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) $x = (x_1, x_2, \dots, x_p) \in \mathcal{X}$. 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 and n 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 features [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 $X \colon \Omega \to \mathcal{X} \subseteq \mathbb{R}^p$ 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 $Y \colon \Omega \to \mathcal{Y} \subseteq \mathbb{R}$ be a scalar-valued target random variable, taking values $y \in \mathcal{Y}$, with a marginal probability distribution F_Y. Our goal is to develop an approach that facilitates obtaining an estimate of $F_{Y|X}$ given n pairs of observations that we assume are sampled exchangeable from the joint distribution, $(x_i, y_i) \sim F_{X,Y}$. Let $\mathcal{D}_n = \{(x_i, y_i)\}_{i \in [n]}$, where $[n] = \{1, \dots, n\}$.

Our approach is indirect. Rather than directly estimating $F_{Y|X}$, we posit the existence of a latent random variable $Z \colon \Omega \to \mathcal{Z} \subseteq \mathcal{X}$, where $\mathcal{Z}$ is only d "dimensional", where $d \ll p$. Note that $\mathcal{Z}$ need not be a linear subspace of $\mathcal{X}$, rather, $\mathcal{Z}$ could be, for example, a union or affine subspaces, or a smooth compact Riemannian manifold. Regardless of the nature of $\mathcal{Z}$, we assume that we can approximately decompose the joint distribution as follows, $F_{X,Y,Z} = F_{X,Y|Z} F_Z \approx F_{X|Z} F_{Y|Z} F_Z$. In other words, we assume that the *conditional signal* approximately concentrates around a low-dimensional latent space. 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. Let $Z \sim U(0,1)$, and let F_X be constructed as follows. $x_1 = z\sin(z)$, $x_2 = z\cos(z)$, and $X_j \sim \mathcal{N}(0,1)$ for all $j \in \{3, \ldots, 50\}$. Moreover, let $Y \sim \mathcal{N}(Z, Z+1)$. Thus, clearly, Y is conditionally dependent on Z, which is the low-dimensional signal manifold, of which X is also a function. In particular, X lives on a swissroll embedded in a 50-dimensional ambient space, but Y is only a function of where Z is along the swissroll. Figure 1 depicts this concrete example.

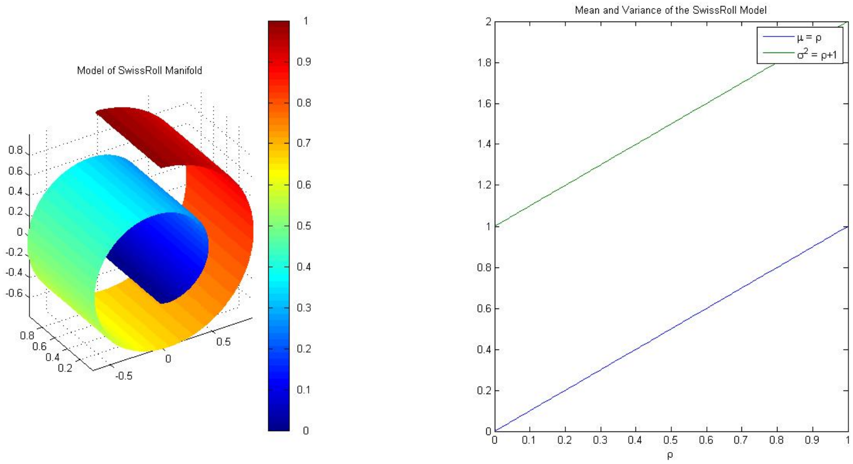

Figure 1: swissroll

3 Methodology

We propose here a general modular methodology consisting of four components: (i) a tree decomposition of the space, (ii) an embedding $\psi \colon \mathcal{X} \to \mathcal{Z}$ of the x_i's that we assume the y_i's are dependent upon, (iii) an assumed form of the conditional probability model, $\mathcal{P}_{Y|\psi(X)}$, and (iv) a prior over scales, π.

Tree Decomposition A tree decomposition yields a multiscale partition of the data. 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 $w \in \mathcal{W}$. We define a tree decomposition as in [**?**]:

Definition 1 *A* tree decomposition τ *of a d-dimensional metric measure space* $(\mathcal{W}, \rho_W, F_W)$ *is a collection of open sets* $\{C_{j,k}\}_{j \in \mathcal{K}_k, k \in \mathbb{Z}}$*, called* cells*, satisfying:*

(i) for all $j \in \mathbb{Z}$, $F_W(\mathcal{W} \setminus \cup_{k \in \mathcal{K}_j} C_{j,k}) = 0$,

(ii) for all $j' \geq j$ and $k' \in \mathcal{K}_j$, either $C_{j',k'} \subseteq C_{j,k}$ or $F_W(C_{j',k} \cap C_{j,k}) = 0$,

(iii) for $j < j'$ and $k' \in \mathcal{K}_{j'}$, there exists a unique $k \in \mathcal{K}_j$ such that $C_{j',k'} \subseteq C_{j,k}$,

(iv) each $C_{j,k}$ contains a point $c_{j,k}$ such that $B^{\mathcal{W}}_{r \cdot 2^{-j}}(c_{j,k}) \subseteq C_{j,k} \subseteq B^{\mathcal{W}}_{2^{-j}}(c_{j,k})$ for a constant r depending on the intrinsic geometric properties of $\mathcal{W}$. In particular, we have $F_{j,k} \approx 2^{-dj}$.

The first condition means that at each scale, the set of cells covers the space almost everywhere. The second and third condition together mean that as we descend the tree, going from a coarser to finer scale, for a given cell at scale j, it is the child of a single coarser scale cell, such that its intersection with any other coarser scale cell has measure zero. The fourth condition means that there exists a point $c_{j,k}$ which is effectively the location of cell $C_{j,k}$, in that it is in $C_{j,k}$, and that all other points in $C_{j,k}$ are within a small distance of $c_{j,k}$. Let $C(w) = \{C_{j,k}(x)\}_{j \in \mathcal{K}_j, j \in \mathbb{Z}}$ be the *path* along the tree for point $w \in \mathcal{W}$. Moreover, 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 $C_{j,k}$. Given a tree decomposition, we can approximate $F_{Y|X}$ at each scale j by $\cup_{k \in \mathcal{K}_j} F_{Y|C_{j,k}}$. Figure 2 depicts a tree decomposition for some data.

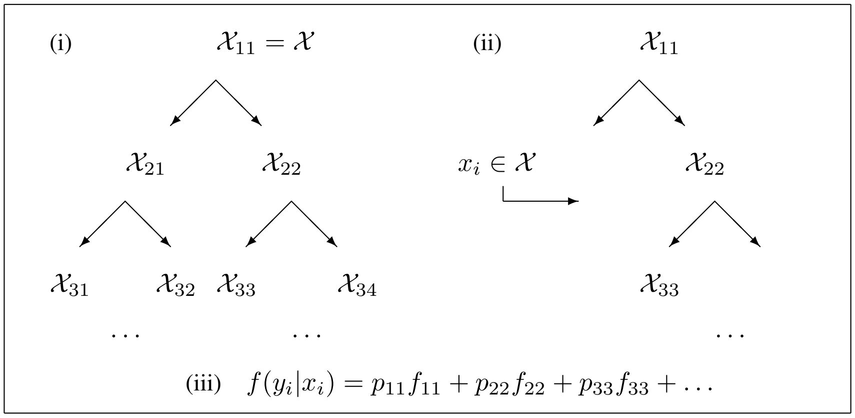

Figure 2: (i) Multiscale partition of the data. (ii) Path through the tree for $x_i \in \Re^q$. (iii) Conditional density of y_i given x_i defined as a convex combination of densities along the path.

Embeddings At each scale, for each cell, we consider some function $\psi_{j,k} \colon C_{j,k} \to \mathcal{Z}$ that maps each point in $C_{j,k}$ to the latent space. Thus, we further approximate $F_{Y|X}$ at scale j by $\cup_{k \in \mathcal{K}_j} F_{Y|\psi(C_{j,k})}$. For example, the authors of [**?**] chose ψ to be a linear projection of the data onto the best fitting d-dimensional hyperplane. In general, one can either choose or learn these embeddings.

Family Each $F_{Y|\psi(C_{j,k})}$ is an element of a family of distributions, $\mathcal{F}$. This family might be quite general, e.g., all possible conditional densities, or quite simple, e.g., Gaussian distributions.

Prior We further assume a prior, $\pi = \{\pi_{j,k}\}_{j \in \mathcal{K}_j, j \in \mathbb{Z}}$ over all paths of the tree. This prior facilitates finding an optimal bias/variance tradeoff, without having to choose a particular scale.

Thus, collectively, any multiscale Bayesian conditional density estimation procedure can be implemented using Pseudocode 1.

Pseudocode 1 Generic "Ms. Deeds" (MultiScale Dictionary learning for conditional DistributionS)

```
Input: the data $\mathcal{D}_n$ and the following choices: (i) a partitioning scheme $\Lambda$, (ii) a class of embeddings $\Psi$, (iii) a family of potential conditional densities $\mathcal{F}$, (iv) a prior $\pi$ over scales
  Estimate a multiscale partition $\tau$ from the data $\mathcal{D}_n$ using $\Lambda$
  Estimate or choose embeddings $\psi_{j,k} \colon C_{j,k} \to \mathbb{Z}$, where $\psi \in \Psi$
  Estimate $F_{Y|\psi_{j,k}(C_{j,k})}$ from the family $\mathcal{F}$
  Let $\hat{F}_{Y|X=x_i} = \sum_j \hat{F}_{Y|\psi_{j,k}(x_i):x \in C_{j,k}} \pi_{j,k}$
Output: $\hat{F}_{Y|X=x_i}$
```

Specific Choices

(i) We let Λ be 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 ρ_W and thresholding.

(ii) We let each $\psi_{j,k}$ simply be a Dirac delta function. We therefore alleviate the computational and theoretical difficulties of estimating hyperplanes via SVD and choosing the dimensions thereof, rather, we assume $\psi_{j,k}(x) = 1$ if an only if $x \in C_{j,k}$. The Dirac function can result in significant computational savings versus linear embedding for each partition $C_{j,k}$. In particular, the ratio of computational costs between partitioning and linear embedding is $\mathcal{O}(3^d/d^2)$, where d is the intrinsic dimension [**?**]. Thus, for low intrinsic dimension, or when the constants are high, SVD computation can dominate. For the simulations we present below, this is indeed the case; in many real data applications, however, intrinsic dimension is quite high. One can think of our approach as using an empirical estimate of the marginal density of F_X, whereas theres is more like fitting a Gaussian to each partition.

(iii) We let $\mathcal{F}$ be Gaussian for simplicity.

(iv) We let π be generated by a stick-breaking process [24]. For each node $C_{j,k}$ in the partition tree, 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 $\sum_{j=1}^{k} \pi_{j,k} = 1$. 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 posteriors 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. Moreover, by integrating over all paths, we are able to borrow information adaptively to the observed data.

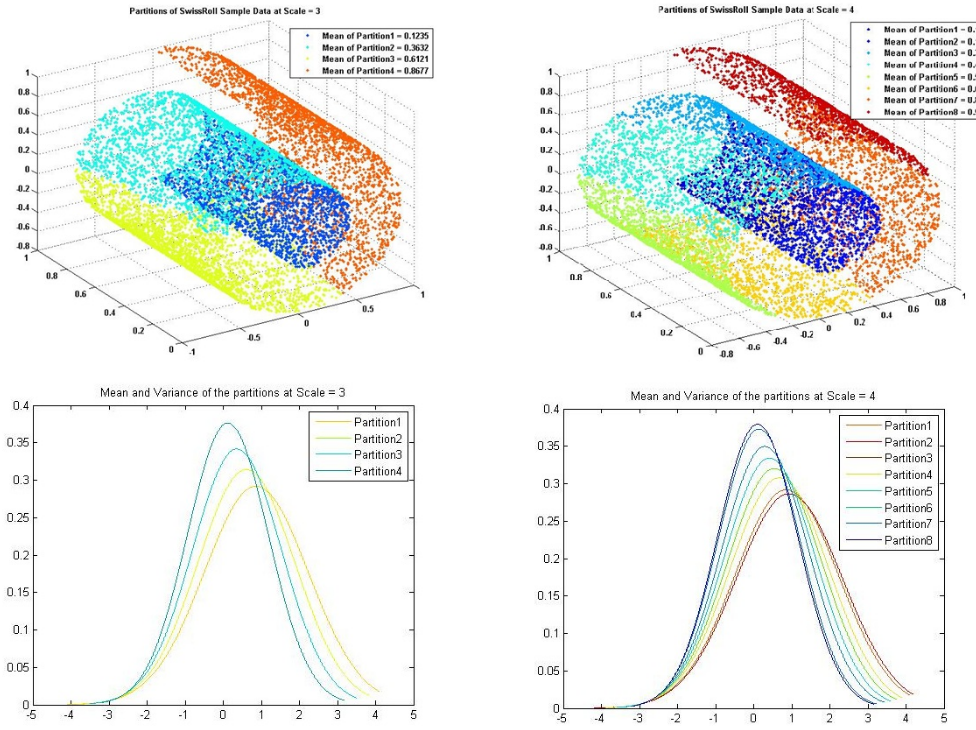

Figure 3: caption

4 Estimation

Parameters involved in the dictionary densities can be estimated using either frequentist or Bayesian methods. Bayesian methods are appealing since they can avoid singularities associated with traditional maximum likelihood inference, the prior has an appealing role as a regularizer, and we can characterize uncertainty in dictionary learning through the resulting posterior. Hence, parameters involved in dictionary densities will be estimated through Bayesian methods and inference on stick breaking weights and dictionary density parameters will be carried out using the Gibbs sampler. Details on full conditionals and Gibbs sampler steps can be found in the supplementary material.

5 Simulation studies

In order to assess the predictive performance of the proposed model, different simulation scenarios were considered. Let n be the number of observations, $y \in \Re$ the response variable and $x \in \Re^p$ a set of predictors. The Gibbs sampler was run considering $20,000$ as the maximum number of iterations with 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 σ_{B_j}s and V_{B_j}s were set respectively equal to $(3, 1)$ and 1.

In all simulation scenarios, predictors were assumed to lie close a r-dimensional space, either a lower dimensional plane or a non linear manifold, with $r << p$. For each synthetic dataset, the proposed model was compared with CART and lasso in terms of mean squared error. Mean squared errors were computed based on leave-one=out predictions. For CART and Lasso standard Matlab packages were utilized and the regularization parameter of Lasso was chosen based on the AIC. The supplementary material includes more results about experiments involved in this section.

5.1 Illustrative Example

First let us consider the simple toy example of §2. We created an equally spaced grid of points $t_i = 0, \dots, 20$. Then, we let $\eta_i = \sin(t_i)$ and predictors be a linear function of η_i plus Gaussian noise, i.e. $x_{ij} = \eta_i + \epsilon_{ij}$ with $\epsilon_{ij} \sim N(0, 0.1)$ for $j \in \{1, \dots, p\}$. In particular, we set $p = 1,000$. The response was drawn from the following mixture of Gaussians

$$y_i \sim w_i \mathcal{N}(-2, 1) + (1 - w_i)\mathcal{N}(2, 1) \tag{1}$$

with $w_i = |\eta_i|$. Figure 4 shows the estimated density of two data points. These estimates were obtained by performing leave-one-out prediction for different number of observations in the training set. As the figure clearly shows our construction facilitates an estimate of the density y that become closer to the true density as the number of observations in the training set increases.

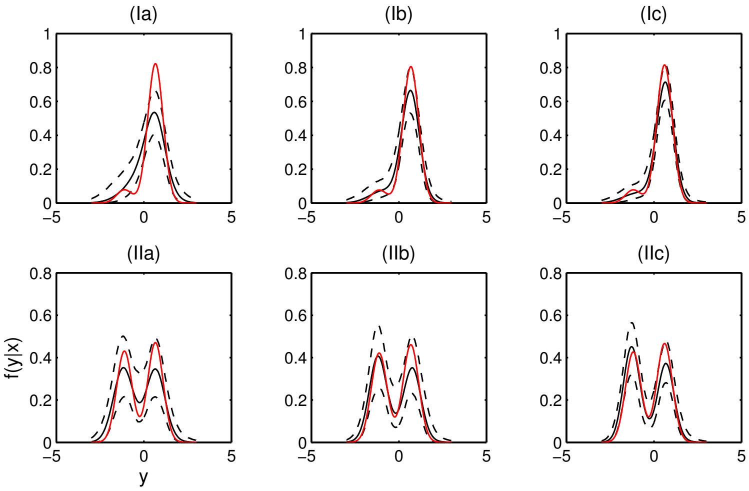

Figure 4: Illustrative example: Plot of true (red dashed-dotted line) and estimated (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) density for two data points (I, II) considering different training set size (a:100, b:150, c:200).

5.2 Linear lower dimensional space

In this section, predictors and response were assumed to lie close to a lower dimensional plane. In practice, we modeled $z_i = (y_i, x_i^t)^t$ through the following factor model

$$z_i = \Lambda \eta_i + \epsilon_i \tag{2}$$

with $\epsilon_i \sim \mathcal{N}(0, \Sigma_0)$, $\Sigma_0 = diag(\sigma_1, \dots, \sigma_p)$, Λ being a $p \times r$ matrix, $\eta_i \sim \mathcal{N}(0, I)$ and $r << p$. The loading matrix was derived as the product of a matrix with orthogonal columns and a diagonal matrix with positive elements on the diagonal, i.e. $\Lambda = \Gamma\Theta$. In particular, the columns of Γ were uniformly sampled from the Stiefel manifold while the diagonal matrix of Θ were sampled from an inverse Gamma with shape and rate parameters $(1, 4)$. We set $r = 5$.

As measure of comparison we adopted the ratio between cpu times and the ratio between mean squared errors. Define t_m^ℓ as

$$t_m^\ell = m(MSB)/m(\ell)$$

where m is either CPU time expressed in seconds or mean squared errors, MSB is our approach and ℓ is the competitor. For each simulation scenario, we sampled M datasets so that M values of t_m^ℓ were obtained. We sampled $M = 20$ datasets involving 100 observations and for each method we performed leave-one-out predictions. Figure 5(I) shows boxplots of t_{mse}^ℓ as p increases. Clearly, our method outperforms the competitors in terms of mean squared errors. Furthermore, as shown in figure 5(II), our approach can scale substantially better than competitors to huge dimensions of the predictor space.

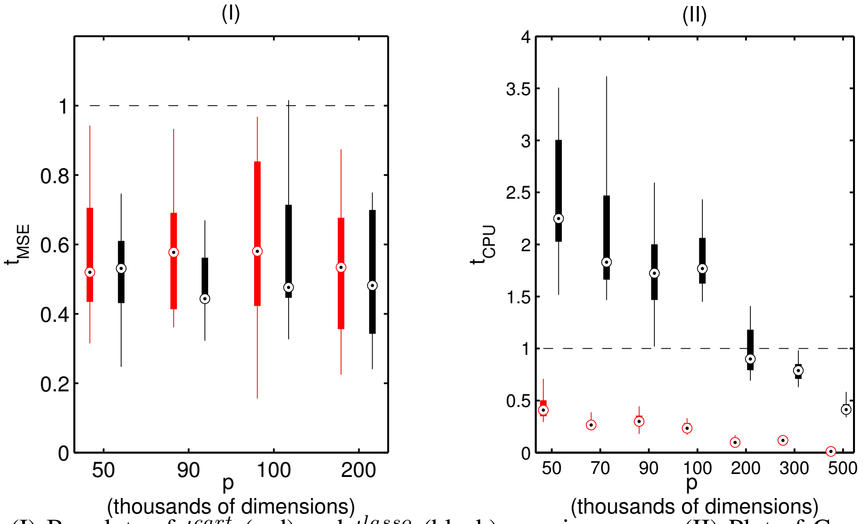

Figure 5: (I) Boxplots of t_{mse}^{cart} (red) and t_{mse}^{lasso} (black) as p increases. (II) Plot of Cpu time (in seconds) for Lasso (dash), Cart (solid) and MSB (dash-dot) under the first simulation scenario

5.3 Non-Linear lower dimensional space

In this section predictors were assumed to lie close to a lower dimensional non-linear manifold. In the first simulation study, predictors and response were jointly sampled from an N components mixture of factor analyzers (MFA). For each mixture components, the loading matrix and variances were sampled as in §5.2, while mixture weights were sampled from a $Dirichlet(1, \dots, 1)$. The number of latent factors was considered to be increasing in the number of components. In practice, we let the hth mixture component be modeled through h factors. We set $N = 5$ (the supplementary material shows results for different numbers of mixture components). In the other simulation scenario predictors were assumed to lie close to the Swissroll manifold (see figure 1 in the supplementary material), a two dimensional manifold embedded in $\Re^p$ while the response was sampled from a normal with mean equal to one coordinate of the manifold and standard deviation one.

For both data scenarios, we sampled $M = 20$ datasets involving 100 observations and we performed leave-one-out predictions. Figure 6(I) and 6(III) show boxplots of mean squared errors as

p increases. Again our model is associated to better predictive performance compared to Cart and lasso. To show how the performance of our model varies for different sample sizes, we sampled datasets involving different number of observations. In practice, the dimension of the predictor space was considered fixed, i.e. $p = 300,000$ and ratios t^{ℓ}_{cpu} were computed considering sample sizes $n \in \{100, 200, 300\}$. As shown in figure 6(II) and 6(IV), the gap between our model and competitors improves as n increases.

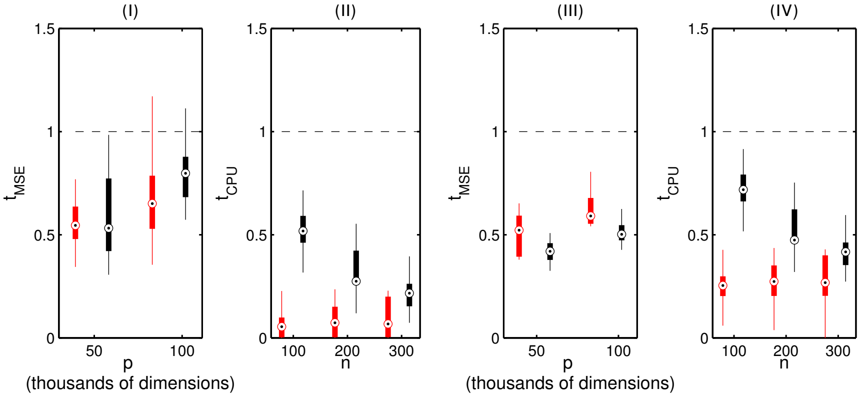

Figure 6: Boxplots of t^{ℓ}_{mse} under data drawn from MFA (I) and swissroll (III) and boxplots of t^{ℓ}_{cpu} for a fixed $p = 300,000$ and different sample sizes (n) under data drawn from MFA (II) and swissroll (IV) with $\ell = Cart$ (red) and $\ell = Lasso$ (black)

6 Real application

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 same prior specification and Gibbs sampler as in §3 was utilized.

In the first experiment we investigated the extent to which we could predict creative (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 $\binom{70}{2} = 2,415$ potential weighted edges. The vector of covariates consists in the natural logarithm of the total number of connections between all pairs of cortical regions, i.e. $p = 2,415$.

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].

For the first data example, we compared our approach (multiscale stick-breaking; MSB) to CART, lasso and random forests. Table 1 shows that MSB outperforms all the competitors in terms of mean square error; this is in addition to yielding an estimate of the entire conditional density for each y_i. It is also significantly faster that random forests, the next closest competitor, and faster than lasso. For this relatively low-dimensional example, CART is reasonably fast. For the second data application, given the huge dimensionality of the predictor space, we were unable to get either CART or random forest to run to completion, yielding memory faults on our workstation (Intel Core i7-2600K Quad-Core Processor memory 8192 MB). We thus only compare performance to lasso. As in the previous example, MSB outperforms lasso in terms of predictive accuracy measured via mean-squared error, and significantly outperforms lasso in terms of computational time.

Table 1: Real Data: Mean and standard deviations of squared error under multiscale stick-breaking (MSB), CART, Lasso and random forest (RF).

DATA	n	p	MODEL	MSE	t_T	t_M	t_V
(1)	108	2,415	MSB	0.56	100	1.1	0.02
			CART	1.10	87	0.9	0.01
			LASSO	0.63	50	0.40	0.10
			RF	0.57	$7,817$	78.2	0.59
(2)	56	$10e+05$	MSB	0.76	690	20.98	2.31
			LASSO	1.02	$5,836$	96.18	9.66

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 definition 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.