there was a speed up of over $30\times$ in prediction time over LSE.

Method	MSE
3BE	7.78E-4
LSE	1.3E-3
Dyna	**2.40E-4**

Table 3: MoCap MSE.

7 CONCLUSION

In conclusion, this paper presents a new estimator, the Triple Basis Estimator (3BE), for performing function to function regression in a scalable manner. Since functional data is complex, it is important to have an estimator that is capable of using massive data-sets in order to achieve a low estimation risk. To the best of our knowledge, the 3BE is the first nonparametric FFR estimator that is capable to scaling to big data-sets. The 3BE achieves this through the use of a basis representation of input and output functions and random kitchen sink basis functions. We analyzed the risk of the 3BE given non-parametric assumptions. Furthermore, we showed an improvement of several orders of magnitude for prediction speed and a reduction in error over previous estimators in various real-world data-sets.

Acknowledgements

This work was supported in part by NSF grant IIS1247658.

[1] F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer, 2006.

[2] Simon Fothergill et al. "Instructing people for training gestural interactive systems". In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM. 2012, pp. 1737–1746.

[3] Y. Ingster and N. Stepanova. "Estimation and detection of functions from anisotropic Sobolev classes". In: *Electronic Journal of Statistics* 5 (2011), pp. 484–506.

[4] Hachem Kadri et al. "Nonlinear functional regression: a functional RKHS approach". In: *JMLR Workshop and Conference Proceedings*. Vol. 9. 2010, pp. 374–380.

[5] B. Laurent. "Efficient estimation of integral functionals of a density". In: *The Annals of Statistics* 24.2 (1996), pp. 659–681.

[6] Lei Li et al. "Dynammo: Mining and summarization of coevolving sequences with missing values". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 507–516.

[7] Junier B Oliva, Barnabás Póczos, and Jeff Schneider. "Distribution to Distribution Regression". In: *ICML* (2013).

[8] Junier B Oliva et al. "Fast Distribution To Real Regression". In: *AISTATS* (2014).

[9] Junier B Oliva et al. "FuSSO: Functional Shrinkage and Selection Operator". In: *AISTATS* (2014).

[10] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2007, pp. 1177–1184.

[11] James O Ramsay and B.W. Silverman. *Functional data analysis*. Wiley Online Library, 2006.

[12] J.O. Ramsay and B.W. Silverman. *Applied functional data analysis: methods and case studies*. Vol. 77. Springer New York: 2002.

[13] Roman Scoccimarro. "Transients from initial conditions: a perturbative analysis". In: *Monthly Notices of the Royal Astronomical Society* 299.4 (1998), pp. 1097–1118.

[14] Le Song et al. "Hilbert space embeddings of hidden Markov models". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 991–998.

[15] Hy Trac and Ue-Li Pen. "Out-of-core hydrodynamic simulations for cosmological applications". In: *New Astronomy* 11.4 (2006), pp. 273–286.