Multiscale Dictionary Learning for Estimating Conditional Distributions

Anonymous Author(s)
Affiliation
Address
email

Abstract

Nonparametric estimation of the conditional distribution of a response given high-dimensional features is a challenging problem. In many settings it is important to allow not only the mean but also the variance and shape of the response density to change flexibly with features, which are massive-dimensional with a distribution concentrated near a lower-dimensional subspace or manifold. We propose a multiscale model based on a novel stick-breaking prior placed on the dictionary weights. The algorithm scales efficiently to massive numbers of features, and can be implemented efficiently with slice sampling. State of the art predictive performance is demonstrated for toy examples and a real data application.

1 Introduction

Massive datasets are becoming a ubiquitous by-product of modern scientific and industrial applications. These data present statistical and computational challenges for machine learning because many previously developed approaches do not scale-up sufficiently. Specifically, challenges arise because of the ultrahigh-dimensionality, and relatively low sample size (the "large p, small n" problem). Parsimonious models for such big data assume that the density in the ambient dimension concentrates around a lower-dimensional (possibly nonlinear) subspace. Indeed, a plethora of methodologies are emerging to estimate such lower-dimensional "manifolds" from high-dimensional data [1, 2].

We are interested in using such lower-dimensional embeddings to obtain estimates of the conditional distribution of some target variable(s). This *conditional regression* setting arises in a number of important application areas, including neuroscience, genetics, and video processing. For example, one might desire automated estimation of a predictive density for a continuous neurologic *phenotype* of interest, such as intelligence or a creativity score, on the basis of available data for a patient including neuroimaging. The challenge is to estimate the probability density function of the phenotype *nonparametrically* based on an $\mathcal{O}(10^6)$ dimensional image of the subject's brain. It is crucial to avoid parametric assumptions on the density, such as Gaussianity, while allowing the density to change flexibly with predictors. Otherwise, one can obtain misleading predictions and poorly characterize predictive uncertainty.

There is a rich machine learning and statistical literature on conditional density estimation of a response $y \in \mathcal{Y}$ given a set of features (predictors) $x = (x_1, x_2, \ldots, x_p) \in \mathcal{X}$. Common approaches include hierarchical mixtures of experts [3, 4], kernel methods [5, 6, 7, 8], Bayesian finite mixture models [9, 10, 11] and Bayesian nonparametrics [12, 13, 14, 15, 16].

However, there has been limited consideration of scaling to large p settings, with the variational Bayes approach of [10] being a notable exception. For dimensionality reduction, Tran et al. follow a greedy variable selection algorithm. Their approach does not scale to the sized applications we are interested in. For example, in a problem with $p = 1,000$ and $n = 500$, they reported a CPU time

of 51.7 minutes for a single analysis. We are interested in problems with p having many more orders of magnitude, requiring a faster computing time while also accommodating flexible non-linear dimensionality reduction (variable selection is a limited sort of dimension reduction). To our knowledge, there are no nonparametric density regression competitors to our approach, which maintain a characterization of uncertainty in estimating the conditional densities; rather, all sufficiently scalable algorithms provide point predictions and/or rely on restrictive assumptions such as linearity.

In big data problems, scaling is often accomplished using divide-and-conquer techniques. Well known examples are classification and regression trees (CART) [17] and multivariate adaptive regression splines (MARS) [18]. These algorithms fit surfaces to data by explicitly dividing the input space into a nested sequence of regions, and by fitting simple surfaces within these regions. Though these methods are appealing in providing a simple, flexible and interpretable mechanism of dimension reduction, it is well known that single tree estimates commonly have high variance and poor performance. There is a rich literature proposing improvements based on bagging [19], boosting [20] and random forests [21]. Though these algorithms can substantially improve mean square error performance, computation can be expensive and performance degrades as dimensionality p increases.

In fact, a significant downside of many divide-and-conquer algorithms is their poor scalability to high dimensional predictors. As the number of features increases, the problem of finding the best splitting attribute becomes intractable so that CART, MARS and multiple trees models cannot be efficiently applied. Also mixture of experts models become computationally demanding, since both mixture weights and dictionary densities are predictor dependent. In an attempt to make mixtures of experts more efficient, sparse extensions relying on different variable selection algorithms have been proposed [22]. However, performing variable selection in high dimensions is effectively intractable: algorithms need to efficiently search for the best subsets of predictors to include in weight and mean functions within a mixture model, an NP-hard problem.

In order to efficiently deal with massive datasets, we propose a novel multiscale approach which starts by learning a multiscale dictionary of densities,. This tree is efficiently learned in a first stage using a fast and scalable graph partitioning algorithm applied to the high-dimensional observations [23]. Expressing the conditional densities $f(y|x)$ for each $x \in \mathcal{X}$ as a convex combination of coarse-to-fine scale dictionary densities, the learning problem in the second stage estimates the corresponding multiscale probability tree. This is accomplished in a Bayesian manner using a novel multiscale stick-breaking process, which allows the data to inform about the optimal bias-variance tradeoff; weighting coarse scale dictionary densities more highly decreases variance while adding to bias if the finer scale structure is needed. This results in a model that allows borrowing information across different resolution levels and reaches a good compromise in terms of the bias-variance tradeoff. We show that the algorithm scales efficiently to massive numbers of features.

2 Setting

Let $X : \Omega \to \mathcal{X} \subseteq \mathbb{R}^p$ be a p-dimensional Euclidean vector-valued predictor random variable, taking values $x \in \mathcal{X}$, with a marginal probability distribution F_X. Similarly, let $Y : \Omega \to \mathcal{Y}$ be a $\mathcal{Y}$-valued target random variable, taking values $y \in \mathcal{Y}$, with a marginal probability distribution F_Y (we will specify specific forms of $\mathcal{Y}$, e.g., $\mathcal{Y} \subseteq \mathbb{R}^q$, below). We assume that the pair (X, Y) is sampled from a joint distribution, $F_{X,Y} \in \mathcal{F}$.

For inferential expedience, we posit the existence of a latent random variable $\boldsymbol{\eta} : \Omega \to \mathcal{M} \subseteq \mathcal{X}$, where $\mathcal{M}$ is only d "dimensional" and $d \ll p$. Note that $\mathcal{M}$ need not be a linear subspace of $\mathcal{X}$, rather, $\mathcal{M}$ could be, for example, a union or affine subspaces, or a smooth compact Riemannian manifold. Regardless of the nature of $\mathcal{M}$, we assume that we can approximately decompose the joint distribution as follows, $F_{X,Y,\boldsymbol{\eta}} = F_{X,Y|\boldsymbol{\eta}} F_{\boldsymbol{\eta}} = F_{Y|X,\boldsymbol{\eta}} F_{X|\boldsymbol{\eta}} F_{\boldsymbol{\eta}} \approx F_{X|\boldsymbol{\eta}} F_{Y|\boldsymbol{\eta}} F_{\boldsymbol{\eta}}$. In words, we assume that the *signal* approximately concentrates around a low-dimensional latent space, $F_{Y|X,\boldsymbol{\eta}} = F_{Y|\boldsymbol{\eta}}$. Note that this is a much less restrictive assumption than the commonplace assumption in manifold learning that the marginal distribution F_X concentrates around a low-dimensional latent space.

To provide some intuition around this model, we provide the following concrete example where the distribution of $y \in \mathbb{R}$ is a Gaussian function of the coordinate $\eta \in \mathcal{M}$ along the swissroll, which is

embedded in a high-dimensional ambient space:

$$Y|\eta \sim \mathcal{N}(\mu(\eta), \sigma(\eta)) \tag{1a}$$

$$X_r \sim \mathcal{N}(0,1) \text{ for } r \in \{3,\dots,p\}, \qquad X_1 = \eta\sin(\eta), \qquad X_2 = \eta\cos(\eta) \tag{1b}$$

$$\eta \sim U(0,1), \tag{1c}$$

where $\mathcal{N}(\mu,\sigma)$ denotes a Gaussian distribution with mean μ and standard deviation σ, and $U(0,1)$ denotes the uniform distribution on $(0,1)$. Clearly, Y is conditionally dependent on η, which is the low-dimensional signal manifold, of which X is also a function. In particular, X lives on a swissroll embedded in a p-dimensional ambient space, but Y is only a function of the coordinate η along the swissroll $\mathcal{M}$. The left panels of Figure 1 depict this concrete example when $\mu(\eta) = \eta$ and $\sigma(\eta) = \eta + 1$.

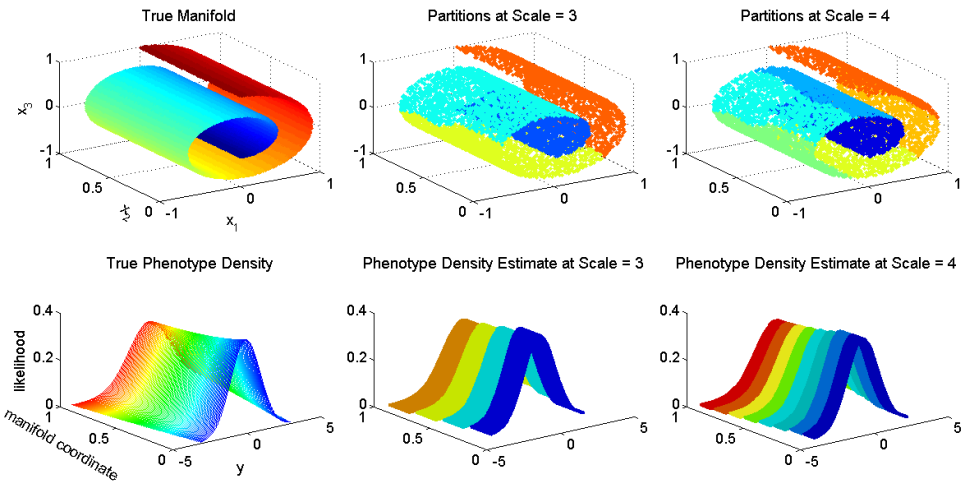

Figure 1: Illustration of our generative model and algorithm on a swissroll. The top left panel shows the manifold $\mathcal{M}$ (a swissroll) embedded in a p-dimensional ambient space, where the color indicates the coordinate along the manifold, η (only the first 3 dimensions are shown for visualization purposes). The bottom left panel shows the distribution of Y as a function of η, in particular, $F_{Y|\eta} = \mathcal{N}(\eta, \eta+1)$. The middle and right panels show our estimates of $F_{Y|\eta}$ at scales 3 and 4, respectively, which follow from partitioning our data. Sample size was $n = 10\,000$.

3 Goal

Our goal is to develop an approach to learn about $F_{Y|X}$ from n pairs of observations that we assume are sampled exchangeable from the joint distribution, $(x_i, y_i) \sim F_{X,Y} \in \mathcal{F}$. Let $\mathcal{D}^n = \{(x_i, y_i)\}_{i\in[n]}$, where $[n] = \{1,\dots,n\}$. More specifically, we seek to obtain a posterior over $F_{Y|X}$. We insist that our approach satisfies several desiderata, including most importantly: (i) scales up to $p \approx 10^6$ in reasonable time, (ii) yields good empirical results, and (iii) automatically adapts to the complexity of the data corpus. To our knowledge, no extant approach for estimating conditional densities or posteriors thereof satisfies even our first criterion.

4 Methodology

4.1 Ms. Deeds Framework

We propose here a general modular approach which we refer to as multiscale dictionary learning for estimating conditional distributions ("Ms. Deeds"). Ms. Deeds consists of three components: (i) a tree decomposition of the space, (ii) an embedding of the data into a lower-dimensional space, and (iii) an assumed form of the conditional probability model.

Tree Decomposition A tree decomposition τ yields a multiscale partition of the data or the ambient space in which the data live. Let $(\mathcal{W}, \rho_W, F_W)$ be a measurable metric space, where F_W is a Borel probability measure, $\mathcal{W}$, and $\rho_W : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ is a metric on $\mathcal{W}$. Let $B_r^{\mathcal{W}}(w)$ be the ρ_W-ball inside $\mathcal{W}$ of radius $r > 0$ centered at $w \in \mathcal{W}$. For example, $\mathcal{W}$ could be the data corpus $\mathcal{D}_n$, or it could be $\mathcal{X} \times \mathcal{Y}$. We define a tree decomposition as in [**?**, **?**]. A partition tree τ of $\mathcal{W}$ consists of a collection of cells, $\tau = \{C_{j,k}\}_{j \in \mathbb{Z}, k \in \mathcal{K}_j}$. At each scale j, the set of cells $C_j = \{C_{j,k}\}_{k \in \mathcal{K}_j}$ provides a disjoint partition of $\mathcal{W}$ almost everywhere, and $\mathcal{K}_j$ is the set of partitions at scale j. We define $j = 0$ as the root node/cell. For each $j > 0$, each $C_{j,k}$ has a unique parent node $C_{j-1,k'}$ containing $C_{j,k}$, and conversely, any $C_{j,k} \subseteq C_{j-1,k'}$ is called a child of $C_{j-1,k'}$.

Let $A_{j,k} = \{k' \in \mathcal{K}_{j'} : j' < j \text{ s.t. } C_{j,k} \subseteq C_{j',k'}\}$ denote the ancestors of $C_{j,k}$, and let $D_{j,k} = \{k' \in \mathcal{K}_{j'} : j' > j \text{ s.t. } C_{j',k'} \subseteq C_{j,k}\}$ denote the descendants of $C_{j,k}$. The result of this partitioning is that we can approximate $F_{Y|X}$ at each scale j via a mixture over densities, $F_{Y|X} \approx \hat{F}_{Y|X} = \sum_{j \in \mathbb{Z}, k \in \mathcal{K}_j} \hat{F}^{j,k}_{Y|X}$.

Embeddings At each scale, for each cell, we consider some embedding $\psi_{j,k} \colon C_{j,k} \to \Xi$, where $(\xi_x, \xi_y) \in \Xi$. Thus, rather than approximating $F_{Y|X}$ by a mixture of densities conditional on high-dimensional predictors, we can approximate $F_{Y|X}$ by a mixture of densities conditional on low-dimensional embeddings, $F_{Y|X} \approx \hat{F}_{Y|X} = \sum_{j \in \mathbb{Z}, k \in \mathcal{K}_j} \hat{F}^{j,k}_{\boldsymbol{\xi}_y|\boldsymbol{\xi}_x}$.

Family Each $\hat{F}^{j,k}_{\boldsymbol{\xi}_y|\boldsymbol{\xi}_x}$ is an element of a family of distributions, $\mathcal{F}^{j,k}$. This family might be quite general, e.g., all possible conditional densities, or quite simple, e.g., Gaussian distributions. Moreover, the family can adapt with j or k, being more complex at the coarser scales (for which $n_{j,k}$'s are larger), and simpler for the finer scales (or partitions with fewer samples).

Thus, collectively, any multiscale conditional density estimation procedure makes choices for the above three components. This encompasses a wide range of possible approaches.

4.2 Specific Choices

Our specific choices for each of the three components were guided by: qualitative desiderata, computational considerations, and feasibility of theoretical analysis. In terms of qualitative desiderata, we desire a strategy that estimates posteriors over all potential marginal distributions so as to automatically obtain estimates of uncertainty. Moreover, we would like a procedure with a few "tuning knobs" (hyper-parameters) as possible. These desiderata motivate using a fully Bayesian strategy. However, a fully Bayesian approach is computational intractable for the ultrahigh-dimensionality problems that motivate this work ($p \in \mathcal{O}(10^6)$). Thus, we adopt a partially Bayesian strategy.

Tree Partition Unlike classical harmonic theory which presupposes τ (e.g., in wavelets [**?**]), we choose to learn τ from the data. Previously, Chen et al. [**?**] developed a multiscale measure estimation strategy, and proved that there exists a scale j such that the approximate measure is within some bound of the true measure, under certain relatively general assumptions. We could, therefore, adopt that strategy for both $\hat{F}_{X,Y}$ and $\hat{F}_X$, and then divide to obtain $\hat{F}_{Y|X}$. Instead, we decided to simply partition the X's, ignoring the Y's in the partitioning strategy.

Our justification for this choice is as follows. First, sometimes there are many different $\mathcal{Y}$'s for many different applications. In such cases, we do not want to bias the partitioning to any specific $\mathcal{Y}$'s, all the more so when new unknown $\mathcal{Y}$'s may later emerge. Second, because the X's are so much higher dimensional than the Y's in our applications of interest, the partitions would be dominated by the X's, unless we chose a partitioning strategy that emphasized the Y's. Thus, our strategy mitigates this difficulty (while certainly introducing others).

Given that we are going to partition using only the X's, we still face the choice of precisely how to partition. A fully Bayesian approach would construct a large number of partitions, and integrate over them to obtain our posteriors. However, such a fully Bayesian strategy remains computationally intractable at scale, so we adopt a hybrid strategy. Specifically, we employ METIS [**?**], a well-known relatively efficient multiscale partitioning algorithm with demonstrably good empirical performance on a wide range of graphs. Graph construction follows via computing all pairwise distances using

$\rho_{uv} = \rho_W(x_u, x_v) = \|\widetilde{x}_u - \widetilde{x}_v\|_2$, where $\widetilde{x}$ is the whitened x (i.e., mean subtracted and variance normalized). We let there be an edge between x_u and x_v whenever $\mathrm{e}^{-\rho_{uv}^2} > t$, where t is some threshold chosen to elicit the desired sparsity level.

Applying METIS on the graph constructed in this way yields a single tree. We then place a non-parametric prior π over the leaves of the tree, to facilitate borrowing strength across the paths. More specifically, we let π be generated by a stick-breaking process [24]. For each node $C_{j,k}$ in the partition tree, we define a stick length $V_{j,k} \sim \text{Beta}(1, \alpha)$. The parameter α encodes the complexity of the model, with $\alpha = 0$ corresponding to the case in which $f(y|x) = f(y)$. The stick-breaking process is defined as follows:

$$\pi_{j,k}(x) \propto V_{j,k} \prod_{C_{j',k'} \in A_{j,k}} [1 - V_{j',k'})],$$

where $\sum_{k=1}^{|\mathcal{K}_j|} \pi_{j,k} = 1$. We refer to this prior as a *multiscale stick-breaking process*. Note that this Bayesian nonparametric prior assigns a positive probability to all possible paths, including those not observed in the training data. Thus, by adopting this Bayesian formulation, we are able to obtain posterior estimates for any newly observed data, regardless of the amount and variability of training data. This is a pragmatically useful feature of the Bayesian formulation, in addition to the alleviation of the need to choose a scale [**?**].

Embedding We let each $\psi_{j,k}$ simply be a Dirac delta function operating *only on the X's*. This is because, in our application of interest, X's are quite high-dimensional, and the Y's are relatively low-dimensional (e.g., one-dimensional). The choice of Dirac delta functions over, say, hyperplanes, alleviates the computational and theoretical difficulties of estimating hyperplanes via SVD and choosing the dimensions thereof. That said, theoretical considerations imply that the relative computation cost to computing SVDs for each partition is $\mathcal{O}(d^2)$, whereas partitioning is $\mathcal{O}(3^d)$, where d is the intrinsic dimension [**?**]. In practice, except for very low-dimensional intrinsic dimensional data, building the partition dominates the computational burden [**?**]. Moreover, empirical results seem to be robust to choice of embedding dimension [**?**]. Nonetheless, results from multiscale measure estimation [**?**] suggest that choosing a Dirac delta function is sufficient to guarantee accurate estimates of the empirical marginal measure, F_X, for some scale j. Thus, we view our chosen strategy as the simplest approach, making code, computations, and theory all more tractable.

Family We let the family of conditional densities for Y be Gaussian for simplicity, that is, we assume that $\mathcal{F}^{j,k} = \{\mathcal{N}(\mu_{j,k}, \sigma_{j,k}) : \mu \in \mathbb{R},\, \sigma \geq 0\}$. Because we are interested in posteriors over the conditional distribution $F_{Y|X}$, we place relatively uninformative but conjugate priors on $\mu_{j,k}$ and $\sigma_{j,k}$, specifically, assuming the y's have been whitened and are unidimensional, $\mu_{j,k} \sim \mathcal{N}(0, 1)$ and $\sigma_{j,k} = \mathcal{IG}(a, b)$. Obviously, other choices, such as finite or infinite mixtures of Gaussians are also possible for continuous valued data.

4.3 Estimation

We introduce the latent variable $\ell_i \in \{1, \ldots, k\}$, for $i = 1, \ldots, n$, denoting the multiscale level used by the i^{th} observation. Let $n_{j,k}$ be the number of observations in $C_{j,k}$. Each Gibbs sampler iteration can be summarized in the following steps: *francy: i changed the notation a bit to be consistent with previous literature. can you check to make sure i didn't screw anything up?*

(i) Update ℓ_i by sampling from the multinomial full conditional with

$$\Pr(\ell_i = j \,|\, \cdot) = \frac{\pi_{j,k}(x_i) f_{j,k}(y_i|x_i)}{\sum_{k'=1}^{k} \pi_{j,k'}(x_i) f_{j,k'}(y_i|x_i)}$$

(ii) Update stick-breaking random variable $V_{j,k}(x_i)$, for $j = 1, \ldots, |\mathcal{K}_j|$ and $i = 1, \ldots, n$, from $\text{Beta}(\beta', \alpha')$ with $\beta' = 1 + n_{j,k}$ and $\alpha' = \alpha + \sum_{C_{j,K} \in D_{j,k}(x_i)} n_{j,k}(x_i)$.

(iii) Update $\mu_{j,k}(x_i)$ and $\sigma_{j,k}(x_i)$ by sampling from

$$\mu_{j,k} \sim \mathcal{N}\left(n_{j,k} \frac{\bar{y}_{j,k}}{\sigma_{j,k}}, (1 + \frac{n_{j,k}}{\sigma_{j,k}})^{-1}\right), \quad \sigma_{j,k} \sim \mathcal{IG}\left(a_\sigma, b + 0.5 \sum_{i \in \mathcal{I}_{j,k}} (y_i - \mu_{j,k})^2\right)$$

with $a_\sigma = a + n_{j,k}/2$, $\bar{y}_{j,k}$ being the average of the observation $\{y_i\}$ allocated to cell $C_{j,k}$ and $\mathcal{I}_{j,k} = \{i : \ell_i = j, x_i \in C_{j,k}\}$.

4.4 Predictions

Consider the case we want to predict the response y_{n+1} for a future observation based on the predictors x_{n+1}. Because the partitioning strategy that we adopted lacks an elegant out-of-sample embedding function (unlike other paritioning strategies), we adopt a Voronoi expansion procedure by which the new predictors x_{n+1} are allocated to $C_{j,k}$'s having the closest centers with respect to ρ_W. For a new observation the predictive density is defined as *francy: it seems like the training x_i's must be in here somewhere. are they implictly in Ω? also, where is $f\left(y_{n+1}|x_{n+1}, \Omega\right)$ defined? where is* **??***?*

$$p(y_{n+1}|x_{n+1}, y_1, \ldots, y_n) = \int f\left(y_{n+1}|x_{n+1}, \Omega\right) dp\left(\Omega|y_1, \ldots, y_n\right)$$

with $f\left(y_{n+1}|x_{n+1}, \Omega\right)$ defined as in (1) and Ω being the set of all parameters involved, i.e. weights, location and scale parameters. In order to make inference on the predictive density of y_{n+1}, at the sth Gibbs sampler iteration, we will first sample parameters involved in **??** from its posterior, i.e. $\Omega^{(s)} \sim p\left(\Omega|y_1, \ldots, y_n\right)$ and then we will sample $y_{n+1}^{(s)}$ from $p\left(y_{n+1}|x_{n+1}, \Omega^{(s)}\right)$. Let us assume the number of iterations is S an a burn-in of b is considered. Then, given the sequence $\left(y_{n+1}^{(b+1)}, \ldots, y_{n+1}^{(S)}\right)$, summaries of the predictive density such as mean, variance and quantiles can be computed.

To make predictions, the Gibbs sampler was run with up to $20,000$ iterations, including a burn-in of $1,000$. Gibbs sampler chains were stopped testing normality of normalized averages of functions of the Markov chain [25]. Parameters (a, b) and α involved in the prior density of parameters $\sigma_{j,k}$'s and $V_{j,k}$'s were set respectively equal to $(3, 1)$ and 1. All predictions used a leave-one-out strategy.

4.5 Simulation Studies

In order to assess the predictive performance of the proposed model, we considered the four different simulation scenarios described below:

(1) Nonlinear Mixture We first consider a relatively simple yet nonlinear joint model:

$$Y|\eta \sim |\eta|\mathcal{N}(\mu_1, \sigma_1) + (1 - |\eta|)\mathcal{N}(\mu_2, \sigma_2), \tag{2a}$$
$$X_r|\eta \sim \mathcal{N}(\eta, \sigma_x), \qquad r \in \{1, 2, \ldots, p\}, \tag{2b}$$
$$\eta \sim \sin(U(0, c)). \tag{2c}$$

In the simulations we let $(\mu_1, \sigma_1) = (-2, 1)$, $(\mu_2, \sigma_2) = (2, 1)$, $\sigma_x = 0.01$, and $c = 20$, and $p = 1000$. Thus, $F_{Y|X}$ is a highly nonlinear function of X, and even η, and X is high-dimensional.

(2) Swissroll We then return to the swissroll example of Figure 1; here we let $(\mu, \sigma) = (\eta, 1)$ and $p \in 1000 \times \{50, 100\}$.

(3) Linear Subspace Letting $\Gamma \in \mathbb{R}^{p+1 \times q}$ and Θ be an $q \times d$ "diagonal" matrix (meaning all entires other than the first $d < q$ elements of the diagonal are zero), we assume the following model: $Y, X|\eta \sim \mathcal{N}_{p+1}(\Gamma\Theta\eta, \Sigma_0)$, where $\Gamma \sim \mathcal{S}_{p+1,d}$ indicates Γ is uniformly sampled from the set of all orthonormal d frames in $\mathbb{R}^{p+1}$ (a Stiefel manifold), $\theta_{ii} \sim \mathcal{IG}(a_\theta, b_\theta)$ for $i \in \{1, \ldots, d\}$ and all other elements of Θ are zero, and $\eta \sim \mathcal{N}_d(0, I)$. In the simulation, we let $q = d = 5$, $(\alpha_\theta, \beta_\theta) = (??, ??)$, and $p \in 1000 \times \{50, 70, 90, 100, 200, 300, 500\}$.

(4) Union of Linear Subspaces This model is a direct extension of the linear subspace model, as it is a union of subspaces. We let the dimensionality of each subspace vary to demonstrate the generality of our procedure. Specifically, we assume $Y, X|\eta \sim \sum_{g=1}^{G} \omega_g \mathcal{N}_{p+1}(\Gamma_g\Theta_g\eta, \Sigma_0)$, $\omega \sim Dirichlet(\boldsymbol{\alpha})$ $\eta \sim \mathcal{N}_d(0, I)$, where $\Gamma \sim \mathcal{S}_{p+1,g}$ and Θ_g is a "diagonal" with $\theta_{ii} \sim \mathcal{IG}(a_g, b_g)$ for $i \in \{1, \ldots, g\}$, and the remaining elements of Θ are zero. In the simulation, we let $G = 5$, $\boldsymbol{\alpha} = (1, \ldots, 1)^\intercal$, $(\alpha_g, \beta_g) = (\alpha_\theta, \beta_\theta)$ as above, for $p \in 1000 \times \{50, 100\}$.

4.6 Neuroscience Applications

We assessed the predictive performance of the proposed method on two very different neuroimaging datasets. First, we consider a structural connectome dataset collected at the Mind Research Network. Data were collected as described in Jung et al. [26]. For the analysis, all variables were normalized by subtracting the mean and dividing by the standard deviation. The prior specification and Gibbs sampler described in §3 were utilized.

In the first experiment we investigated the extent to which we could predict creativity (as measured via the Composite Creativity Index [27]). For each subject, we estimate a 70 vertex undirected weighted brain-graph using the Magnetic Resonance Connectome Automated Pipeline [28] from diffusion tensor imaging data [29]. Because our graphs are undirected and lack self-loops, we have a total of $p = \binom{70}{2} = 2,415$ potential weighted edges. The vector of covariates consists in the logarithm of the total number of connections between all pairs of cortical regions.

The second dataset comes from a resting-state functional magnetic resonance experiment as part of the Autism Brain Imaging Data Exchange [30]. We selected the Yale Child Study Center for analysis. Each brain-image was processed using the Configurable Pipepline for Analysis of Connectomes [31]. For each subject we computed a measure of normalized power at each voxel called fALFF [32]. To ensure the existence of nonlinear signal relating these predictors, we let y_i correspond to an estimate of overall head motion in the scanner, called mean framewise displacement (FD) computed as described in Power et al. [33]. In total, there were $p =??$ voxels.

4.7 Evaluation Criteria

To compare algorithmic performance we considered $r_m^{\mathcal{A}}$ defined as $r_m^{\mathcal{A}} = \phi(MSB)/\phi(\mathcal{A})$, where ϕ is the quantity of interest (for example, CPU time in seconds or mean squared error), MSB is our approach and $\mathcal{A}$ is the competitor algorithm. To obtain mean-squared error estimates from MSB, we select our posterior mode as a point-estimate (the comparison algorithms do not generate posterior predictions, only point estimates). For each simulation scenario, we sampled multiple datasets and compute the *matched* distribution of $r_m^{\mathcal{A}}$. In other words, rather than running simulations and reporting the distribution of performance for each algorithm, we compare the algorithms per simulation. This provides a much more informative indication of algorithmic performance, in that we indicate the fraction of simulations one algorithm outperforms another on some metric. This is akin to power gained by matched two-sample tests. For each example, we sampled 20 datasets to obtain estimates of the distribution over $r_m^{\mathcal{A}}$. All experiments were performed on a typical workstation, Intel Core i7-2600K Quad-Core Processor with 8192 MB of RAM.

5 Results

5.1 Illustrative Example

The middle and right panels of Figure 1 depict the quality of partitioning and density estimation for the swissroll example described in §2, with the ambient dimension $p = 1000$ and the manifold dimension $d = 1$. We sampled $n = 10\,000$ samples for this illustration. At scale 3, we have $4 = 2^{3-1}$ partitions and at scale 4, we have $8 = 2^{4-1}$ (note that the partition tree, in general, need not be binary). The top panels are color coded to indicate which x_i's fall into which partition. Although imperfect, it should be clear that the data are partitioned very well. The bottom panels show the resulting estimate of the posteriors at the two scales. These posteriors are "piecewise linear" in a certain sense, as they are invariant to the manifold coordinate within a given partition.

To obviate the need to choose a scale to use to make a prediction, we choose to adopt a Bayesian approach and integrate across scales. Figure 2 shows the estimated density of two observations of Model (1). Posteriors of the conditional density $F_{Y|X}$ were computed various sample sizes. Figure 2 suggests that our estimate of $F_{Y|X}$ approaches the true density as the number of observations in the training set increases. We are unable to compare our strategy for posterior estimation to previous literature because we are unaware of previous Bayesian approaches for this problem that scale up to problems of this size. Therefore, we numerically compare the performance of our point-estimates (which we define as the posterior mode of $\hat{F}_{Y|X}$) with the predictions of the competitor algorithms.

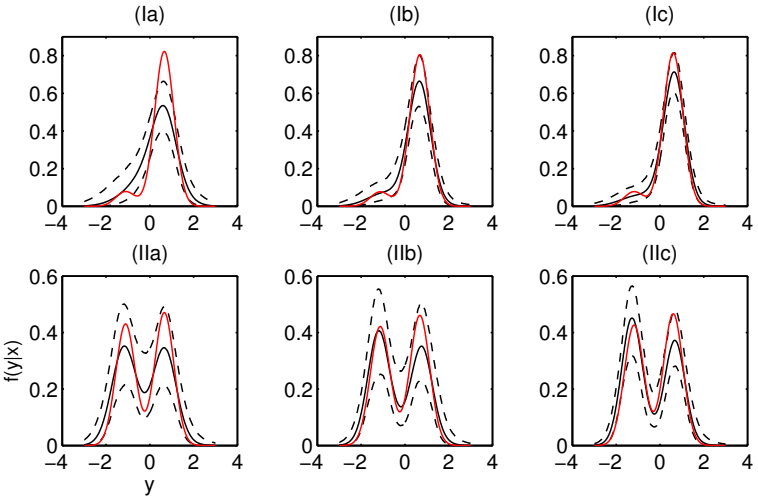

Figure 2: Illustrative example suggesting that our posterior estimates of the conditional density are converging as n increases even when $F_{Y|\eta}$ is highly nonlinear and $F_{X|\eta}$ is very high-dimensional. We simulated data according to Model (1) with parameters $(\mu_1, \sigma_1) = (-2, 1)$, $(\mu_2, \sigma_2) = (2, 1)$, $\sigma_x = 0.01$, and $c = 20$ for different sample sizes. True (red) and estimated (black) density (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) for two data points (I, II) considering different training set size (a: $n = 100$, b: $n = 150$, c: $n = 200$).

5.2 Quantitative Comparisons for Simulated Data

Figure 3 compares the numerical performance of our algorithm (MSB) with Lasso (black), CART (red), and PC regression (green) in terms of both mean-squared error (top) and CPU time (bottom) for models (2), (3), and (4) in the left, middle, and right panels respectively. These figures show relative performance on a *per simulation basis*, thus enabling a much more powerful comparison than averaging performance for each algorithm over a set of simulations. Note that these three simulations span a wide range of models, including nonlinear smooth manifolds such as the swissroll (model 2), relatively simple linear subspace manifolds (model 3), and a union of linear subspaces model (model 4).

In terms of predictive accuracy, the top panels show that for all three simulations, in every dimensionality that we considered—including $p \in \mathcal{O}(10^6)$—MSB is more accurate than either Lasso, CART, or PC regression. Note that this is the case even though MSB provides much more information about the posterior $F_{Y|X}$, yielding an entire posterior over $F_{Y|X}$, rather than merely a point estimate.

In terms of computational time, MSB is much faster than the competitors for large p and n, as shown in the bottom three panels. The supplementary materials show that computational time for MSB is relatively constant as a function of p, whereas Lasso's computational time grows considerably with p. Thus, for large enough p, MSB is significantly faster that Lasso. MSB is faster than CART and PC regression for all p and n under consideration. Thus, it is clear from these simulations that MSB has better scaling properties—in terms of both predictive accuracy and computational time—than the competitor methods.

5.3 Quantitative Comparisons for Neuroscience Applications

Table 1 shows the mean and standard deviation of point-estimate predictions per subject (using leave-one-out) for the two neuroscience applications that we investigated: (i) predicting creativity from diffusion MRI (creativity) and, (ii) predicting head motion based on functional MRI (movement). For the creativity application, p was relatively small, "merely" $2,415$, so we could run Lasso, CART, and random forests (RF) [**?**]. For the movement application, p was greater than $100\,000$.

For both applications, MSB yielded improved predictive accuracy over all competitors. Although CART and Lasso both yielded better performance on the relatively low-dimensional predictor ex-

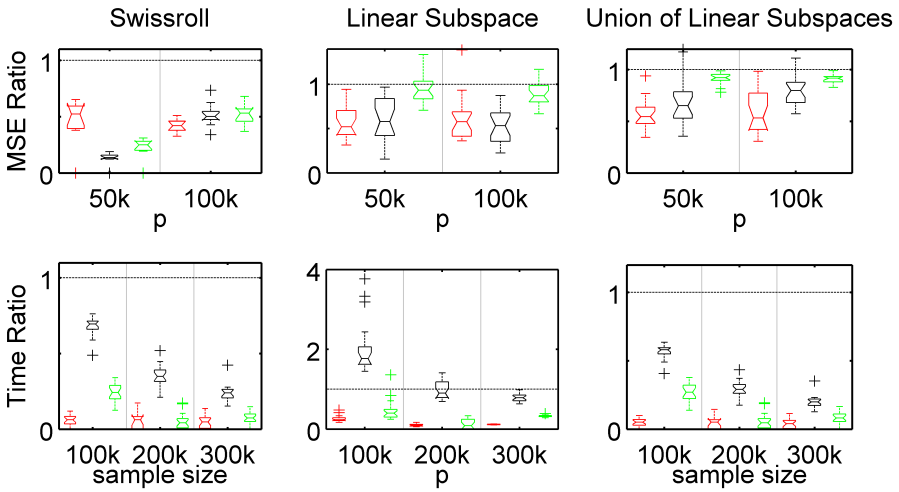

Figure 3: Numerical results for various simulation scenarios. Top plots depict the relative mean-squared error of MSB (our approach), versus CART (red), Lasso (black), and PC regression (green) for as a function of ambient dimension of x. Bottom plots depict the ratio of CPU time as a function of sample size. The three simulation scenarios are: swissroll (left), linear subspaces (middle), union of linear subspaces (right). MSB outperforms both CART, Lasso, and PC regression in all three scenarios regardless of ambient dimension ($r^{\mathcal{A}}_{mse} < 1$ for all p). MSB compute time is relatively constant as n or p increase, whereas Lasso's compute time increases, thus, as n or p increase, MSB CPU time becomes less than Lasso's. MSB was always significantly faster than CART and PC regression, regardless of n or p.

Table 1: Neuroscience application quantitative performance comparisons. Squared error predictive accuracy per subject (using leave-one-out) was computed. we report the mean and standard deviation (s.d.) across subjects of squared error, as well as CPU time (in seconds). We compare multiscale stick-breaking (MSB), CART, Lasso and random forest (RF). MSB outperforms all the competitors in terms of predictive accuracy and scalability for both applications.

DATA	n	p	MODEL	MSE (S.D.)	TIME (S.D.)
CREATIVITY	108	2,415	MSB	0.56 (??)	1.1 (0.02)
			CART	1.10 (??)	0.9 (0.01)
			LASSO	0.63 (??)	0.40 (0.10)
			RF	0.57 (??)	78.2 (0.59)
MOVEMENT	56	10^5	MSB	0.76 (??)	20.98 (2.31)
			LASSO	1.02 (??)	96.18 (9.66)

ample (creativity), their computational scaling was poor, such that CART yielded a memory fault on the higher-dimensional case, and Lasso required substantially more time than MSB.

6 Discussion

In this work we have introduced a general formalism to estimate conditional distributions via multiscale dictionary learning. An important property of any such strategy is the ability to scale up to ultrahigh-dimensional predictors. We considered simulations and real-data examples where the dimensionality of the predictor space exceeded several hundred thousand. To our knowledge, no other approach to learn conditional distributions can run at this scale. Our approach explicitly assumes that the posterior $F_{Y|X}$ can be well approximated by projecting X onto a lower-dimensional space, $F_{Y|X} \approx F_{Y|\eta}$, where $\eta \in \mathcal{M} \subset \mathbb{R}^d$, and $x \in \mathbb{R}^d$. Note that this assumption is much less restrictive than assuming that X is close to a low-dimensional space; rather, we only assume that the part of F_X that "matters" to predict Y lives near a low-dimensional subspace. Because a fully Bayesian strategy

remains computationally intractable at this scale, we developed a pseudo-Bayesian approach, fixing the partition tree, but integrating over scales and posteriors.

We demonstrate that even though we obtain posteriors over the conditional distribution $F_{Y|X}$, our approach, dubbed multiscale stick-breaking (MSB), outperforms several standard machine learning algorithms in terms of both predictive accuracy and computational time, as the sample size (n) and ambient dimension (p) increase. This improvement was demonstrated when the $\mathcal{M}$ was a swissroll, a latent subspace, a union of latent subspaces, and real data (for which the latent space may not even exist).

In future work, we will extend these numerical results this Bayesian theory. Indeed, while multiscale methods benefit from a rich theoretical foundation [**?**], the relative advantages and disadvantages of a fully Bayesian approach, in which one can estimate posteriors over all functionals of $F_{Y|X}$ at all scales, remains relatively unexplored.

we gotta remove about 10 references to make our reference list fit

References

[1] I. U. Rahman, I. Drori, V. C. Stodden, and D. L. Donoho. Multiscale representations for manifold- valued data. *SIAM J. Multiscale Model*, 4:1201–1232, 2005.

[2] W.K. Allard, G. Chen, and M. Maggioni. Multiscale geometric methods for data sets II: geometric wavelets. *Applied and Computational Harmonic Analysis*, 32:435–462, 2012.

[3] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixture of local experts. *Neural Computation*, 3:79–87, 1991.

[4] W. X. Jiang and M. A. Tanner. Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation. *Annals of Statistics*, 27:987–1011, 1999.

[5] J. Q. Fan, Q. W. Yao, and H. Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83:189–206, 1996.

[6] J. Q. Fan and T. H. Yim. A crossvalidation method for estimating conditional densities. *Biometrika*, 91:819–834, 2004.

[7] M. P. Holmes, G. A. Gray, and C. L. Isbell. Fast kernel conditional density estimation: a dual-tree Monte Carlo approach. *Computational statistics & data analysis*, 54:1707–1718, 2010.

[8] G. Fu, F. Y. Shih, and H. Wang. A kernel-based parametric method for conditional density estimation. *Pattern recognition*, 44:284–294, 2011.

[9] D. J. Nott, S. L. Tan, M. Villani, and R. Kohn. Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics*, 21:797–820, 2012.

[10] M. N. Tran, D. J. Nott, and R. Kohn. Simultaneous variable selection and component selection for regression density estimation with mixtures of heteroscedastic experts. *Electronic Journal of Statistics*, 6:1170–1199, 2012.

[11] A. Norets and J. Pelenis. Bayesian modeling of joint and conditional distributions. *Journal of Econometrics*, 168:332–346, 2012.

[12] J. E. Griffin and M. F. J. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101:179–194, 2006.

[13] D. B. Dunson, N. Pillai, and J. H. Park. Bayesian density regression. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 69:163–183, 2007.

[14] D. B. Dunson, N.S. Pillai, and J. H. Park. Bayesian density regression. *Journal of the Royal Statistical Society*, 69:163–183, 2007.

[15] Y. Chung and D. B. Dunson. Nonparametric Bayes conditional distribution modeling with variable selection. *Journal of the American Statistical Association*, 104:1646–1660, 2009.

[16] S. T. Tokdar, Y. M. Zhu, and J. K. Ghosh. Bayesian density regression with logistic Gaussian process and subspace projection. *Bayesian Analysis*, 5:319–344, 2010.

[17] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[18] J. H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–141, 1991.

[19] L. Breiman. Bagging predictors. *Machine Learning*, 24:123140, 1996.

[20] R. Shapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26:16511686, 1998.

[21] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[22] I. Mossavat and O. Amft. Sparse bayesian hierarchical mixture of experts. *IEEE Statistical Signal Processing Workshop (SSP)*, 2011.

[23] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing 20*, 1:359392, 1999.

[24] J. Sethuraman. A constructive denition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[25] Didier Chauveau and Jean Diebolt. An automated stopping rule for mcmc convergence assessment. *Computational Statistics*, 14:419–442, 1998.

[26] Rex E Jung, Rachael Grazioplene, Arvind Caprihan, Robert S Chavez, and Richard J Haier. White matter integrity, creativity, and psychopathology: Disentangling constructs with diffusion tensor imaging. *PloS one*, 5(3):e9818, 2010.

[27] R. Arden, R. S. Chavez, R. Grazioplene, and R. E. Jung. Neuroimaging creativity: a psychometric view. *Behavioural brain research*, 214:143–156, 2010.

[28] William R. Gray, John A Bogovic, Joshua T. Vogelstein, Bennett A Landman, Jerry L Prince, and R. Jacob Vogelstein. Magnetic resonance connectome automated pipeline: an overview. *IEEE pulse*, 3(2):42–8, March 2010.

[29] Susumu Mori and Jiangyang Zhang. Principles of diffusion tensor imaging and its applications to basic neuroscience research. *Neuron*, 51(5):527–39, September 2006.

[30] Abide.

[31] Sharad Sikka, Joshua T. Vogelstein, and Michael Peter Milham. Towards Automated Analysis of Connectomes: The Configurable Pipeline for the Analysis of Connectomes (C-PAC). In *Organization of Human Brain Mapping*. Neuroinformatics, 2012.

[32] Qi-Hong Zou, Chao-Zhe Zhu, Yihong Yang, Xi-Nian Zuo, Xiang-Yu Long, Qing-Jiu Cao, Yu-Feng Wang, and Yu-Feng Zang. An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: fractional ALFF. *Journal of neuroscience methods*, 172(1):137–141, July 2008.

[33] J. D. Power, K. A. Barnes, C. J. Stone, and R. A. Olshen. Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. *Neuroimage*, 59:2142–2154, 2012.