Multiresolution dictionary learning for conditional distributions

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.

Key words: Density regression; Dictionary learning, Manifold learning; Mixture of experts; Multiresolution stick-breaking; Nonparametric Bayes.

1. Introduction

Massive datasets are becoming a ubiquitous by-product of modern scientific and industrial applications. These data present novel statistical and computational challenges for machine learning because many previously developed theoretical and methodological approaches do not scale-up sufficiently. Specifically, these data are problematic because of their 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 (Rahman et al., 2005; Allard et al., 2012).

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 (Jacobs et al., 1991; Jiang & Tanner, 1999), kernel methods (Fan et al., 1996; Fan & Yim, 2004; Holmes et al., 2010; Fu et al., 2011), Bayesian finite mixture models (Nott et al., 2012; Tran et al., 2012; Norets & Pelenis, 2012) and Bayesian nonparametrics (Griffin & Steel, 2006; Dunson et al., 2007a;b; Chung & Dunson, 2009; Tokdar et al., 2010). In general, there has been limited consideration of scaling to large p settings, with the variational Bayes approach of (Tran et al., 2012) being a notable exception; for dimensionality reduction, they 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; all sufficiently scal-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

able 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) (Breiman et al., 1984) and multivariate adaptive regression trees (MARS) (Friedman, 1991). 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 (Breiman, 1996), boosting (Shapire et al., 1998) and random forests (Breiman, 2001). 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 (Mossavat & Amft, 2011). 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; a non-convex 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 (Karypis & Kumar, 1999). 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. Setting

Let $x \in \mathcal{X} \subseteq \mathbb{R}^p$ 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 $y \in \mathcal{Y} \subseteq \mathbb{R}$ be a real-valued *target* variable. We further assume that the *conditional* distribution is a function of only the position μ of x along the manifold, $f(y|x) = f(y|\mu)$. Let x and y be sampled from some true but unknown joint distribution. We would like to know $f(y|x)$. In particular, we obtain an estimate of this conditional density via a *point cloud*. Specifically, we assume that we obtain n independently and identically sampled observations, (x_i, y_i), for $i \in \{1, 2, \ldots, n\}$. For example, x might live on some smooth one-dimensional Riemannian submanifold embedded in $\mathbb{R}^p$, and y could be a univariate Gaussian random variable whose mean and variance vary with the location of x along its geodesic.

We can formalize this model as follows. Consider $x_i \sim \mathcal{N}(\psi(\mu_i), \sigma^2 I_p)$, where $\Psi = \{\psi \colon \mathcal{M} \to \mathbb{R}^p\}$, $\mu_i \in \mathcal{M}$, $\sigma \in (0, \infty)$, I_p is the $p \times p$ dimensional identity matrix, and $\mathcal{N}(\cdot, \cdot)$ indicates a Gaussian distribution. 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 $\psi(\mu) = 1_p \mu$ with 1_p being a p-dimensional vector with all elements equal to 1. Define the conditional $f(y|x) = \mathcal{N}(\mu, g(\mu))$ for some positive function defined on $(0, \infty]$. In other words, both the mean and standard deviation of y depend on the position of x along its geodesic. We will show in §5 that our construction facilitates a smooth estimate of the mean and variance of y, even though we are not explicitly smoothing, rather, the smoothness is induced via the model averaging over spatial scales.

3. Model Specification

3.1. Approach

Our approach follows from assuming that the conditional distribution of the target variable is a simple function of a low-dimensional representation of the predictor variable embedded in a high-dimensional ambient space. We pursue a two-stage strategy. In the first stage, we try to find a multiscale nonlinear partitioning of the data. In other words, we recursive partition $\{x_i\}_{i=1}^n$ to obtain subsets that are increasingly homogeneous according to some metric. Thus, associated with each sample i is a *path* along the partition tree encoding to which child i belongs in each scale of the tree. In the second stage, we estimate the conditional distribution of the target variable as a function of the multiscale embedding of the predictors.

3.2. Model Structure

Suppose we define a multiscale partition of $\mathcal{X}$. Generation one corresponds to the entire $\mathcal{X}$ denoted as $\mathcal{X}^1$. At generation two, $\mathcal{X}^1$ is split into two mutually exclusive partition sets, $\mathcal{X}^1 = \left(\mathcal{X}_1^2, \mathcal{X}_2^2\right)$. Each subset is recursively partitioned into two subsets so that for a general partition level ℓ the partition will be given by $\mathcal{X}^\ell = \left(\mathcal{X}_1^\ell, \ldots, \mathcal{X}_{2^{\ell-1}}^\ell\right)$. Let us assume this process proceeds for k levels. Let (ℓ, s) be the node associated to the sth subset at resolution level ℓ. Let $de(\ell, s)$ and $an(\ell, s)$ be respectively the set of descendants and ancestors of node (ℓ, s). Let $A_\ell(x) \in \{1, \ldots, 2^{\ell-1}\}$ be the subset at level ℓ including predictor x, with $A_1(x)$ equal to 1 by definition.

We characterize the conditional density $f(y|x)$ as a convex combination of multiscale dictionary densities. At level one, the global parent density is denoted by f_1. The dictionary density at generation j is f_{B_j} with $B_j = \{j, A_j\}$, for $j = 2, \ldots, k$. Then, $f(y|x)$ is defined as the convex combination of densities $\{f_{B_j(x)}\}_{j=1}^k$ with weights $\{\pi_{B_j(x)}\}_{j=1}^k$, i.e.

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

where $0 \leq \pi_{B_j(x)}$ and $\sum_{j=1}^k \pi_{B_j(x)} = 1$.

Each $B(x)$ is a set encoding the path through the partition tree up to generation k specific to predictor value x. According to model (1), one observation can lie in subsets located at different resolution levels. This is particularly useful to reach a good compromise between bias and variance and borrow information across different resolution levels. Though the proposed approach is reminiscent of a mixture of experts model (Jacobs et al., 1991), the two approaches are quite complementary, since under (1), neither mixture weights nor dictionary densities directly depend on predictors. This allows our model to scale efficiently to high dimensional predictors.

Now let us examine the implications of model (1). For two predictor values x and x' located close together, it is expected that the paths will be similar, which leads to similar weights on the dictionary densities. In the extreme case in which x and x' belong to the same leaf partition set, we have $B(x) = B(x')$ and the path through the tree will be the same. Moreover, in this case, we will have $f(y|x) = f(y|x')$ so that up to k levels of resolution the densities $f(y|x)$ and $f(y|x')$ are identical. If the paths through the tree differ only in the final generation or two, the weights will typically be similar but the resulting conditional densities will not be identical.

To derive mixture weights, a natural choice corresponds to a stick-breaking process (Sethuraman, 1994). For each node $B_j(x_i)$ in the binary partition tree, define a stick length $V\{B_j(x_i)\} \sim \text{Beta}(1, \alpha)$. The parameter α encodes the complexity of the model, with $\alpha = 0$ corresponding to the case in which $f(y|x) = f(y)$. We relate the weights in (1) to the stick-breaking random variables as follows:

$$\pi_{B_j(x)} = V\left\{B_j(x)\right\} \prod_{B_h \in an\{B_j\}} \left[1 - V\left\{B_h(x)\right\}\right],$$

with $V\{B_k(x)\} = 1$ to ensure that $\sum_{j=1}^k \pi_{B_j}(x) = 1$.

4. Estimation

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 (Karypis & Kumar, 1999), a fast multiscale technique used for graph partitioning. Basically, the graph is obtained adding an edge between each pair of data points, i.e. (y_i, y_j) with $i \neq j$, and assigning to any such edge the weight $w_{ij} = \exp\{-d(x_i, x_j)\}$ with $d(\cdot, \cdot)$ being some metric. Though more complicated densities can be considered, dictionary densities f_{B_j} will be estimated by assuming a normal form, i.e. $f_{B_j} = \mathcal{N}(\mu_{B_j}, \sigma_{B_j})$. In particular, densities corresponding to a particular partition set will be estimated considering only observations belonging to that partition set. To be specific, for estimating density $f_{B_j}(y)$, we use the data $\{y_i : x_i \in \mathcal{X}_{A_j}^j\}$. We then conduct the

analysis treating partition sets as fixed.

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. For this reason, parameters involved in dictionary densities will be estimated through bayesian methods and inference on stick breaking weights and dictionary densities parameters will be carried out using the Gibbs sampler. For this purpose, introduce the latent variable $S_i \in \{1, \ldots, k\}$, for $i = 1, \ldots, n$, denoting the multiscale level used by the ith subject. Define priors placed on model parameters as follows $\mu \sim \mathcal{N}(0, I)$, $\sigma = \mathcal{IG}(a, b)$ and $V_{B_j} \sim \text{Beta}(1, \alpha)$. Let n_{B_j} be the number of observations allocated to node B_j . Each Gibbs sampler iteration can be summarized in the following steps

1. Update S_i by sampling from the multinomial full conditional with

$$\Pr(S_i = j \mid -) = \frac{\pi_{B_j(x_i)} f_{B_j(x_i)}(y_i)}{\sum_{h=1}^{k} \pi_{B_h(x_i)} f_{B_h(x_i)}(y_i)}$$

2. Update stick-breaking random variable $V_{B_j(x_i)}$, for $j = 1, \ldots, k$ and $i = 1, \ldots, n$, from $\text{Beta}(a_p, b_p)$ with $a_p = 1 + n_{B_j}$ and $b_p = \alpha + \sum_{B_h(x_i) \in de\{B_j(x_i)\}} n_{B_h(x_i)}$.

3. Update $(\mu_{B_j}, \sigma_{B_j})$ by sampling from

$$\mu_{B_j} \sim \mathcal{N}\left(\bar{y}_{B_j} n_{B_j}/\sigma_{B_j}, I(1 + n_{B_j}/\sigma_{B_j})\right)$$

$$\sigma_{B_j} \sim \mathcal{IG}\left(a + n_{B_j}/2, b + 0.5 \sum_{\{h: S_h = j\}} \left(y_h - \mu_{B_j}\right)^2\right)$$

with $\bar{y}_{B_j}$ being the average of observation allocated to node B_j.

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, $y \in \mathbb{R}$ the response variable and $x \in \mathbb{R}^p$ 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 (Chauveau & Diebolt, 1998). 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.

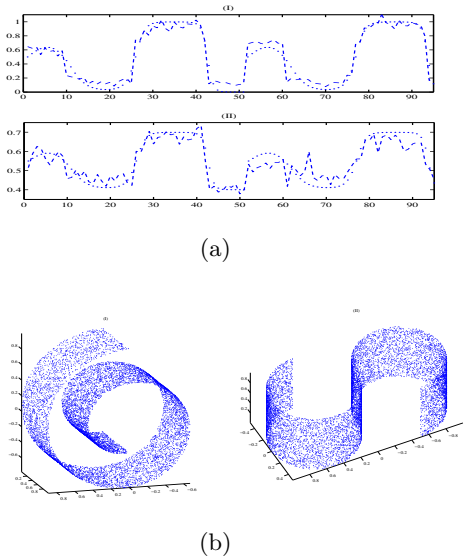

Figure 1. (a) Plot of mean and variance (I-II) for observations $i = 1, \ldots, 95$ (dot:true, dash:estimate); (b) Toy data examples: Swissroll (I) and S-Manifold (II) embedded in $\mathbb{R}^3$

First let us consider the data example in §2. Figure 1(a) depicts the true mean and variance of y and our estimate as x moves along the geodesic. These estimates were obtained by performing leave-one-out prediction and considering the mean and variance of the predictive distribution of y_i as the mean and variance estimate of the ith observation. As the figure clearly shows our construction facilitates a smooth estimate of the mean and variance of y, even though we are not explicitly smoothing, rather, the smoothness is induced via the model averaging over spatial scales.

In all other examples, 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. In the first three simulation studies, the vector of predictors was assumed to lie close to a lower dimensional plane. In practice, predictors were modeled through a factor model, i.e. $x_i = \Lambda \eta_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}_n(0, I_n)$, Λ being a $(p \times r)$ matrix, $\eta_i \sim \mathcal{N}_r(0, I_r)$ and $r << p$. 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 all examples, Λ was assumed to be a sparse matrix with level of sparsity increasing with the number of columns and non zero elements of Λ drawn from a standard normal density. In the last two simulation studies, predictors were assumed to lie close to the swissroll and the S-

Table 1. Mean and standard deviations of squared errors under multiscale stick-breaking (MSB), CART and Lasso for sample size 50 and 100 for different simulation scenarios

SIM	p	r	MSB	CART	LASSO
(1)	1,000	5	1.09	2.29	1.09
			(1.68)	(2.82)	(1.66)
(2)	10,000	5	0.55	0.55	0.99
			(0.86)	(0.62)	(0.79)
(3)	5,000	5	0.78	0.83	0.84
			(1.99)	(2.16)	(2.00)
(4)	50	2	0.80	1.00	1.01
			(0.82)	(1.36)	(1.04)
(5)	50	2	0.60	0.64	1.01
			(0.76)	(0.84)	(1.16)

manifold (see figure 1(b)).

In the first simulation study, the response was assumed to be a linear function of predictors and they were jointly sampled from the above factor model. In the second simulation study, the response was drawn from a two components mixture of normals with mixture weights depending on the first latent factor, i.e. $p = \exp\{\eta_1\}/(1+\exp(\eta_1)\}$, and components with location parameters $(-2, 2)$ and unitary standard deviation. In the third simulation study, the response was drawn from a normal with mean and variance depending on the first latent factor as follows $y \sim \mathcal{N}\{\eta_1^2 - \eta_1^3, \exp(1-\eta_1)\}$. In the last two simulation studies, predictors were drawn from the swissroll and the S-manifold, all two-dimensional manifolds but embedded in $\mathbb{R}^{50}$, while the response was sampled from a normal with mean equal to one of the coordinates of the manifold and standard deviation one.

Table 2 shows mean squared errors under the proposed approach, CART and lasso based on leave-one-out prediction. In particular, for each resolution level, the new observation was allocated to the set with closer center. As shown in table 2, CART performs worse than lasso only when the response is a linear function of predictors. However, in all data scenarios, our model is able to perform as well as or better than the model associated to the lowest mean squared error. Figure 2 shows the plot of CPU usage as a function of the number of features. This plot was obtained drawing (y_i, x_i), for $i = 1, \ldots, 100$, and $x_i \in \mathbb{R}^p$ from the first simulation scenario considering different values of p. Clearly, our approach scales substantially better than competitors to massive number of features.

Another important advantage of the proposed model is the possibility to obtain an estimate of the predictive density of the data. Figure 3 shows the estimated density of two data points sampled from the second simulation scenario. Clearly, the density function varies across different points and its estimate become closer to the true density as the number of observations in the training set increases.

Table 2. Real Data: Mean and standard deviations of squared error under multiscale stick-breaking (MSB), CART, Lasso and random forest (RF)

DATA	MODEL	MSE	t_T	t_M	t_V
(1)	MSB	0.56	100	1.1	0.02
	CART	1.10	87	0.9	0.01
	LASSO	0.63	200	2.8	0.17
	RF	0.57	7,817	78.2	0.59
(2)	MSB	0.76	690	20.98	2.31
	LASSO	1.02	5,836	96.18	9.66

6. Real Application

The first dataset consists of a measurement of creativity observed for 108 subjects. We would like to predict the value of creativity based on the number of connections between different cortical regions of the brain. Therefore, for each subject, a brain graph involving 70 cortical regions was observed. The vector of covariates consists in the logarithm of the total number of connections between all pairs of cortical regions, i.e. $p = 2,415$. The second dataset consists of a measure of head motion observed for 56 subjects. As a measure of head motion we considered the mean framewise displacement (FD) computed as described in (Power et al., 2012). Our interest was predicting the head motion measurement based on 3D brain images involving about one million of pixels. These 3D matrices were vectorized and considered as predictors in our model. In order to reduce the dimensionality of the predictor space the data was re-processed using a brain mask and a vector of about 300,000 predictors was obtained.

For the analysis all variables have been normalized by subtracting the mean and dividing by the variance. The same prior specification and Gibbs sampler as in §5 was utilized. Table 2 shows mean and variance squared error based on leave-one-out predictions. Variable t_T is the amount of time necessary to obtain predictions for all subjects, while variables t_M and t_V are respectively the mean and the variance of amount of time necessary to obtain one point predictions.

For the first data example, our model was compared

to CART, lasso and random forest. As shown in table 2, our approach performs better than random forest in terms of mean squared error and is associated to a much lower CPU time. This real data application does not involve a huge number of predictors so that computationally our model performs almost as well as lasso and CART. However, as shown in section 5, our model can scale substantially better than all other models to huge number of features.

For the second data application, given the huge dimensionality of the predictor space and the poor scalability of CART and random forest, the comparison was made only with lasso. As shown in table 2, our approach is more efficient and accurate than lasso in predicting the response variable. Figure 4 shows the plot of CPU time used to predict each one of the 56 points involved in the experiment. The time needed to compute quantities utilized in all points predictions was divided equally across subjects. Clearly, our approach is able to improve the computational time by up to five orders of magnitude.

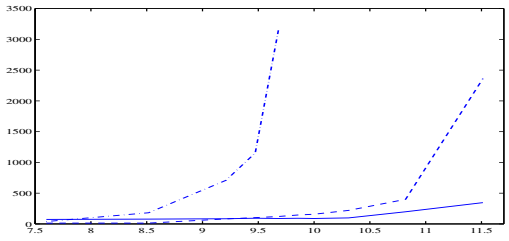

Figure 2. Elapsed CPU time (in seconds) for leave-one-out prediction based on 100 observations for MSB (solid), lasso (dash) and CART (dot-dash) for different number of predictors in log-scale

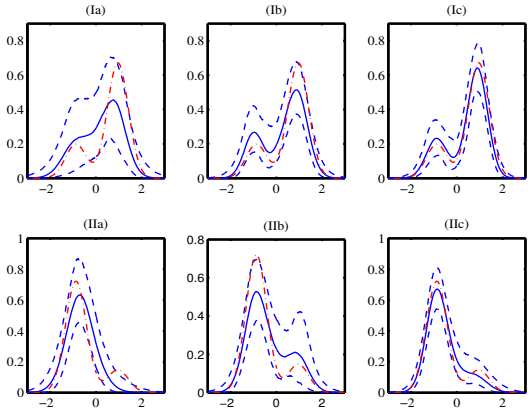

Figure 3. Plot of true (dashed-dotted line) and estimated (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) density for two data points (I, II) considering different training set size (a:50, b:100, c:150).

Discussions

We have proposed a new model which should lead to substantially improved predictive and computational performance to learn the density of a target variable given a high dimensional vector of predictors. As shown the proposed two stage approach can scale substantially better than other existing algorithms to massive number of features. We have focused on Bayesian MCMC-based methods, but there are numerous interesting directions for ongoing research.

References

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

Breiman, L. Bagging predictors. *Machine Learning*,

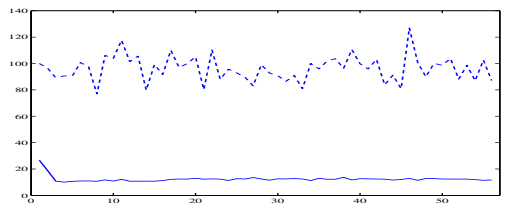

Figure 4. Plot of CPU time used to predict each one of the 56 measurement involved in experiment (2) under MSB (solid) and lasso (dash)

24:123140, 1996.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Tran, M. N., Nott, D. J., and Kohn, R. Simultaneous variable selection and component selection for

regression density estimation with mixtures of heteroscedastic experts. *Electronic Journal of Statistics*, 6:1170–1199, 2012.