Multiresolution dictionary learning for 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 multiresolution 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].

In general, 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 many orders of magnitude or more larger than this, and require 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 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 multiresolution approach which starts by learning a multiscale dictionary of densities, constructed as Gaussian within each set of a multiscale partition tree for the features. 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 is how to estimate the corresponding multiresolution probability tree. This is accomplished in a Bayesian manner using a novel multiresolution 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 Proposed approach

2.1 Setting

Let $x \in \mathcal{X} \subseteq \Re^p$ be a p-dimensional Euclidean vector-valued predictor random variable. Let $f(x)$ denote the marginal probability density of x. We assume that $f(x)$ concentrates around a lower-dimensional, possibly nonlinear, subspace $\mathcal{M}$. For example, $\mathcal{M}$ could be a union of affine subspaces, or a smooth compact Riemannian manifold.

Let $y \in \mathcal{Y} \subseteq \Re$ be a real-valued target variable. We further assume that the conditional distribution is a function of only the position μ of x within the subspace $\mathcal{M}$, $f(y|x) = f(y|\mu)$. Let x and y be sampled from some true but unknown joint distribution. We would like to learn $f(y|x)$. We assume that we obtain n independently and identically sampled observations, (x_i, y_i), for $i \in \{1, 2, \ldots, n\}$. Our proposed model is very general in accommodating an unknown density $f(y|x)$ which changes according to the location of x in the lower-dimensional subspace. However, for exposition and testing of the model, it is useful to consider a simple example in which x lives on a smooth one-dimensional Riemannian submanifold embedded in $\Re^p$, and y is a univariate Gaussian random variable whose mean and variance vary with the location of x along its geodesic.

We can formalize this simple example model as follows. Consider

$$x_i \sim \mathcal{N}(\psi(\mu_i), \sigma^2 I),$$

where $\Psi = \{\psi \colon \mathcal{M} \to \Re^p\}$, $\mu_i \in \mathcal{M}$, $\sigma \in (0, \infty)$, I is the $p \times p$ dimensional identity matrix. Let $\mathcal{M}$ be a smooth compact Riemannian manifold, such as the swissroll or the S-manifold. For simplicity, let us assume that $\mathcal{M}$ is a curve. Let $\psi(\mu) = 1\mu$ with 1 being a p-dimensional vector with all elements equal to 1. Define the conditional $f(y|x)$ as a function of μ, i.e. a mixture density with mixture weights depending on μ. We will show in §3 that our construction facilitates an estimate of the density of y.

2.2 Overview

We aim to build a flexible and scalable model for the density of $y_i \in \Re$ given a set of predictors $x_i \in \mathcal{X} \subseteq \Re^p$. Our approach proceeds in a two stage fashion as follows. We first learn a multiscale nonlinear partition tree of the feature data. Starting from the coarsest scale, corresponding to the entire set $\mathcal{X}$, denoted as $\mathcal{X}_{11}$, each set $\mathcal{X}_{mj}$ is split into two or more mutually exclusive subsets. This process continues until some convergence criteria is satisfied, e.g. the number of observations allocated to the finest scales is below some chosen threshold. Figure 1(i) shows a dyadic partition of the predictor space where a generic set $\mathcal{X}_A$ is partitioned into two subsets $\mathcal{X}_B$ and $\mathcal{X}_C$ with $\mathcal{X}_A = \mathcal{X}_B \cup \mathcal{X}_C$ and $\mathcal{X}_B \cap \mathcal{X}_C = \emptyset$. Considering this tree partition of $\mathcal{X}$, each $x_i \in \mathcal{X}$ has an associated path characterized by the sets including x_i (see figure 1(iii)). We assume that the density of y_i depends on x_i through this tree partition. Specifically, the conditional density $f(y_i|x_i)$ will be a mixture of densities with component-specific parameters depending on the sets contained in the path of x_i (see figure 1(iv)). In the extreme case in which two predictor values x and x' belong to the same leaf partition sets, the conditional distributions $f(y'|x')$ and $f(y|x)$ will be identical. If the two paths differ only in the final generation or two, the conditional densities will typically be similar but not identical.

Before illustrating the model let us introduce some notation. Let L be the number of levels of the partition tree. Let us assume each set in the partition tree is split into two subsets (as shown in figure 1(i)). Let $A_m(x) \in \{1, \ldots, 2^{m-1}\}$ be the location of predictor x at level m, with $A_1(x)$ equal to 1 by definition and the vector $A(x) = [A_1(x), \ldots, A_L(x)]^T$ encoding the path through the partition tree up to generation L specific to predictor value x. Let $de(h, s)$ and $an(h, s)$ be respectively the set of descendants and ancestors of node (h, s). Define $B_j(x)$ as the node at generation j containing the vector of features x, i.e. $B_j(x) = \{j, A_j(x)\}$. Each $B(x)$ is a set encoding the path through the partition tree up to generation L specific to predictor value x.

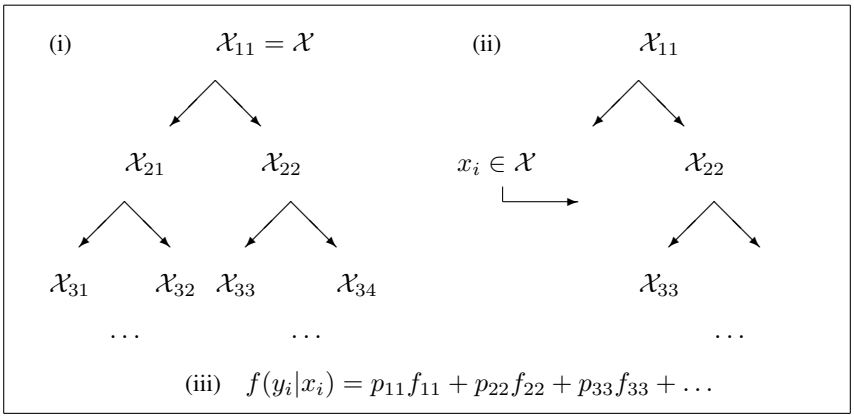

Figure 1: (i) Multiscale partition of the data. (ii) Path through the tree for $x_i \in \Re^q$. (iii) Conditional density of y_i given x_i defined as a convex combination of densities along the path.

2.3 Model Specification and estimation

The conditional density $f(y|x)$ is defined as the convex combination of densities $\{f_{B_j(x)}\}_{j=1}^{L}$ with weights $\{\pi_{B_j(x)}\}_{j=1}^{L}$, i.e.

$$f(y|x) = \sum_{j=1}^{L} \pi_{B_j(x)} f_{B_j(x)}, \tag{1}$$

with $(\pi_{B_j(x)}, f_{B_j(x)})$ being the weight and dictionary density associated to node $B_j(x)$, $0 \leq \pi_{B_j(x)}$ and $\sum_{j=1}^{L} \pi_{B_j(x)} = 1$. The pair (π_A, f_A) in 1 is specific to node A in the partition tree. According to model 1, only observations with predictors contained in A, i.e. $\{y_i : x_i \in A\}$, will have a mixture components with weight π_A and density f_A. Each density $f_{B_j(x)}$ will be defined as a univariate normal densities, i.e. $\mathcal{N}\left(\mu_{B_j(x)}, \sigma_{B_j(x)}\right)$. Notice that, as the weight associated to the first level of resolution approaches one, a non predictor-dependent density for y is obtained.

According to model (1), one observation can lie in subsets located at different resolution levels. This is critical in achieving a good compromise between bias and variance through borrowing information across different resolution levels. Though the proposed approach is reminiscent of a mixture of experts model [3], the two approaches are quite different, since under (1), neither mixture weights nor dictionary densities directly depend on predictors. This allows our model to scale efficiently to high dimensional predictors.

To derive mixture weights, a natural choice corresponds to a stick-breaking process [24]. For each node υ in the partition tree, define a stick length $V_\upsilon \sim Beta(1, \alpha)$ for nodes υ located from generation 1 to $k-1$. The parameter α encodes the complexity of the model, with $\alpha = 0$ corresponding to the case in which $f(y|x) = f(y)$. We relate the weights in (1) to the stick-breaking random variables as follows:

$$\pi_\upsilon = V_\upsilon \prod_{\zeta \in an\{\upsilon\}} (1 - V_\zeta),$$

with $V_\upsilon = 1$ for υ belonging to the last generation level. This condition will ensure that for each path with nodes $(\upsilon_1, \ldots, \upsilon_k)$, $\sum_{j=1}^{k} \pi_{\upsilon_j} = 1$. We refer to this prior as a multiresolution stick-breaking process.

The proposed approach is based on a two-stage algorithm where first the observations are allocated to different subsets in a tree fashion using an efficient partitioning algorithm and then, considering the partition as fixed, a multiresolution stick-breaking process is estimated. In practice, observations are partitioned applying metis [23], a fast multiscale technique used for graph partitioning. An overview of this algorithm can be found in the supplementary material.

Though more complicated densities can be considered, dictionary densities f_υ will be estimated by assuming a normal form, i.e. $f_\upsilon = \mathcal{N}(\mu_\upsilon, \sigma_\upsilon)$. Parameters involved in the 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. We will place a normal prior with parameter $(0, 1)$ on μ_υ and an inverse gamma prior with parameters a_σ and b_σ on σ_υ for each $\upsilon \in \mathcal{T}$. Details on full conditionals and Gibbs sampler steps can be found in the supplementary material.

3 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 belong to an 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. For CART and

Lasso standard Matlab packages were utilized. In order to fairly compare Lasso with the proposed model, a fast Lasso algorithm based on Lars was implemented and the regularization parameter was chosen based on the AIC.

3.1 Illustrative Example

First let us consider the simple toy example of §2.1. We created an equally spaced grid of points $t_i = 0, \ldots, 20$. Then, we let $\eta_i = \sin(t_i)$ and predictors be a linear function of η_i plus Gaussian noise, i.e. $x_i = \eta_i + \epsilon_i$ with $\epsilon_i \sim N(0, 0.1)$. The response was drawn from the following mixture of Gaussians

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

with $w_i = |\eta_i|$. Our model was run considering different sample sizes. Figure 2 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.

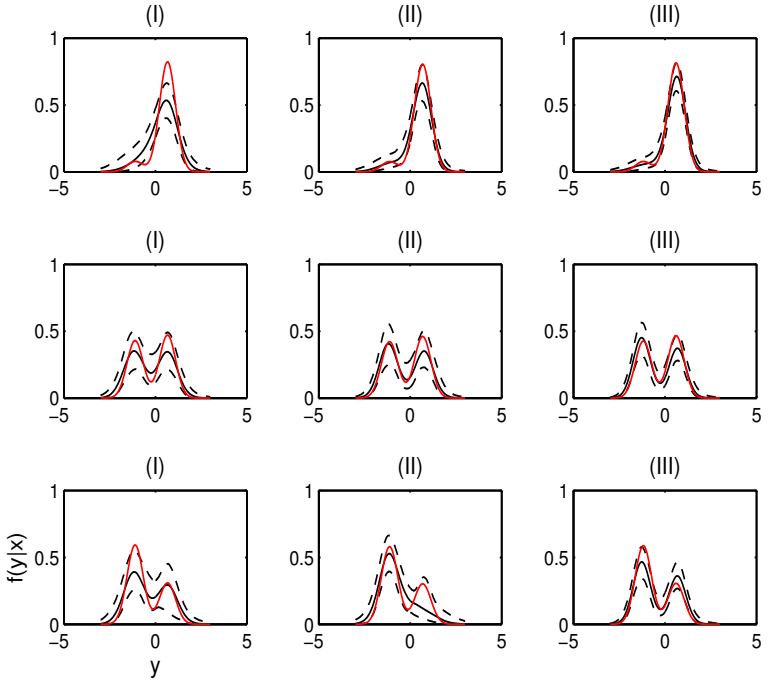

Figure 2: Illustrative example: Plot of true (red dashed-dotted line) and estimated (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) density for five data points (I, II, III, IV, V) considering different training set size (a:100, b:200, c:300).

3.2 Linear lower dimensional space

In this section, the vector of predictors was assumed to lie close to a lower dimensional plane. In practice, predictors were modeled through a factor model as follows

$$x_i = \Lambda \eta_i + \epsilon_i \tag{3}$$

with $\epsilon_i \sim \mathcal{N}(0, \Sigma_0)$, $\Sigma_0 = diag(\sigma_1, \ldots, \sigma_p)$, Λ being a $p \times r$ matrix, $\eta_i \sim \mathcal{N}(0, I)$ and $r << p$. In the first simulation scenario the response y was assumed to be a function of the latent variable η

so that the dependence between response and predictors was induced by the shared dependence on the latent factors. In practice, the pair (y_i, x_i) was jointly sampled from a factor model. 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 $(1, 4)$. In the second simulation scenario, x was sampled from a factor model with sparse loading while y was sampled from a normal with location and scale parameter $(1, 1)$ if the first variable was positive, i.e. $x_1 > 0$, and from a normal with location and scale $(-1, 1)$ otherwise. In this example, the non zero elements of the loading matrix were sampled from a normal with zero mean and standard deviation 3. In all the examples, an inverse gamma prior with parameters $(1, 4)$ were utilized for σ_j with $j = 1, \ldots, p$.

We sampled M datasets from the above simulation scenarios. Define $t^m(\ell)$ as the the ratio between mean squared errors under MSB and method ℓ for dataset m. Figure 3 shows boxplots of $\{t^m(Cart)\}_{m=1}^M$ and $\{t^m(Lasso)\}_{m=1}^M$ as p increases. Intuitively, when the ratio is lower than one, our model beats the competitors. Clearly, our method outperforms the competitors in terms of mean squared errors under both simulation scenarios.

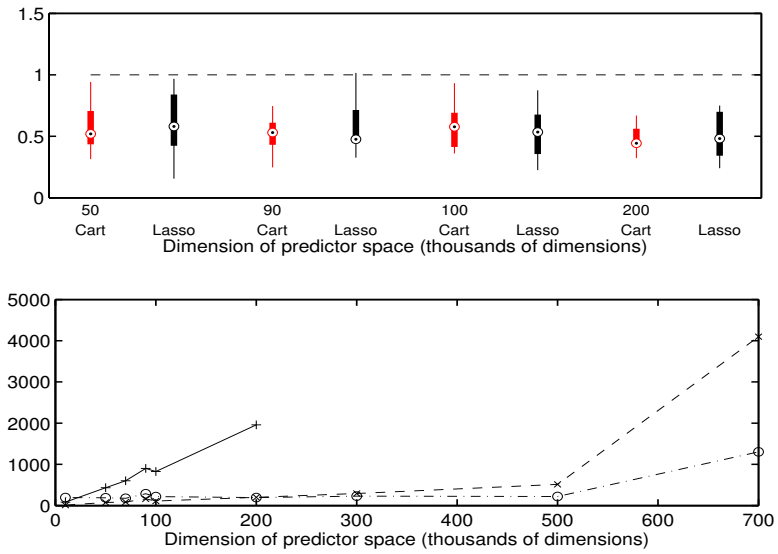

Figure 3:

3.3 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 so that the vector of predictors and response were assumed to lie close to N lower dimensional planes. For each mixture components, the loading matrix and variances were sampled as in the first simulation scenario in §3.2, while mixture weights were sampled from a Dirichlet distribution with parameter $\alpha_j = 1$ for $j = 1, \ldots, N$. 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. In the other simulation scenarios 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 of the coordinates of the manifold and standard deviation one.

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

4 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 §5 was utilized. 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]. We therefore let each $x_i \in \Re^p$ correspond to logarithm of each weighted edge; 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]. fALFF is a highly nonlinear transformation of the time-series data, previously demonstrated to be a reliable property of such data. 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 (multiresolution 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.

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.