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 an important application to predicting drug response to anti-depressants.

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

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

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 well. 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 ultrahigh-dimensional 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 (**?**).

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 a machine diagnosis for a patient presenting with a number of complicated psychiatric symptoms. The challenge would then be to estimate the probability that the patient fits any of the diagnostic criteria for some category of mental illness via a $\mathcal{O}(10^6)$ dimensional image of the subject's brain.

2. Setting

Let $\mathbf{X}\colon \Omega \to \mathcal{X} \subseteq \mathbb{R}^D$ be a D-dimensional Euclidean vector-valued *predictor* random variable. Let F_X denote the *marginal* probability density of $\mathbf{X}$, and f_X be the probability that $\mathbf{X} = X \in \mathcal{X}$. We assume that F_X concentrates around a lower-dimensional (possibly nonlinear) subspace $\mathcal{M} = \{\mu \in \mathcal{M}\}$. For example, $\mathcal{M}$ could be a union of affine subspaces, or a smooth compact Riemannian manifold, etc.

Let $\mathbf{Y}\colon \Omega \to \mathcal{Y} \subseteq \mathbb{R}$ be a real-valued *target* random variable. We further assume that *conditional* distribution is a function of only the position of X along the manifold, $F_{Y|X} = F_{Y|\mu}$. Let X and Y be sampled from some true but unknown joint distribution

$(X, Y) \sim F_{X,Y}$. Given a realization X of predictors, we would like to know $F_{Y|\mathbf{X}=X}$.

We obtain $\hat{F}_{Y|X}$—an estimate of this conditional density—via a *point cloud*. Specifically, we assume that we obtain n independently and identically sampled observations, $(X_i, Y_i) \overset{iid}{\sim} F_{X,Y}$, for $i \in [n] = \{1, 2, \ldots, n\}$.

3. A Simple Illustrative Example

Factorize the joint distribution $F_{X,Y} = F_X F_{Y|X}$. Let X live on some smooth one-dimensional Riemannian submanifold embedded in $\mathbb{R}^D$. Let Y be a univariate Gaussian random variable whose mean and variance vary with the location of X along its geodesic. See the left panel of Figure **??** for a graphical depiction. The color of the line indicates the position of X along its geodesic, as well as the mean and variance of Y. We can formalize this model as follows. Define the marginal $F_X = \mathcal{N}(\psi(\mu_i), \sigma^2 \boldsymbol{I}_D)$, where $\Psi = \{\psi \colon \mathcal{M} \to \mathbb{R}^D\}$, $\mu_i \in \mathcal{M}$, $\sigma \in \mathbb{R}$, $\mathcal{I}_D$ is the $D \times D$ dimensional identity matrix, and $\mathcal{N}(\cdot, \cdot)$ indicates a Gaussian distribution. Let $\mathcal{M}$ be a smooth compact Riemannian manifold, such as the oscillating D-wave or the swissroll. Let $\psi(\mu) = \mathbf{1}_D \mu$. Define the conditional $F_{Y|X=x} = \mathcal{N}(\mu_x, \mu_x^2)$. In other words, both the mean and standard deviation of Y are equal to the position of X along its geodesic. The middle panel of Figure **??** depicts our estimate of the mean and variance of y as x moves along the geodesic. The right panel depicts our estimate of the mean and variance of y along the best one-dimensional linear projection of x. Thus, it should be clear that our estimate of the distribution of Y conditional on X is a highly nonlinear function of X, even though it is a simple function of $\mathcal{M}$. Moreover, our construction facilitates a smooth estimate of the manifold, even though we are not explicitly smoothing, rather, the smoothness is induced via the model averaging over spatial scales.

4. 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 (which lives in a high-dimensional ambient space). We pursue a two-stage strategy. In the first stage, we try to find a low-dimensional representation of the predictors via a multiscale nonlinear partitioning of the data. In other words, we recursive partition $\{X_i\} = \{X_i\}_{i \in [n]}$ to obtain subsets of $\{X_i\}$ 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 (see Figure **??**). In the second stage, we estimate the conditional distribution of the target variable as a function of the multiscale embedding of the predictor

In all these applications, common models utilized for density estimation, classification, variable selection and predictions fail to be efficient and cannot be applied. Dealing with large amounts of data requires the introduction of new models able to process the data accurately and efficiently. In this paper, we will focus on conditional density estimation for massive datasets. Conditional density estimation aims to estimates the density of the response $y \in \mathcal{Y}$ given a set of predictors $(x_1, x_2, \ldots, x_p) \in \mathcal{X}$. Though, a variety of flexible models have been proposed in the last decade, density estimation remains challenging for large sample sizes and high dimensional predictors.

The need to deal with a large number of observations motivated the literature on divide-and-conquer techniques, a class of algorithms extensively used in density estimation, classification and prediction. Well known examples are classification and regression trees (CART) (Breiman et al., 1984) and multivariate adaptive regression trees (MARS) (**?**). 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 to reduce the dimensionality of the problem, single tree estimates are generally associated to high variance. A possible solution to this problem would be combining estimates resulting from different trees. Well known examples are bagging (Breiman, 1996), boosting (Shapire et al., 1998) and random forest (Breiman, 2001). Though these algorithms can substantially reduce the variance, they can be computationally intensive.

Mixture of experts (Jacobs et al., 1991) is another divide-and-conquer algorithm particularly useful to reduce the variance associated to single tree estimates. As opposed to other divide-and-conquer algorithms, mixture of experts rely on soft partitioning algorithms that allows observations to lie simultaneously in different subsets. A mixture of experts model is a mixture model in which the model parameters, including mixture weights, are functions of covariates. Several mixture of experts models have been proposed in the last twenty years. some of them gain flexibility by dealing with infinitely many experts (Rasmussen & Ghahramani, 2002) (Meeds & Osindero, 2006), others propose a hierarchical structure where a mixture model is fit in each subset (Jordan & Jacobs, 1994) (Bishop &

Svensen, 2003).

A significant downside of all 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 density are predictor dependent. In an attempt to make mixture 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 still a challenging problem, especially when multiple parameters involved in the model, such as weights and mean functions, depend on high dimensional predictors.

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 multiscale partition tree for the features. The proposed approach is based on a two-stage algorithm where first the predictor space is recursively partitioned and then, considering the partition as fixed, a multiresolution stick-breaking process is estimated. According to the proposed process, observations can lie simultaneously in subsets located at different resolution levels. This results in a model that allows borrowing information across different resolution levels and reaches a good compromise in terms of the bias and variance trade-off. The tree partition is found by implementing a fast multiscale technique used for graph partitioning (Karypis & Kumar, 1999). We show that the algorithm scales efficiently to massive numbers of features, and can be implemented efficiently with gibbs sampling. State of the art predictive performance is demonstrated for toy examples and an important application to predicting drug response to anti-depressants.

5. Model Specification

5.1. Model Structure

This paper will mainly focus on the problem of estimating the conditional density $f(y|x)$ of the response given a high dimensional vector of predictors, concentrated near a lower-dimensional subspace $\mathcal{M}$ embedded in $\mathcal{X}$. In dealing with massive datasets, the idea of combining different local models defined on subsets of the predictor space have increased popularity in the last decade. These methods, generally known as divide-and-conquer algorithms, aim to replace complicated conditional density functions with a combination of simple densities defined locally, on subsets of the predictor space.

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 $ch(\ell, s)$ and $pa(\ell, s)$ be respectively the set of children and parents of node (ℓ, s). Let $A_\ell(x) \in \{1, \ldots, 2^{\ell-1}\}$ be the location of predictor x at level ℓ, 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 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 combinations of densities $\{f_{B_j}\}_{j=1}^k$ 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 $0 \leq \pi_{B_j(x)} \leq 1$ 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 simultaneously lie in subsets located at different resolution levels. This is particularly useful to reach a good compromise between the bias and variance trade-off as explained in the next section. Moreover, it allows borrowing information across different resolution levels. Though the proposed approach reminds a mixture of experts model (Jacobs et al., 1991), the two approaches are completely 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.

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:

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

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

5.2. Theoretical properties

6. Estimation

The proposed approach is based on a two-stage algorithm where first the predictor space is recursively divided in different subsets using an efficient partitioning algorithm and then, considering the partition as fixed, a multiresolution stick-breaking process is estimated. The predictor space is partitioned applying metis (Karypis & Kumar, 1999), a fast multiscale technique used for graph partitioning. Though more complicated densities can be considered, dictionary densities f_{B_j} will be estimated by assuming a normal form. In particular, densities corresponding to a particular partition set will be estimated considering observations for all subjects having predictors in that partition set. To be specific, for estimating density $f_{B_j}(y)$, we use the data $\{y_i : x_i \in \mathcal{X}^j_{A_j}\}$. We then conduct the analysis treating the dictionary elements and partition sets as fixed and placing a prior on the weights π_{B_j}.

Parameters involved in the dictionary density can be estimated using either frequentist or bayesian methods. Both methodologies have advantages and disadvantages. The frequentist approach, relying on maximum likelihood estimation, would allow to obtain parameter estimates in a faster and more efficient way. On the other hand, bayesian methods can avoid singularities associated with traditional maximum likelihood inference. Both methodologies will be implemented and compared for all data examples considered.

In any case, inference on stick breaking weights 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. Let $n(B_j)$ be the number of observations allocated to node B_j. Then, at each gibbs sampling iteration, stick breaking are sampled as follows

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

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

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

7. 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 the response variable and $x \in \mathcal{R}^p$ a set of predictors. In order to sample stick breaking weights, 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 (**?**).

In all examples below, predictors were assumed to belong to a lower dimensional manifold. In the first three simulation studies, the manifold was assumed to be a lower dimensional plane. In particular, the vector of predictors was modeled through a factor model, i.e. $x_i = \Lambda\eta_i + \epsilon_i$ with $\epsilon_i \sim N(0, I)$, Λ being a $(p \times k)$ matrix, η_i a k dimensional vector with elements drawn from a standard nromal and $k << p$. 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 fourth simulation study, predictors were drawn from the swiss roll manifold.

In the first simulation study, (k, p) were chosen to be $(5, 1000)$ and response and predictors were jointly sampled from the above factor model. In the second simulation study, (k, p) were chosen to be $(5, 10000)$ and the response was drawn from a $N(2, 1)$ with probability $p = \exp\{\eta_1\}/(1 + \exp(\eta_1))\}$ and from a $N(-2, 1)$ with probability $1 - p$. In the third simulation study, (k, p) were chosen to be $(5, 5000)$ and the response was drawn from a normal with mean and

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

SIM	MODEL	MSE		VAR	
n		50	100	50	100
(1)	MSB	1.02	1.09	1.26	1.68
	CART	2.69	2.29	3.00	2.82
	LASSO	1.02	1.09	1.25	1.66
(2)	MSB	0.56	0.55	0.66	0.86
	CART	0.67	0.55	1.09	0.62
	LASSO	0.84	0.99	0.79	0.79
(3)	MSB	0.89	0.78	2.34	1.99
	CART	1.25	0.83	1.78	2.16
	LASSO	1.00	0.84	2.41	2.00
(4)	MSB	0.89	0.71	3.30	4.20
	CART	1.77	1.24	5.23	4.63
	LASSO	1.38	0.79	3.29	4.49

variance depending on the first latent factor as follows $y \sim N\{\eta_1^2 - \eta_1^3, \exp(1-\eta_1)\}$.

Table 1 shows mean squared errors based on leave-one-out predictions under multiscale stick breaking, CART and lasso. In the first example, where a linear relationship is assumed between response and predictors, lasso performs better than CART. Instead, our approach is able to perform well in both linear and non linear case. Figure 2 [coming soon] shows a plot of CPU usage under all three approaches as the number of predictors increases. Clearly, our multiscale stick breaking scales substantially better to massive number of features.

It is important to note that under the proposed approach an estimate of the predictive densities of the data can be obtained. Figure **??** shows the estimated density of two observations sampled from the second example. Clearly, as the number of observations in the training set increases, the two densities become closer to the true one. [will provide 95% intervals, need to rerun codes]

8. Real data examples

We assessed the predictive performance of the proposed method on a real dataset. This 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. For each subject, we observe a brain graph consisting of 70 cortical regions whose centers are depicted as vertices. Therefore, the

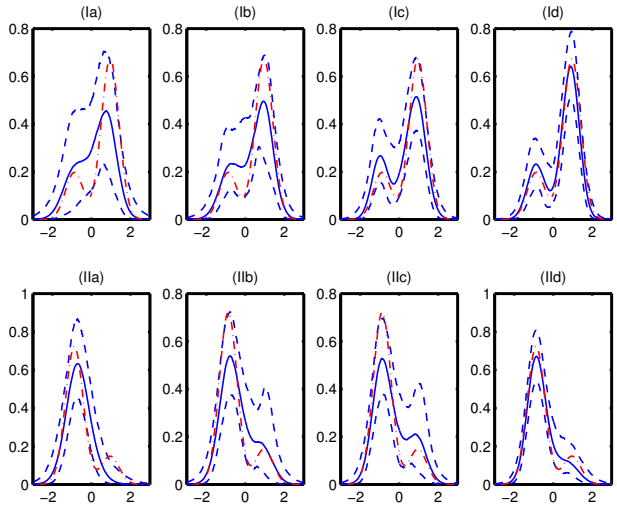

Figure 1. Plot of true density (dashed-dotted line) and estimated density (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) for two observations (I, II) considering different training set size (a:50, b:100, c:150, d:200).

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

MODEL	MSE	VAR	t_T	t_M	t_V
MSB	1.00	1.42	100	1.1	0.02
CART	1.99	3.23	87	0.9	0.01
LASSO	1.10	1.43	200	2.8	0.17
RF	1.09	1.54	7,817	78.2	0.59

brain graph involves $4,900$ vertices and $2,415$ different pairs of vertices. The vector of covariates consists in the logarithm of the total number of connections between all pairs of vertices. Both response and covariates have been normalized by subtracting the mean and dividing by the variance. The same Gibbs sampler as in section 4 was utilized.

Table 3 shows mean and variance squared error based on leave-one-out predictions for CART, random forest, lasso and our approach. Variable t_T is the amount of time necessary to obtain predictions for all subjects, while variables t_M and t_V are the mean and variance of time necessary to obtain prediction for one subject. Though CART is associated to a lower CPU time, it performs worse than all other approaches in terms of mean squared error.

References

Bishop, C.M. and Svensen, M. Bayesian Hierarchical mixtures of experts. *Nineteenth Conference on Uncertainty in Artificial intelligence*, pp. 57–64, 2003.

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

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

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

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.

Meeds, E. and Osindero, S. Bayesian Hierarchical mixtures of experts. *Advances in Neural Information Processing Systems*, 2006.

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

Rasmussen, C. E. and Ghahramani, Z. Infinite mixtures of Gaussian process experts. *Advances in neural information processing systems 14*, 2002.

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.