Compressed Gaussian Process Manifold Regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Nonparametric regression for massive numbers of samples (n) and features (p) is an important problem. We propose a Bayesian approach for scaling up Gaussian process (GP) regression to big n and p settings using random compression. The proposed compressed GP is particularly motivated by the setting in which features can be projected to a low-dimensional manifold with minimal loss of information about the response. Conditionally on a random compression matrix and a smoothness parameter, the posterior and posterior predictive distributions are available analytically. Running the analysis in parallel for many random compression matrices and smoothness parameters, model averaging is used to combine the results. The algorithm can be implemented rapidly even in very big n and p problems, has strong theoretical justification, and is found to yield state of the art predictive performance.

1 Introduction

Data are routinely collected containing massive numbers of features, ranging from thousands to millions or more. Nonparametric regression models are appealing in accommodating complex nonlinear relationships between the features and response, with a typical model having the general form:

$$y = \mu_0(\boldsymbol{x}) + \epsilon, \;\; \epsilon \sim N(0, \sigma^2),$$

where $\boldsymbol{x} \in \mathcal{R}^p$, $\mu_0(\cdot)$ is the unknown regression function and ϵ is a residual. When p is not small, estimating μ_0 can lead to a statistical and computational curse of dimensionality. One strategy for combating this curse is dimensionality reduction via variable selection or (more broadly) subspace learning, with the high-dimensional features replaced with their projection to a d-dimensional subspace or manifold with $d \ll p$. In many applications, the relevant information about the high-dimensional features can be encoded in such low dimensional coordinates.

There is a rich literature on subspace learning for regression, typically employing a two stage approach. However, as the sample size increases, eigen-decomposition of an $n \times n$ matrix becomes computationally burdensome, so that the two stage approaches involving isomap ([1]) and/or Laplacian eigenmaps ([2,3]) are challenging computationally. To free this bottleneck, recently ([4]) employs a column sampling algorithm that only requires eigen-decomposition of an $m \times m$ matrix without losing much accuracy, even with $m \ll n$. Once lower dimensional features are obtained, the second stage uses these features in standard regression and classification procedures as if they were observed initially. Such two stage approaches rely on learning the manifold structure embedded in the high dimensional features, which adds unnecessary computational burden when inferential interest lies mainly in prediction.

An alternative strategy focuses on divide-and-conquer techniques. As the number of features increases, the problem of finding the best splitting attribute becomes intractable, so that CART ([5]), MARS and multiple tree models, such as Random Forest ([6]) cannot be efficiently applied. A much simpler approach is to apply high dimensional clustering techniques, such as metis, cover trees and

spectral clustering. Once the observations are clustered into a few groups, simple models (glm, Lasso etc) are fitted in each cluster. Although often excellent at point predictions, such methods can be sensitive to tuning parameters, do not characterize predictive uncertainty, and may lack efficiency outside of the $n \gg p$ setting in treating subsets independently. There is also a recent literature on scaling up sparse optimization methods, such as Lasso, to huge n and p settings relying on algorithms that can exploit multiple processors in a distributed manner ([7]). However, such methods are yet to be developed for non-linear manifold regression, which is the central focus of this article.

This naturally motivates Bayesian models that simultaneously learn the mapping to the lower-dimensional subspace along with the regression function in the coordinates on this subspace, providing a characterization of predictive uncertainties. There is a small literature on relevant Bayesian methods that accommodate non-linear subspaces, ranging from Gaussian process latent variable models (GP-LVMs) ([8]) for probabilistic nonlinear PCA to mixture factor models ([9]). However, for large n, p, there is a heavy computational price for learning the number and the distribution of the latent variables, and the mapping functions while maintaining identifiability restrictions. In general, current Bayesian nonparametric regression methods that provide a realistic characteristic of predictive uncertainty face bottlenecks in scaling to large n and p.

Recently, ([10]) show that when the features lie on a d-dimensional manifold embedded in $\mathcal{R}^p$ with $d \ll p$ and the regression function not highly smooth, the optimal performance can be obtained using GP regression with a squared exponential covariance in the original high-dimensional feature space. This is an exciting theoretical result, which provides motivation for the approach in this article, which is focused on scalable Bayesian nonparametric regression in large p and n settings. For broader applicability than ([10]), we accommodate features that are contaminated by noise and hence do not lie exactly on a low-dimensional manifold. In addition, we facilitate scaling in both p and n by bypassing MCMC and reducing matrix inversion bottlenecks via random projections. Sensitivity to the random projection and to tuning parameters is eliminated through the use of Bayesian model averaging. To our knowledge, no Bayesian manifold regression technique has yet been developed that can scale up for large sample size and massive number of features yielding accurate predictive inference rapidly.

Section 2 proposes the model and computational approach in large p settings. Section 3 describes extensions to large n, and section 4 develops theoretical justification. Section 5 contains simulation and real data examples comparing with alternative approaches. Section 6 concludes the paper with a discussion.

2 Compressed Gaussian process regression

2.1 Model

For examples $i = 1, \ldots, n$, let $y_i \in \mathcal{Y}$ denote a continuous response with associated features (lying on a noise corrupted manifold) $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})' = (z_{i1}, \ldots, z_{ip})' + (\delta_{i1}, \ldots, \delta_{ip})' = \boldsymbol{z}_i + \boldsymbol{\delta}_i$, $\boldsymbol{z}_i \in \mathcal{M}$, $\boldsymbol{\delta}_i \in \mathcal{R}^p$, where $\mathcal{M}$ is a d-dimensional manifold embedded in the ambient space $\mathcal{R}^p$. We assume a compressed nonparametric regression model

$$y_i = \mu\big(\boldsymbol{\Psi}\boldsymbol{x}_i\big) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \tag{1}$$

with the residuals modeled as Gaussian with variance σ^2, though other distributions including heavy-tailed ones can be accommodated. $\boldsymbol{\Psi}$ is an $m \times p$ matrix that compresses p-dimensional features to dimension m. Following a Bayesian approach, we choose a prior distribution for the regression function μ and residual variance σ^2, while randomly generating $\boldsymbol{\Psi}$ following precedence in the literature on feature compression ([11,12,13]). These earlier approaches differ from ours in focusing on parametric regression. We independently draw elements $\{\Psi_{ij}\}$ of $\boldsymbol{\Psi}$ from $N(0, 1)$, and then normalize the rows using Gram-Schmidt orthogonalization.

We assume that $\mu \in \mathcal{H}_s$ is a continuous function belonging to $\mathcal{H}_s$, a Holder class with smoothness s. To allow μ to be unknown, we use a Gaussian process (GP) prior, $\mu \sim \text{GP}(0, \sigma^2\kappa)$ with the covariance function chosen to be squared exponential $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j; \lambda) = \exp\big(-\lambda||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\big)$, with λ a smoothness parameter and $||\cdot||^2$ the Euclidean norm. To additionally allow the residual variance σ^2 and smoothness λ to be unknown, we let $\sigma^2 \sim IG(a, b), \quad \lambda^d \sim Ga(a_0, b_0)$. ([10]) shows that

a GP prior with such powered gamma prior on the smoothness can achieve the minimax optimal adaptive rate when $\boldsymbol{x}_i \in \mathcal{M}$.

In many applications, features may not lie exactly on $\mathcal{M}$ due to noise and corruption in the data. We apply random compression in (1) to de-noise the features, obtaining $\boldsymbol{\Psi}\boldsymbol{x}_i$ much more concentrated near a lower-dimensional subspace than the original $\boldsymbol{x}_i$. In addition to de-noising, this approach has the major advantage of bypassing estimation of a geodesic distance along the unknown manifold $\mathcal{M}$ between any two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$.

Let $\boldsymbol{\mu} = (\mu(\boldsymbol{\Psi}\boldsymbol{x}_1), ..., \mu(\boldsymbol{\Psi}\boldsymbol{x}_n))'$ and $\boldsymbol{K}_1 = (\kappa(\boldsymbol{\Psi}\boldsymbol{x}_i, \boldsymbol{\Psi}\boldsymbol{x}_j; \lambda))_{i,j=1}^n$, $b_1 = \boldsymbol{y}' (\boldsymbol{K}_1 + \boldsymbol{I})^{-1} \boldsymbol{y}/2$. With the prior on $\boldsymbol{\mu}, \sigma^2$ as above, the predictive of $\boldsymbol{y}^* = (y_1^*, ..., y_{n_{pred}}^*)'$ given $\boldsymbol{X}^* = \left(\boldsymbol{x}_1^{*'}, ..., \boldsymbol{x}_{n_{pred}}^{*'}\right)'$ and $\boldsymbol{\Psi}, \lambda$ for new n_{pred} subjects marginalizing out $(\boldsymbol{\mu}, \sigma^2)$ over their posterior distribution is available analytically as

$$\boldsymbol{y}^* | \boldsymbol{x}_1^*, ..., \boldsymbol{x}_{n_{pred}}^*, \boldsymbol{y} \sim t_n \left(\mu_{pred}, \sigma_{pred}^2\right), \tag{2}$$

where $\boldsymbol{K}_{pred} = \{\kappa(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*; \lambda)\}_{i,j=1}^{n_{pred}}$, $\boldsymbol{K}_{pred,1} = \{\kappa(\boldsymbol{x}_i^*, \boldsymbol{x}_j; \lambda)\}_{i=1,j=1}^{i=n_{pred},j=n}$, $\boldsymbol{K}_{1,pred} = \boldsymbol{K}'_{pred,1}$, $\mu_{pred} = \boldsymbol{K}_{pred,1} (\boldsymbol{I} + \boldsymbol{K}_1)^{-1} \boldsymbol{y}$, $\sigma_{pred}^2 = (2b_1/n) \left[\boldsymbol{I} + \boldsymbol{K}_{pred} - \boldsymbol{K}_{pred,1} \{\boldsymbol{I} + \boldsymbol{K}_1\}^{-1} \boldsymbol{K}_{1,pred}\right]$.

2.2 Model averaging

To accomplish robustness with respect to the choice of $(\boldsymbol{\Psi}, \lambda)$ and the subspace dimension m, following ([13]), we propose to generate s random matrices having different m and s different λ from $Unif(3/d_{max}, 3/d_{min})$, $(\Psi^{(l)}, \lambda^{(l)})$, and then use model averaging to combine the results. We choose $d_{max} = \max_{i,j=1,..,n} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2$, $d_{min} = \min_{i,j=1,..,n} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2$. To make matters more clear, let $\mathcal{M}_l$, $l = 1, \ldots, s$, represent (1) with m_l number of rows. Corresponding to the model $\mathcal{M}_l$, we denote $\boldsymbol{\Psi}$, λ, $\boldsymbol{\mu}$ and σ^2 by $\boldsymbol{\Psi}^{(l)}$, $\lambda^{(l)}$, $\boldsymbol{\mu}^{(l)}$ and $\sigma^{2(l)}$ respectively. Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_s\}$ denote the set of models corresponding to different random projections, $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$ denote the observed data. Then, the predictive density of $\boldsymbol{y}^*$ given $\boldsymbol{X}^*$ is $f(\boldsymbol{y}^* | \boldsymbol{X}^*, \mathcal{D}) = \sum_{l=1}^s f(\boldsymbol{y}^* | \boldsymbol{X}^*, \mathcal{M}_l, \mathcal{D}) P(\mathcal{M}_l \,|\, \mathcal{D})$, where the predictive density of $\boldsymbol{y}^*$ given $\boldsymbol{X}^*$ under projection $\mathcal{M}_l$ is given in (2) and the posterior probability weight on projection $\mathcal{M}_l$ is $P(\mathcal{M}_l \,|\, \mathcal{D}) = \frac{P(\mathcal{D} \,|\, \mathcal{M}_l) P(\mathcal{M}_l)}{\sum_{h=1}^s P(\mathcal{D} \,|\, \mathcal{M}_h) P(\mathcal{M}_h)}$. Assume equal prior weights for each random projection, $P(\mathcal{M}_l) = 1/s$. After some algebra, one observes that for (1) with $(\boldsymbol{\mu} \,|\, \sigma^2) \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{K}_1)$, $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$, $P(\mathcal{D} \,|\, \mathcal{M}_l)$ is $P(\mathcal{D} \,|\, \mathcal{M}_l) = \frac{1}{|\boldsymbol{K}_1 + \boldsymbol{I}|^{\frac{1}{2}}} \frac{2^{\frac{n}{2}} \Gamma(\frac{n}{2})}{\left[\boldsymbol{y}'(\boldsymbol{K}_1 + \boldsymbol{I})^{-1} \boldsymbol{y}\right]^{\frac{n}{2}} (\sqrt{2\pi})^n}$. Plugging in the expression for $P(\mathcal{D} \,|\, \mathcal{M}_l)$ thus obtained, one finds the posterior predictive distribution as a weighted average of t densities. We adopt the choice of m suggested in ([13]) to have a window of $[\lceil 2log(p) \rceil, min(n, p)]$, which implies that the number of possible models to be averaged across is $s = min(n, p) - \lceil 2log(p) \rceil + 1$. Note that the computation over different sets of $\boldsymbol{\Psi}, \lambda$ are not dependent on each other, the calculations are embarrassingly parallel with a trivial expense for combining. The main computational expense comes from the inversion of an $n \times n$ matrix under the lth random projection. In the next section, we develop approaches for accelerating this inversion.

3 Scaling to large n

Fitting (1) using model averaging requires computing inverses and determinants of covariance matrices of the order $n \times n$. In problems with large n, this adds a heavy computational burden of the order of $O(n^3)$. Additionally, as dimension increases, matrix inversion becomes more unstable with the propagation of errors due to finite machine precision. This problem is further exacerbated if the covariance matrix is nearly rank deficient.

To address such issues, existing solutions mainly rely on approximating $\mu(\cdot)$ by another process $\tilde{\mu}(\cdot)$, which is more tractable computationally. Some of the most recent popular approaches include ([14,15,16]). These approaches have the advantages of being easily adaptable, less sensitive to the choice of additional parameters and computationally stable.

We adapt ([16]) from usual GP regression to our compressed manifold regression setting. In particular, let $\tilde{\mu}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}\boldsymbol{x}) = E[\mu(\boldsymbol{\Psi}\boldsymbol{x}) \,|\, \boldsymbol{\Phi}\mu(\boldsymbol{X}\boldsymbol{\Psi}')]$, $\epsilon_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}\boldsymbol{x}) \,|\, \sigma^2 \sim$

$N(0, \sigma^2_\epsilon(\boldsymbol{x}))$, $\sigma^2_\epsilon(\boldsymbol{x}) = \sigma^2\left[\kappa(\boldsymbol{\Psi x}, \boldsymbol{\Psi x}; \lambda) - (\boldsymbol{\Phi k_x})'\left\{\boldsymbol{\Phi K}_1\boldsymbol{\Phi}'\right\}^{-1}(\boldsymbol{\Phi k_x})\right]$, $\boldsymbol{k_x} = (\kappa(\boldsymbol{\Psi x}, \boldsymbol{\Psi x}_1; \lambda), ..., \kappa(\boldsymbol{\Psi x}, \boldsymbol{\Psi x}_n; \lambda))'$. We model

$$y = \tilde{\mu}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi x}) + \epsilon_{\boldsymbol{\Phi}}(\boldsymbol{\Psi x}) + \epsilon,\ \epsilon \sim N(0, \sigma^2). \tag{3}$$

Denote $\boldsymbol{H}_1 = diag(\boldsymbol{K}_1 - \boldsymbol{K}_1\boldsymbol{\Phi}'(\boldsymbol{\Phi K}_1\boldsymbol{\Phi}')^{-1}\boldsymbol{\Phi K}_1) + \boldsymbol{I}$ and $\boldsymbol{H}_2 = \boldsymbol{K}_1\boldsymbol{\Phi}'(\boldsymbol{\Phi K}_1\boldsymbol{\Phi}')^{-1}\boldsymbol{\Phi}$, $\boldsymbol{K}_2 = \{\kappa(\boldsymbol{x}^*_i, \boldsymbol{x}_j; \lambda)\}_{i,j=1}^{n_{pred},n}$, $\boldsymbol{K}_3 = \{\kappa(\boldsymbol{x}^*_i, \boldsymbol{x}_j; \lambda)\}_{i,j=1}^{n_{pred},n_{pred}}$ and $\boldsymbol{H}_3 = diag(\boldsymbol{K}_3 - \boldsymbol{K}_2\boldsymbol{\Phi}'(\boldsymbol{\Phi K}_1\boldsymbol{\Phi}')^{-1}\boldsymbol{\Phi K}'_2) + \boldsymbol{I}$. Further assume $\boldsymbol{K}_{11,RGP} = \boldsymbol{H}_3 + \boldsymbol{K}_2\boldsymbol{\Phi}'\left(\boldsymbol{\Phi K}_1\boldsymbol{\Phi}'\right)^{-1}\boldsymbol{\Phi K}'_2$, $\boldsymbol{K}_{12,RGP} = \boldsymbol{K}_2\boldsymbol{H}'_2$, $\boldsymbol{K}_{22,RGP} = \boldsymbol{H}_1 + \boldsymbol{H}_2\boldsymbol{K}_1$. The predictive of $\boldsymbol{y}^*$ given $\boldsymbol{X}^*$ and $\boldsymbol{\Psi}, \boldsymbol{\Phi}, \lambda$ for new n_{pred} subjects marginalizing out $(\boldsymbol{\mu}, \sigma^2)$ over their posterior distribution is available analytically as $\boldsymbol{y}^*|\boldsymbol{x}^*_1, ..., \boldsymbol{x}^*_{n_{pred}}, \boldsymbol{y} \sim t_n\left(\mu_{pred,RGP}, \sigma^2_{pred,RGP}\right)$, where $\mu_{pred,RGP} = \boldsymbol{K}_{12,RGP}\boldsymbol{K}^{-1}_{22,RGP}\boldsymbol{y}$, $b_2 = \boldsymbol{y}'(\boldsymbol{H}_1 + \boldsymbol{H}_2\boldsymbol{K}_1)^{-1}\boldsymbol{y}/2$ and $\sigma^2_{pred,RGP} = (2b_2/n)\left[\boldsymbol{K}_{11,RGP} - \boldsymbol{K}_{12,RGP}\boldsymbol{K}^{-1}_{22,RGP}\boldsymbol{K}'_{12,RGP}\right]$. Evaluating the above expression requires inverting matrices of order $m_{\boldsymbol{\Phi}} \times m_{\boldsymbol{\Phi}}$. Model averaging is again employed on a wide interval of possible m values in $[\lceil 2\log(p)\rceil, \min(m_{\boldsymbol{\Phi}}, p)]$.

An important question that remains is how much information is lost in compressing the high-dimensional feature vector to a much lower dimension? In particular, one would expect to pay a price for the huge computational gains in terms of predictive performance or other metrics. We address this question in two ways. First we argue satisfactory theoretical performance in prediction in a large p, large n asymptotic paradigm in Section 4. Then, we will consider practical performance in finite samples using simulated and real data sets.

4 Convergence analysis

This section provides theory on posterior convergence in the large n and p setting. The feature vector $\boldsymbol{x}$ is assumed to be $\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{\delta}$, $\boldsymbol{z} \in \mathcal{M}$, $\boldsymbol{\delta} \in \mathcal{R}^p$. Compressing the feature vector results in compressing $\boldsymbol{z}$ and the noise followed by their addition, $\boldsymbol{\Psi x} = \boldsymbol{\Psi z} + \boldsymbol{\Psi\delta}$. We would like to show that such compression results in near "optimal" inference. We build such a result from two angles.

(A) We show that when features lie on a manifold, random compression followed by a GP regression leads to optimal convergence properties. In particular, using $\{\boldsymbol{\Psi z}_i\}_{i=1}^n$ as features in GP regression yields the optimal rate of convergence.

(B) It is argued that noise compression through $\boldsymbol{\Psi}$ mitigates the deleterious effect of noise in $\boldsymbol{x}$ on the resulting performance.

Let $\mu_0(\cdot)$ and $\mu(\cdot)$ be the true and the fitted regression functions respectively. Define $\rho(\mu, \mu_0)^2 = \frac{1}{n}\sum_{i=1}^n(\mu(\boldsymbol{x}_i) - \mu_0(\boldsymbol{x}_i))^2$ as the distance between μ, μ_0 under a fixed design. When the design is random, let $\rho(\mu, \mu_0)^2 = \int_{\mathcal{M}}(\mu(\boldsymbol{x}) - \mu_0(\boldsymbol{x}))^2F(d\boldsymbol{x})$, where F is the marginal distribution of the features. Denote $\Pi(\cdot|y_1, ..., y_n)$ to be the posterior distribution given $y_1, ..., y_n$. Then the interest lies in the rate at which posterior contracts around μ_0 under the metric $\rho(\cdot, \cdot)$. This calls for finding a sequence $\{\zeta_n\}_{n\geq 1}$ of lower bounds such that

$$\Pi(\rho(\mu, \mu_0) > \zeta_n \mid y_1, ..., y_n) \to 0, \text{ as } n \to \infty. \tag{4}$$

Definition: Given two manifolds $\mathcal{M}$ and $\mathcal{N}$, a differentiable map $f : \mathcal{M} \to \mathcal{N}$ is called a diffeomorphism if it is a bijection and its inverse $f^{-1} : \mathcal{N} \to \mathcal{M}$ is differentiable. If these functions are r times continuously differentiable, f is called a C^r-diffeomorphism.

Our analysis builds on the following result (Theorem 2.3 in [10]).

Theorem 4.1 *Assume $\mathcal{M}$ is a d dimensional $\mathcal{C}^{r_1}$ compact sub-manifold of $\mathcal{R}^p$. Let $G : \mathcal{M} \to \mathcal{R}^p$ be the embedding map so that $G(\mathcal{M}) \simeq \mathcal{M}$. Further assume $T : \mathcal{R}^p \to \mathcal{R}^m$ is a dimensionality reducing map s.t. the restriction $T_{\mathcal{M}}$ of T on $G(\mathcal{M})$ is a $\mathcal{C}^{r_2}$-diffeomorphism onto its image. Then for any $\mu_0 \in \mathcal{C}^s$ with $s \leq \min\{2, r_1 - 1, r_2 - 1\}$, a Gaussian process prior on μ with features $\{T(\boldsymbol{z}_i)\}_{i=1}^n$, $\boldsymbol{z}_i \in \mathcal{M}$, leads to a posterior contraction rate at least $\zeta_n = n^{-s/(2s+d)}\log(n)^{d+1}$, which is the minimax optimal adaptive rate (ignoring the log factor). This is a huge improvement upon the minimax optimal adaptive rate of $n^{-s/(2s+p)}$ without the manifold structure in the features.*

We use the above result in our context. Define the linear transformation $T(\boldsymbol{z}) = \boldsymbol{\Psi}\boldsymbol{z}$. Using the property of random projection matrix, we have that, given $\kappa \in (0,1)$, if the projected dimension $m > O(\frac{m}{\kappa^2}\log(Cp\kappa^{-1})\log(\phi_n^{-1}))$ then with probability greater than $1-\phi_n$, the following relationship holds for every point $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{M}$,

$$(1-\kappa)\sqrt{\frac{m}{p}}||\boldsymbol{z}_i - \boldsymbol{z}_j|| < ||T(\boldsymbol{z}_i) - T(\boldsymbol{z}_j)|| < (1+\kappa)\sqrt{\frac{m}{p}}||\boldsymbol{z}_i - \boldsymbol{z}_j||, \tag{5}$$

implying that T is a diffeomorphism onto its image with probability greater than $(1-\phi_n)$. Define $\mathcal{A}_n = \{\text{Equation } 5 \text{ holds}\}$ so that $P(\mathcal{A}_n) > 1-\phi_n$.

$$\begin{aligned}\Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n) &= \Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n,\mathcal{A}_n)P(\mathcal{A}_n) + \\ &\quad \Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n,\mathcal{A}_n')P(\mathcal{A}_n') \\ &< \Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n,\mathcal{A}_n) + P(\mathcal{A}_n') < \Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n,\mathcal{A}_n) + \phi_n.\end{aligned}$$

On $\mathcal{A}_n$, T is a diffeomorphism. Therefore, Theorem 4.1 implies that with features $\{T(\boldsymbol{z}_i)\}_{i=1}^n$ $\Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n,\mathcal{A}_n) \to 0$. Finally, assuming $\phi_n \to 0$ yields $\Pi(d(\mu,\mu_0) > \zeta_n|y_1,...,y_n) \to 0$ with features $\{T(\boldsymbol{z}_i)\}_{i=1}^n$. This proves (A).

Let $\boldsymbol{\Psi}^{(l)}$ be the l-th row of $\boldsymbol{\Psi}$, $l = 1,...,m$. Denote $\boldsymbol{\Delta} = [\boldsymbol{\delta}_1 : \cdots : \boldsymbol{\delta}_n] \in \mathcal{R}^{p\times n}$ and assume $\boldsymbol{z}_i$ is the i-th row of $\boldsymbol{\Delta}$. Using Lemma 2.9.5 in ([17]), we obtain $\sqrt{p}\sum_{j=1}^p \Psi_{lj}\boldsymbol{z}_j \to N(\boldsymbol{0}, \text{Cov}(\boldsymbol{z}_1))$. Therefore, $\sum_{j=1}^p \Psi_{lj}\boldsymbol{z}_j = O_p(p^{-1/2})$, reducing the magnitude of noise in the original features. Hence (B) is proved. Thus, even if noise exists, asymptotic performance of $\{T(\boldsymbol{x}_i)\}_{i=1}^n$ will be similar to $\{T(\boldsymbol{z}_i)\}_{i=1}^n$ in the GP regression (which by (A) has optimal asymptotic performance).

5 Experiments

We assess the predictive performance of compressed Gaussian process (CGP) regression, in terms of mean squared prediction error (MSPE), coverage and lengths of 95% prediction intervals (PI), in a number of simulation examples and a real data example. We consider various numbers of features (p), sample size (n) and level of noise in the features to study their impact on the performance.

5.1 Competiotrs

In all the experiments out of sample predictive performance of the proposed CGP regression was compared to that of uncompressed Gaussian process (GP), BART (Bayesian Additive Regression Trees) ([18]), RF (Random Forests) ([6]) and TGP (Treed Gaussian process) ([19]). Unfortunately, with massive number of features, traditional BART, RF and TGP are computationally prohibitive. Therefore, we consider compressed versions in which we generate a single projection matrix to obtain a single set of compressed features, running the analysis with compressed features instead of original features. This idea leads to compressed versions of random forest (CRF), Bayesian additive regression tree (CBART) and Treed Gaussian process (CTGP). These methods entail faster implementation when the number of features is massive. As a default in these analyses, we use $m = 60$, which seems to be a reasonable choice of upper bound for the dimension of the linear subspace to compress to. Additionally, we implement a two stage GP procedure, where in the first stage Laplacian eigenmap is employed to obtain lower dimensional representations of the high dimensional features followed by a GP regression on these lower dimensional features. We denote this procedure by 2GP. Finally, we employ a two stage technique of clustering the massive sample into a number of clusters followed by fitting Lasso in each of these clusters. To facilitate clustering of high dimensional features in the first stage, we use the spectral clustering algorithm outlined in ([20]). Once observations are clustered, separate Lasso is fitted in each of these clusters. We refer to this procedure as distributed supervised learning (DSL).

5.2 Simulation Experiments:Manifold Regression on the Swiss Roll

To provide some intuition for our model, we start with a toy example where the distribution of the response is a nonlinear function of the coordinates along a swissroll (see Figure 1(a)), which is

embedded in a high dimensional ambient space. To be more specific, we sample manifold coordinates, $t \sim U(\frac{3\pi}{2}, \frac{9\pi}{2})$, $h \sim U(0, 5)$. A high dimensional feature $\boldsymbol{x} = (x_1, ..., x_p)$ is then sampled following

$$x_1 = t\cos(t) + \delta_1,\ x_2 = h + \delta_2,\ x_3 = t\sin(t) + \delta_3,\ x_i = \delta_i,\ i \geq 4, \delta_1, .., \delta_p \sim N(0, \tau^2).$$

Finally responses are simulated to have nonlinear relationship with the features

$$y_i = \sin(5\pi t) + h^2 + \epsilon_i,\ \epsilon_i \sim N(0, 0.02^2). \tag{6}$$

Clearly, $\boldsymbol{x}$ and y are conditionally independent given the low-dimensional signal manifold t, h.

The geodesic distance between two points on a swiss roll can be substantially different from their Euclidean distance in the ambient space $\mathcal{R}^p$. For example, in Figure 1(b) two points joined by the line segment have much smaller Euclidean distance than geodesic distance. Theorem 4.1 in Section 4 guarantees optimal performance when the compact sub-manifold $\mathcal{M}$ is sufficiently smooth, so that the locally Euclidean distance serves as a good approximation of the geodesic distance. The Swiss roll presents a challenging set up for CGP, since points on $\mathcal{M}$ that are close in a Euclidean sense can be quite far in a geodesic sense.

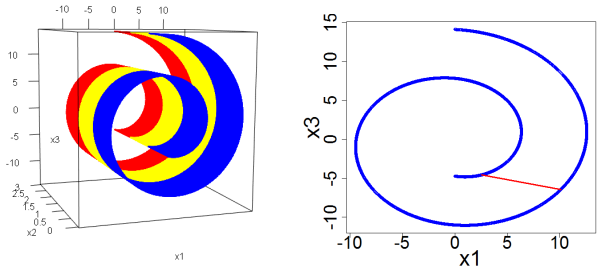

Figure 1: Swiss roll data.

MSPE of all the competing methods are calculated along with their bootstrap standard errors and presented in Table 1 for various feature noises. With smaller noise variance, CGP along with other compressed methods outperform uncompressed GP and 2GP. As τ increases and exceeds a certain *tipping point*, the manifold structure is more and more disrupted, with performance of all the competitors worsening. With increasing noise variance, performance of CGP and GP start becoming comparable, while the other compressed methods provide inferior performance. In all the simulation scenarios, DSL is the best performer in terms of MSPE, consistent with the routine use of DSL in large scale settings. However, the performance is extremely sensitive to the choice of clusters. In real data applications often inaccurate clustering leads to "suboptimal" performance, as will be seen in the data analysis. Additionally, we are not just interested in obtaining a point prediction approach, but want to obtain methods that provide an accurate characterization of predictive uncertainty. With this in mind, we additionally examine coverage probabilities and lengths of 95% predictive intervals (PIs).

Table 1: $MSPE \times 0.1$ along with the bootstrap $sd \times 0.1$ for all the competitors

	Competing Methods						
	τ	CGP	GP	CRF	CBART	DSL	2GP
$p = 10,000$	0.03	$0.63_{0.038}$	$2.02_{0.499}$	$0.92_{0.070}$	$0.79_{0.072}$	$0.44_{0.023}$	$3.80_{0.481}$
	0.06	$1.63_{0.092}$	$1.70_{0.287}$	$2.20_{0.176}$	$1.58_{0.111}$	$0.45_{0.015}$	$4.05_{0.434}$
	0.10	$2.31_{0.161}$	$3.07_{0.360}$	$3.44_{0.136}$	$2.74_{0.068}$	$0.50_{0.035}$	$4.10_{0.350}$
$p = 20,000$	0.03	$1.24_{0.042}$	$2.31_{0.418}$	$1.62_{0.070}$	$1.22_{0.082}$	$0.48_{0.024}$	$3.93_{0.592}$
	0.06	$2.01_{0.104}$	$2.23_{0.323}$	$2.99_{0.224}$	$2.59_{0.146}$	$0.47_{0.016}$	$4.10_{0.372}$
	0.10	$3.48_{0.217}$	$3.29_{0.330}$	$4.21_{0.224}$	$3.84_{0.191}$	$0.53_{0.080}$	$4.55_{0.481}$

Figure 2 shows that CGP, GP and CBART demonstrate satisfactory predictive coverage while CRF and DSL produce extremely narrow PI's resulting in gross under-coverage. Importantly, CGP

achieves such a predictive coverage with much narrower interval widths compared to GP, in presence of low feature noise. In terms of computation also CGP is most efficient. DSL and 2GP suffer from eigen-decomposition of an $n \times n$ matrix and computing an $n \times n$ distance matrix. CTGP is computationally prohibitive for large n (not shown here), although, for moderate n it produces inferior results than CGP. On the other hand, 2GP doesn't improve much upon GP.

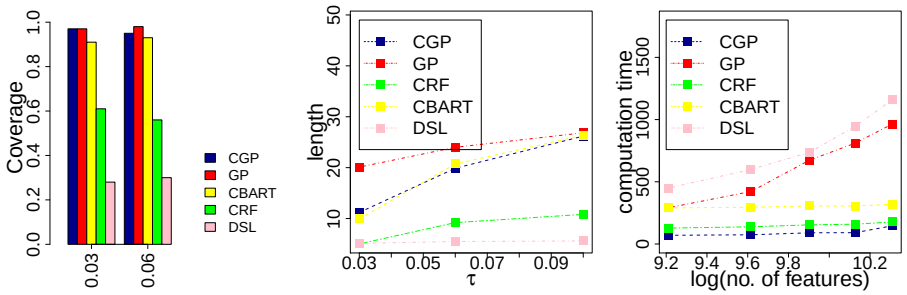

Figure 2: left panel: coverage of 95% predictive intervals, x-axis presents τ; middle panel: length of 95% predictive intervals; right panel: computation time in seconds for $n = 5000$

5.3 Isomap Face Dataset

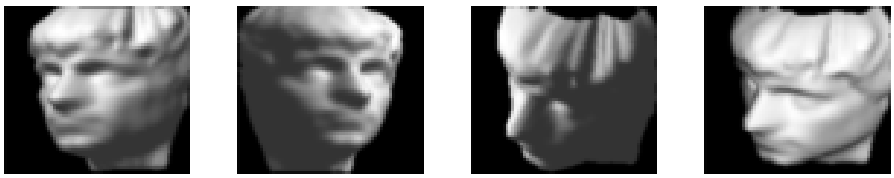

Figure 3: Representative images from the Isomap face data.

In our simulation examples, the underlying manifold is three dimensional and can be directly visualized. In this section we analyze image data where both the dimension and the structure of the underlying manifold is unknown. The dataset consists of 698 images of an artificial face and is referred to as the *Isomap face data* ([1]). A few such representative images are presented in Figure 3. Each image represents a two dimensional projection of a 3D-image in the form of a matrix of the order 64×64 pixels in size. Intuitively a limited number of additional features are needed for different views of the face. This is confirmed by the recent work of ([21]) where the intrinsic dimensionality is estimated to be small from these images. More details about the dataset can be found in `http://isomap.stanford.edu/datasets.html`.

We apply CGP and all the competitors to the dataset to assess relative performances. To set up the regression problem, we consider horizontal pose angles (vary in $[-75^0, 75^0]$) of the images, after standardization, as the responses. The features are taken $64 \times 64 = 4096$ dimensional vectorized images for each sample. To deal with more realistic situations, $N(0, \tau^2)$ noise is added to each pixel of the images, with varying τ, to make predictive inference more challenging from the noisy images. We carry out random splitting of the data into $n = 648$ training cases and $n_{pred} = 50$ test cases. To avoid spurious inference due to small validation set, this experiment is repeated 20 times.

It is clear from Table 2 that CGP along with its compressed competitors explain a lot of variation in the response. GP and 2GP are the worst performers in terms of MSPE. DSL also performs much worse than the compressed competitors. This is consistent with our experience that, in the presence of a complex and unknown manifold structure along with noise, DSL can be unreliable relative to CGP which tends to be more robust to the type of manifold and noise level. To assess how well calibrated these methods are, Figure 4 provides coverage probabilities along with the length of PI's

Table 2: MSPE and standard error (computed using 100 bootstrap samples) for all the competitors over 20 replications

τ	CGP	GP	CBART	CRF	DSL	2GP
0.03	$0.14_{0.059}$	$0.92_{0.074}$	$0.06_{0.005}$	$0.05_{0.007}$	$0.68_{0.023}$	$0.95_{0.062}$
0.06	$0.09_{0.006}$	$0.79_{0.056}$	$0.09_{0.007}$	$0.09_{0.008}$	$0.75_{0.015}$	$0.94_{0.041}$
0.10	$0.12_{0.008}$	$0.83_{0.077}$	$0.12_{0.005}$	$0.13_{0.011}$	$0.54_{0.014}$	$0.92_{0.013}$

for all the competitors. It is evident from the figure that CGP, GP and CBART yield excellent coverage. However, for CGP and CBART this coverage is achieved with much narrower predictive intervals compared to GP and 2GP. On the other hand, both CRF and DSL (not shown) produce extremely narrow predictive intervals resulting in severe under-coverage.

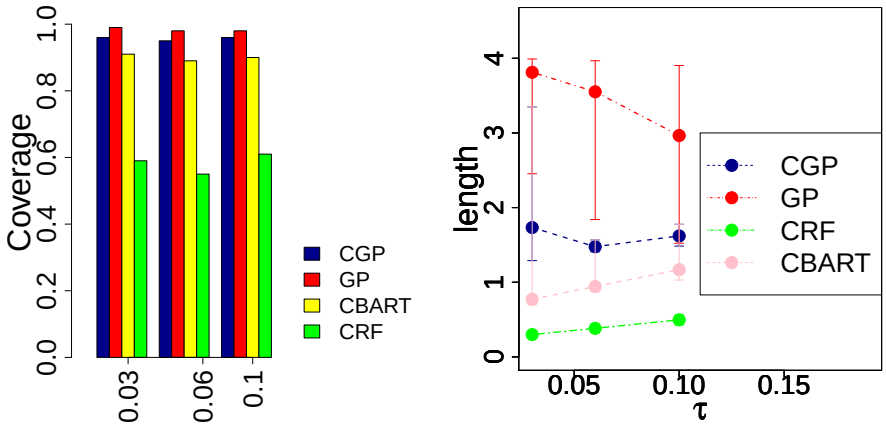

Figure 4: coverage and length of 95% PI's for CGP, GP, CBART, CRF. 95% CI's are shown at each point in figure 4(b)

6 Discussion

The overarching goal of this article is to develop nonparametric regression methods that scale to massive n and/or p when features lie on a *noise corrupted* manifold. The statistical and machine learning literature is somewhat limited in robust and flexible methods that can accurately provide predictive inference for massive n and p, while taking into account the geometric structure. We develop a method based on nonparametric *low-rank* Gaussian process methods combined with random feature compression to accurately characterize predictive uncertainties quickly, bypassing the need to estimate the underlying manifold. The computational template exploits model averaging to limit sensitivity of the inference to the specific choices of the random projection matrix $\mathbf{\Psi}$. The proposed method is also guaranteed to yield minimax optimal convergence rates.

There are many future directions motivated by our work. For example, the present work is not able to estimate the true dimensionality of the noise corrupted manifold. Arguably, a nonparametric method that can simultaneously estimate the intrinsic dimensionality of the manifold in the ambient space would improve performance both theoretically and practically. One possibility is to simultaneously learn the marginal distribution of the features, accounting for the low-dimensional structure. Other possible directions include adapting to massive streaming data where inference is to be made online. Although random compression both in n and p provides substantial benefit in terms of computation and inference, it might be worthwhile to learn the matrix $\mathbf{\Psi}$, $\mathbf{\Phi}$ while attempting to limit the associated computational burden.

Reference

[1] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323, 2000.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373-1396, 2003.

[3] R. Guerrero, R. Wolz, and D. Rueckert. Laplacian eigenmaps manifold learning for landmark localization in brain mr images. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2011*, pages 566-573. Springer, 2011.

[4] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1-8. IEEE, 2008.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5-32, 2001.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1-122, 2011.

[8] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783-1816, 2005.

[9] M. Chen, J. Silva, J. Paisley, C. Wang, D. B. Dunson, and L. Carin. Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds. *Signal Processing, IEEE Transactions on*, 58(12):6140-6155, 2010.

[10] Y. Yang and D. B. Dunson. Bayesian manifold regression. *arXiv preprint arXiv:1305.0617*, 2013.

[11] O. A. Maillard and R. Munos. Compressed least-squares regression. In *NIPS*, pages 1213-1221, 2009.

[12] M. M. Fard, Y. Grinberg, J. Pineau, and D. Precup. Compressed least-squares regression on sparse spaces. In *AAAI*, 2012.

[13] R. Guhaniyogi and D. B. Dunson. Bayesian compressed regression. *arXiv preprint arXiv:1303.0642*, 2013.

[14] A. J. Smola and B. Scholkopf. Sparse greedy matrix approximation for machine learning. 2000.

[15] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudoinputs. In *NIPS*, 1257-1264, 2006.

[16] A. Banerjee, D. B. Dunson, and S. T. Tokdar. Efficient gaussian process regression for large datasets. *Biometrika*, 100(1):75-89, 2013.

[17] A. W. Van der Vaart and J. A. Wellner. *Weak convergence and empirical processes*. Springer, 1996.

[18] H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266-298, 2010.

[19] R. B. Gramacy and H. K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119-1130, 2008.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering analysis and an algorithm. In *NIPS*, 849-856, 2001.

[21] A. Aswani, P. J. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48-81, 2011.