Incremental Nonparametric Bayesian Regression

Frank Wood, Dan H. Grollman, K. A. Heller, Odest
Chad Jenkins and Michael Black

Department of Computer Science
Brown University
Providence, Rhode Island 02912

Incremental Nonparametric Bayesian Regression

F. Wood[1], D. H. Grollman[2], K. A. Heller[1], O. C. Jenkins[2], and M. Black[2]

[1] Gatsby Computational Neuroscience Unit
University College London
London WC1N 3AR, UK
{fwood,heller}@gatsby.ucl.ac.uk

[2] Department of Computer Science
Brown University
Providence, RI 02912-1910, USA
{dang,cjenkins,black}@cs.brown.edu

Abstract. In this paper we develop an *incremental* estimation algorithm for infinite mixtures of Gaussian process experts. Incremental, local, non-linear regression algorithms are required for a wide variety of applications, ranging from robotic control to neural decoding. Arguably the most popular and widely used of such algorithms is currently Locally Weighted Projection Regression (LWPR) which has been shown empirically to be both computationally efficient and sufficiently accurate for a number of applications. While incremental variants of non-linear Bayesian regression models have superior theoretical properties and have been shown to produce better function approximations than LWPR, they suffer from high computational and storage costs. Through exploitation of locality, infinite mixtures of Gaussian process experts (IMGPE) offer the same function approximation performance with reduced computation and storage cost. Our contribution is an incremental regression approach that has the theoretical benefits of a fully Bayesian model and computational benefits that derive from exploiting locality.

1 Introduction

Demand for incremental, online learning algorithms arises from fields as diverse as robotic control and planning, neural modeling, active learning, and reinforcement learning. *Incremental* algorithms allow each datapoint to be processed once, in sequence. *Online* learning algorithms are characterized by the property that model estimation, which incorporates new data, is fast enough to be interleaved between closely spaced predictions from the model.

For example consider learning from demonstration [1, 2], the problem of incrementally training an online controller for a robot through human interaction. One approach is to incrementally learn a model which maps from a desired robot configuration (the input) to the forces required to achieve that configuration (the output) from human-provided demonstration training data. In [3] locally weighted projection regression (LWPR) [1] was used to incrementally

learn such a model. LWPR is a local, incremental regression algorithm. Being "local" means that LWPR divides up the input space into regions (called receptive fields) and builds a regressor for each. This not only allows LWPR to approximate extremely complicated functions with nonlinearities and discontinuities in the input/output map, but also makes it very fast because prediction only requires querying a simple local expert.[1]

LWPR is not the only algorithm suitable for applications that possess the incremental requirement which arises from learning from human demonstration (see [4, 5, 6] among many others); however, we focus on it as a basis for comparison in this paper because it has desirable computational benefits that derive from exploiting local learning, is widely used, and produces good results in a variety of domains, particularly the kind of robotics applications in which we are ourselves are interested.

Unfortunately LWPR defines locality in terms of an ad hoc algorithmic procedure for partitioning the input space. This means that LWPR is not fully probabilistic and therfore cannot be compared or combined with other probabilistic models used in control algorithms. Also, it is known that LWPR can be bettered in terms of modeling accuracy with even a single Gaussian process (GP) regressor [7]. Unfortunately GP regression has $O(n^3)$ computation and $O(n^2)$ storage cost, where n is the number of input/output pairs, whereas LWPR has $O(n)$ computation and $O(k)$ storage cost, where k is a constant.

Infinite mixtures of GP experts (IMGPE) models [8, 9] are fully probabilistic (Bayesian) local regression models that have accuracy no worse than GP regressors and reduced computation and storage costs ($O(sn^2\log(n)$ computation and $O(sn^2/\log(n))$ storage, where s is the number of samples used to represent the posterior) [8, 9]. They partition the input space into local regions using a nonparametric Bayesian mixture model of the input space rather than an ad hoc algorithmic procedure. However, only batch estimation algorithms for IMGPEs currently appear in the literature, ruling them out for the kinds of incremental, online applications suggested above. In this paper we address this problem by introducing incremental estimation for IMGPE models.

We see LWPR and our incremental IMGPE model as representing two endpoints on a continuum of incremental local regression algorithms; at one end algorithms are very fast but lack a probabilistic framework and are potentially inaccurate, while at the other they are very slow, but fully Bayesian and are therefore optimally accurate (given the modeling assumptions). Our contribution, while not yet computationally efficient enough for our intended robotics applications, is the establishment a theoretical endpoint from which practical Bayesian incremental, local regression approaches can be developed through approximate estimation schemes or simplifications of the IMGPE model (for instance, using simpler local regressors). Promising results from empirical comparisons of accuracy between LWPR and incremental IMGPE models, presented in Section 5, support the idea of pursuing such ends.

---

[1] Some variants of LWPR average the output of all local regressors.

2　Background

LWPR and IMGPE models are both designed to estimate potentially non-linear, discontinuous mappings from some input space, here $\mathbb{R}^D$, to some output space, here $\mathbb{R}$. Both are supervised methods meaning that they are both given noisy input/output pairs (observations) $\{\mathbf{x}_i, y_i\}_{i=1}^N$ from the true mapping and use them to produce an estimate of that mapping which can be used for prediction. However, they differ fundamentally as learning methods; LWPR exists as a set of algorithms, while the IMGPE provides a generative model and the ability to perform posterior estimation.

2.1　Locally Weighted Projection Regression

LWPR is an incremental algorithm that performs global nonlinear function approximation by combining the output of weighted local regressors. The input space is incrementally divided into a set of K (possibly overlapping) regions called "receptive fields", defined by center point $\mathbf{c}_k$ chosen at runtime and a Gaussian area of influence parameterized by a matrix $\mathbf{D}_k$, initially set to a default value $\mathbf{D}^*$ and updated incrementally. For each receptive field there is a local partial least squares (PLS) regressor [4], parameterized by ψ_k, which is incrementally fitted to the data assigned to the receptive field. New receptive fields are created as necessary, based on an empirical threshold, $w_{\texttt{gen}}$.

During each training iteration, all receptive fields calculate their "activation", or weight, measuring how close the new input, $\mathbf{x}'$, is to their center $\mathbf{c}_k$.

$$w_k(\mathbf{x}') = \exp(-\frac{1}{2}(\mathbf{x}' - \mathbf{c}_k)^\top \mathbf{D}_k(\mathbf{x}' - \mathbf{c}_k)) \tag{1}$$

Each receptive field then applies (weighted) incremental updates to the local PLS regressor, modifying its local parameter ψ_k accordingly. An iterative stochastic gradient ascent algorithm is employed to adjust the receptive field's influence $\mathbf{D}_k$, and all update equations are local in the sense that they only require sufficient statistics (i.e. no training data is retained).

For prediction, each receptive field produces its own estimate of the output, then either one or the weighted sum of all estimates are combined and returned as the prediction. LWPR can also provide output confidence bounds, for complete algorithmic details we refer the reader to [1, 3]. Code for LWPR is online at [10].

2.2　Infinite Mixture of GP Experts

IMGPE models are similar to LWPR in the sense that both models specify methods for dividing up the input space such that a single "expert" is responsible for approximating the underlying true mapping in that region of input space. Otherwise they differ fairly widely; training an LWPR model produces a single estimate of the underlying mapping. Because IMGPE models are Bayesian, training involves estimating a distribution over mappings. Predicting an output

using an IMGPE model also requires averaging over a distribution over mappings. Batch sampling, the only kind of estimation procedure currently available for IMGPE models, requires that all training data be present. However we will show it is possible to perform model estimation incrementally.

An IMGPE model is characterized by having an infinite Gaussian mixture model as an input gating mechanism [11] which stochastically "gates" each input to one of an infinite number of Gaussian process experts. The simplest way of conceptualizing an infinite Gaussian mixture model (IGMM) is as a finite Gaussian mixture model in the limit as the number of latent classes goes to infinity [12]. IGMM's may also be refered to as Dirichlet process mixture of Gaussian models. The key characteristic embodied by this gating network is assignment of data to experts (and the number of experts) is inferred from the data (and then marginalized out) rather than set a prori or arrived at through an ad hoc procedure. Better still, since these hidden variables can be averaged out, the overall model is robust to uncertainty that arises from partitioning the input space.

Instead of the "lightweight" PLS receptive field experts employed by LWPR, IMGPE models use Gaussian process experts. A Gaussian process (GP) is a prior over functions [13] which is parameterized by a kernel (covariance function) and its parameters. The kernel function computes the similarity or distance between pairs of input points. In a regression setting the GP prior serves to regularize the function mapping inputs to outputs. Intuitively a GP expert is a Gaussian process regressor trained on the input/output pairs local to the expert.

3 Batch IMGPE

The IMGPE model we use is similar to that detailed in [8, 9] but differs in at least one important way. We review batch sampling for theses models here and elaborate on the aspects specific to our own. In reviewing batch sampling for these models we also establish many of the conditional distributions necessary for incremental estimation of this model.

The IMGPE model is a generative latent variable model in which indicator variables are used to identify which local expert gave rise to a particular input/output pair. We call these variables *latent class indicator variables*. The N latent class indicator variables (one for each input/ouput pair) are $\mathbf{z} = \{z_i\}_{i=1}^N$. These class indicator variables are generated by a Chinese restaurant process (CRP) [14] with concentration parameter α. The concentration parameter specifies how uniform the assignment of input/output pairs to experts is thought to be a priori (large α implies many experts). The CRP prior

$$P(z_i = k|\mathbf{z}_{-i}) = \begin{cases} \frac{m_k}{N+\alpha-1}, \, k \leq K_+ \\ \frac{\alpha}{N+\alpha-1}, \, k = K_+ + 1 \end{cases} \tag{2}$$

can be described as a sequential process that generates sequences of integers where the probability that the next integer in the sequence is k is proportional to the number of times k has already appeared in the sequence. The probability

that the next integer takes on a new value of k is proportional to α. Here $m_k = \sum_{i=1}^{N} \mathbb{I}(z_i = k)$ is the number of times k appears in the sequence ($\mathbb{I}()$ is the indicator function), K_+ is the number of unique integers that appear, and N is the total sequence length.

Generating the class indicator variables gives rise to some total number of classes K_+. The IMGPE model specifies an expert for each class consisting of a multivariate-normal input model and GP regressor. In other words, there are K_+ multivariate-normal classes that generate input points, and a GP expert for each class which is responsible for generating outputs given the inputs. Each input space model has mean parameter μ_k and covariance parameter Σ_k. These input class parameters are themselves drawn from a standard normal-inverse-Wishart conjugate prior. This choice of prior allows the user to influence how input space is partitioned by the model. By choosing a conjugate prior we have simplified the model of [8] in a way that makes possible our incremental estimation approach but does not significantly sacrifice the expressivitiy of the model. All N input points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ are drawn from the input space models indicated by their class indicator variables. Once the input points are generated, each GP expert generates corresponding outputs $\mathbf{y}_k$ for the input points assigned to its class.

To summarize:

$$
\begin{aligned}
z_i &\sim \mathrm{CRP}(\alpha) \\
\Sigma'_k &\sim \mathrm{Inverse\text{-}Wishart}_{\nu_0}(\Lambda_0) \\
\mu'_k &\sim \mathrm{Multivariate\text{-}normal}(\mu_0, \Sigma_k/\kappa_0) \\
\mathbf{x}_i|z_i &\sim \mathrm{Multivariate\text{-}normal}(\mu'_{z_i}, \Sigma'_{z_i}) \\
\mathbf{y}_k|\mathbf{X}_k, \theta &\sim \mathrm{Multivariate\text{-}normal}(0, \mathbf{Q}_k)
\end{aligned}
$$

The last line of the generative model summary is the conditional distribution of the outputs $\mathbf{y}_k = \{y_i : z_i = k\}_{i=1}^{N}$ given the inputs $\mathbf{X}_k = \{\mathbf{x}_i : z_i = k\}_{i=1}^{N}$ in class k and global GP parameters $\theta = \{v^0, v^1, \{w^d\}_{d=1}^{D}\}$. Each GP regressor defines a joint Multivariate-normal distribution over the outputs in its partition, $\mathbf{y}_k$, with covariance $\mathbf{Q}_k$ characterized by the kernel function

$$
Q_k(\mathbf{x}_e, \mathbf{x}_f) = v^0 e^{-\frac{1}{2}\sum_{d=1}^{D}(\mathbf{x}_e^d - \mathbf{x}_f^d)^2/w^{d^2}} + \mathbb{I}(i = h)v^1
$$

where $\mathbf{x}_e^d$ is the dth dimension of input $\mathbf{x}_e$. Note that $Q_k(\mathbf{x}_e, \mathbf{x}_f)$ will only be evaluated for $\mathbf{x}_e, \mathbf{x}_f \in \mathbf{X}_k$. These GP kernel parameters can be intuitively intrepreted in the following way: v^0 is the expected range of the output, v^1 is the residual noise variance (expected distance between the predicted output and actual output), and w^d is the kernel width in the dth dimension.

From this is straightforward to write down the joint distribution of the inputs $\mathbf{X}$, outputs $\mathbf{y}$, and class labels $\mathbf{z}$ defined by our IMGPE model:

$$
P(\mathbf{X}, \mathbf{y}, \mathbf{z}; \Omega) = P(\mathbf{X}|\mathbf{z}; \Omega)P(\mathbf{z}|\Omega) \prod_{k=1}^{K_+} P(\mathbf{y}_k|\mathbf{X}_k, \Omega). \tag{3}
$$

Here $\Omega = \{\alpha, \mu_0, \kappa_0, \Lambda_0, \nu_0, \theta\}$, is the collection of all parameters. We note that this joint distribution is proportional to the posterior probability of the model

(parameterized by $\mathbf{z}$) given the data (the normalizing constant is intractable to compute for this model). To simplify our notation, we will no longer make dependence on parameters explicit, i.e. we will write $P(\mathbf{X}|\mathbf{z})$ in place of $P(\mathbf{X}|\mathbf{z}; \Omega)$. In Eqn. 3 and those that follow we exploit the conjugacy of the the input parameter prior, and show the resulting joint with $\{\mu'_k, \Sigma'_k\}_{k=1}^{K_+}$ integrated out.

That we utilize a conjugate prior over the input space model parameters and analytically marginalize them out is the critical difference between our model specification and that in [8]. It is for this reason that incremental estimation is practical in our model. For purposes of expositional clarity and brevity we have also, in this paper, also opted to not to demonstrate GP hyperparameter estimation. This also is a simplification of the model in [8] which also makes incremental estimation more practical.

3.1   Batch Estimation

Estimating the model presented above involves drawing L samples $\{\mathbf{z}_\ell\}_{\ell=1}^L$, by simulating a Markov chain with equilibrium distribution given by Eqn. 3. As in [8], where such Markov chain Monte Carlo (MCMC) sampling methods were used to perform model estimation, we provide the distributions required to implement a Gibbs sampler for our model. In our case, we must be able to sample a single class label (z_i) conditioned on the remaining class labels ($\mathbf{z}_{-i}$) and the observations ($\mathbf{X}$ and $\mathbf{y}$). This conditional distribution can be written as

$$P(z_i = k|\mathbf{z}_{-i}, \mathbf{X}, \mathbf{y}) = P(z_i = k|\mathbf{z}_{-i})P(\mathbf{x}_i, y_i|z_i = k, \mathbf{X}\backslash\mathbf{x}_i, \mathbf{y}\backslash y_i). \qquad (4)$$

where $\mathbf{X}\backslash\mathbf{x}_i$ means the set of points $\mathbf{X}$ with $\mathbf{x}_i$ removed and $\mathbf{y}\backslash y_i$ means the vector $\mathbf{y}$ with the element y_i removed.

The first term on the right hand side of this expression can be computed using the CRP prior from Eqn. 2. The second term factorizes under our model to:

$$P(\mathbf{x}_i, y_i|z_i = k, \mathbf{X}\backslash\mathbf{x}_i, \mathbf{y}\backslash y_i) = P(\mathbf{x}_i|z_i = k, \mathbf{X}_k\backslash\mathbf{x}_i)P(y_i|z_i = k, \mathbf{X}_k, \mathbf{y}_k\backslash y_i)$$

where $\mathbf{X}_k\backslash\mathbf{x}_i = \mathbf{X}_k$ if $\mathbf{x}_i \notin \mathbf{X}_k$. Because we use conjugate priors on the input space model parameters, these distributions can be calculated directly. If $k \le K_+$

$$P(\mathbf{x}_i|z_i = k, \mathbf{X}_k\backslash\mathbf{x}_i) = \text{Student-}t_{\nu_k-D+1}(\mu_k, \mathbf{\Lambda}_k(\kappa_k + 1)/(\kappa_k(\nu_k - D + 1)))$$
$$P(y_i|z_i = k, \mathbf{X}_k, \mathbf{y}_k\backslash y_i) = \text{Normal}(\mathbf{k}^T\mathbf{Q}_{\backslash\mathbf{x}_i}^{-1}(\mathbf{y}_k\backslash y_i), Q_k(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{k}^T\mathbf{Q}_{\backslash\mathbf{x}_i}^{-1}\mathbf{k})$$

If $k = K_+ + 1$

$$P(\mathbf{x}_i|z_i = K_+ + 1) = \text{Student-}t_{\nu_0-D+1}(\mu_0, \mathbf{\Lambda}_0(\kappa_0 + 1)/(\kappa_0(\nu_0 - D + 1)))$$
$$P(y_i|z_i = K_+ + 1) = \text{Normal}(0, v^0 + v^1)$$

Here Student-$t_d(a, B)$ is a multivariate Student-t distribution with d degrees of freedom, mean parameter a, and scale matrix B. These equations follow Gel-

man et al. ([15] page 87) in making the following variable substitutions

$$\begin{aligned}
\mu_k &= \frac{\kappa_0}{\kappa_0+N}\mu_0 + \frac{m_k}{\kappa_0+m_k}\bar{y}_k \\
\kappa_k &= \kappa_0 + m_k \\
\nu_k &= \nu_0 + m_k \\
\mathbf{\Lambda}_k &= \mathbf{\Lambda}_0 + \mathbf{S_k} + \frac{\kappa_0 m_k}{\kappa_0+m_k}(\bar{y}_k - \mu_0)(\bar{y}_k - \mu_0)^T \\
\mathbf{S_k} &= \sum_{j:z_j=k}(y_j - \bar{y}_k)(y_j - \bar{y}_k)^T \\
\bar{y}_k &= \frac{1}{m_k}\sum_{j:z_j=k} y_j
\end{aligned}$$

The symbol $\mathbf{Q}_{\backslash \mathbf{x}_i}^{-1}$ denotes the covariance matrix for the kth GP expert with, if necessary, observation (x_i) removed. The vector $\mathbf{k} = [Q(\mathbf{x}_1, \mathbf{x}_i), Q(\mathbf{x}_2, \mathbf{x}_i), \ldots, Q(\mathbf{x}_{m_k}, \mathbf{x}_i)]^T$ is the covariance function evaluated at all points assigned to expert k except x_i (i.e. leaving out $Q(\mathbf{x}_i, \mathbf{x}_i)$). The output model parameters arise from incremental, partitioned updates of the GP covariance matrices [16].

MCMC sampling methods, including the Gibbs sampling method presented above, are batch methods in the sense that they require all observations to be collected before model estimation can proceed. Due to this fact, batch methods are inappropriate for applications like those discussed in the introduction. In the next section we introduce an incremental learning algorithm in order to address this deficiency, and in Section 5 we compare Gibbs sampling to our new incremental estimation approach.

4 Incremental IMGPE

In order to develop an incremental estimation algorithm for the IMGPE model we factor the posterior distribution (proportional to Eqn. 3) in the following way:

$$\begin{aligned}
&P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i)}, \mathbf{y}^{(1:i)}) \\
&\propto P(\mathbf{x}_i, y_i|\mathbf{z}^{(1:i)}, \mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)})P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)}). \quad (5)
\end{aligned}$$

Here $\mathbf{z}^{(1:i)}$ are the class identifiers up to and including input/output pair i (inputs $\mathbf{X}^{(1:i)}$ and outputs $\mathbf{y}^{(1:i)}$ are defined analagously).

We can draw samples from the posterior distribution using importance sampling provided that we can sample from the posterior predictive distribution, $P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)})$. We establish a recurrence for incremental posterior estimation, by noting that the posterior predictive distribution is related to the posterior including the previous input/output pair, $P(\mathbf{z}^{(1:i-1)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)})$, in the following way

$$\begin{aligned}
&P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)}) \\
&\propto \int P(\mathbf{z}^{(1:i)}|\mathbf{z}^{(1:i-1)})P(\mathbf{z}^{(1:i-1)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)})d\mathbf{z}^{(1:i-1)}
\end{aligned}$$

Suppose that we have L weights and samples $\{w_\ell^{i-1}, \mathbf{z}_\ell^{i-1}\}_{\ell=1}^L$, where $w_\ell^{i-1} \in \mathbb{R}$ is a weight with $\sum_{\ell=1}^L w_\ell^{i-1} = 1$, from $P(\mathbf{z}^{(1:i-1)}|\mathbf{X}^{(1:i-1)}, \mathbf{y}^{(1:i-1)})$ then we

can use Monte Carlo integration to write the posterior predictive distribution in the following way

$$P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i-1)},\mathbf{y}^{(1:i-1)}) \approx \sum_{\ell=1}^{L} w_{\ell}^{i-1} P(z_i|\mathbf{z}_{\ell}^{(1:i-1)}).$$

This is a mixture model and thus it can easily be sampled from. To sample $\mathbf{z}_{\ell}^{(1:i)}$ from this mixture model, first a component of the mixture model must be chosen according to its weight, and then the CRP prior (Eqn. 2) can be used to draw z_i given $\mathbf{z}_{\ell}^{(1:i-1)}$.

We now have the proposal distribution needed for our importance sampler. We can generate samples from the updated posterior $P(\mathbf{z}^{(1:i)}|\mathbf{X}^{(1:i-1)},\mathbf{y}^{(1:i-1)})$ by weighting samples drawn from the proposal distribution just outlined. From Eqn. 5, the updated weight w_{ℓ}^{i} of sample $\mathbf{z}_{\ell}^{(1:i)}$ is w_{ℓ}^{i-1} times the probability of the observation given its assignment to some expert, $P(\mathbf{x}_i, y_i|\mathbf{z}^{(1:i)}, \mathbf{X}^{(1:i-1)},\mathbf{y}^{(1:i-1)})$. Conveniently, we derived how to compute this term in the batch sampling section, Equation 5. The weights must be normalized after all have been computed.

In implementations of IMGPE incremental estimation, a set of GP's must be maintained for each particle. In each of these GPs the partitioned matrix inverse equations [16] must be used to incrementally integrate each new observation into the GP covariance matrix corresponding to the local expert to which they are assigned. Here the incremental updates for the for the kth partition's covariance matrix $\mathbf{Q}_k$ are illustrated. In the following note the subtle overloading of subscript semantics: $\mathbf{Q}_{m_k+1}^{-1}$ is still the inverse covariance matrix for partition k however the number of inputs, $m_k + 1$, is now indicated rather than merely the class k. Given $\mathbf{Q}_{m_k}^{-1}$ and a new input point x_i assigned to class k one can arrive at

$$\mathbf{Q}_{m_k+1}^{-1} = \begin{bmatrix} \begin{bmatrix} & \mathbf{M} & \end{bmatrix} & \begin{bmatrix} \mathbf{m} \end{bmatrix} \\ \begin{bmatrix} & \mathbf{m}^T & \end{bmatrix} & [\eta] \end{bmatrix}$$

by computing $\mathbf{M}$, $\mathbf{m}$, and η according to

$$\eta = (Q_k(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{k}^T \mathbf{Q}_{m_k}^{-1} \mathbf{k})^{-1}$$
$$\mathbf{m} = -\eta \mathbf{Q}_{m_k}^{-1} \mathbf{k}$$
$$\mathbf{M} = \mathbf{Q}_{m_k}^{-1} + \frac{1}{\eta} \mathbf{m}\mathbf{m}^T.$$

All of this together constitutes a recurrence whereby new observations can be incrementally integrated into the model. Notwithstanding the partitioned matrix updates, this style of recursive sampling is called sequential importance sampling and/or particle filtering [17]. While particle filtering has been used
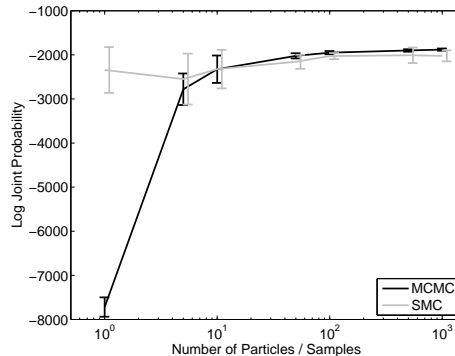
Fig. 1. Log probability of the synthetic data versus computational cost for both incremental (SMC) and batch (MCMC) estimates of an IMGPE model.

to do incremental estimation in nonparametric Bayesian mixture models ([18]), particle filtering in this mixture of experts setting is novel. Further improvements to the basic particle filter can be utilized to improve the basic sequential importance sampling particle filter described here, for instance those proposed in [18]. In all of our experiments we adopted the resampling policy in [18].

5 Experiments

In this section we establish two results; 1) that incremental estimation of IMGPE models is as good or better than batch estimation, and 2) that IMGPE models perform as well or better than LWPR for a number of problems. We demonstrate the first result by using both batch and incremental procedures to estimate a IMGPE model and show that the resulting models are equivalent. The second result is established by estimating IMGPE and LWPR models on several datasets and comparing prediction results. These experiments empirically demonstrate the validity of the incremental estimation approach for IMGPE models and establish some justification for continuing research into incremental, local Bayesian regression models.

5.1 Empirical validation of incremental estimation

To compare the performance of batch vs. incremental estimation we use each to estimate a model of the synthetic dataset from [8]. This synthetic dataset is a non-functional, discontinuous mapping with observation noise that varies across
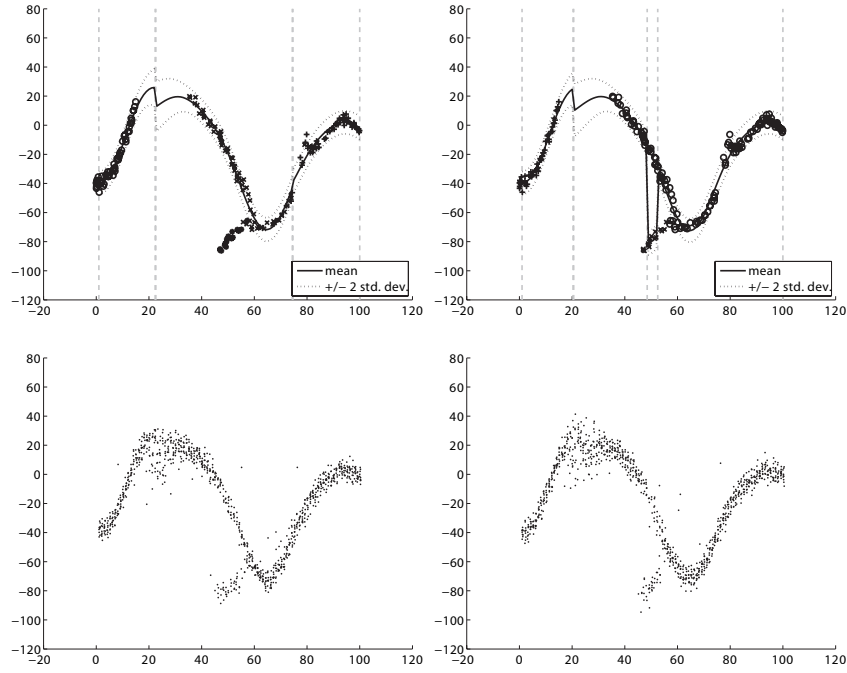
Fig. 2. Top row: maximum aposteriori (MAP) models of synthetic data. The left figure shows the MAP model from 500 incremental estimation particles, the right figure from 500 batch samples. The black symbols are the training input/output observations. Bottom row: horizontally jittered samples drawn from the entire estimated posterior at regularly spaced input points for both incremental (left) and and batch (right) estimators.

the input domain. The data were generated from the following model

$$f_1(x) = 0.25x^2 - 40 + \mathcal{N}(0, \sqrt{7}), \quad x \in [0, 15]$$
$$f_2(x) = -0.0625(x - 18)^2 + .5x + 20 + \mathcal{N}(0, \sqrt{7}), \quad x \in [25, 60]$$
$$f_3(x) = 0.008(x - 60)^3 - 70 + \mathcal{N}(0, \sqrt{4}), \quad x \in [45, 80]$$
$$f_4(x) = -\sin(0.25x) - 6 + \mathcal{N}(0, \sqrt{2}), \quad x \in [80, 100]$$

Note the gap of data when $x \in [15, 25]$ and the overlap when $x \in [45, 60]$ (Fig. 2 and Fig. 4). We are particularly interested in the overlap, and stipulate that proper performace on data in that range is to predict output from *either* of the two functions, but not to return the average of the two possible values.

We produced several estimates of the IMGPE model for this data. The estimates differed in the number of particles and samples used to produce the estimates, starting with one and going up to one thousand. For each of these settings we produced five different estimates of the model by initializing the random number generators with different seeds. The GP expert covariance function

(a) True map (b) LWPR estimate (c) Incremental IMGPE estimate
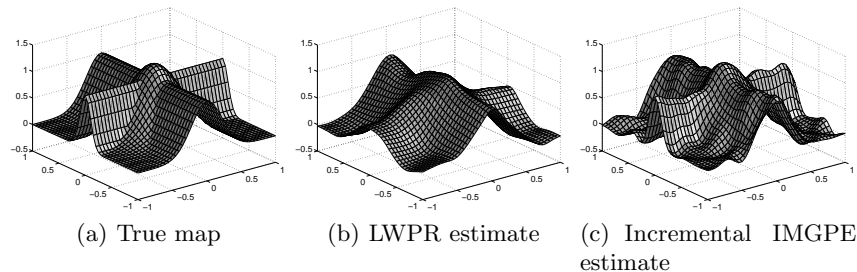
Fig. 3. One-pass LWPR model compared to the MAP incremental IMGPE model.

parameters used were $v_0 = 80, v_1 = 15$, and $w = 10$. The CRP concentration parameter used was $\alpha = .5$. Figure 1 displays the results from this experiment. The average maximum a posteriori (MAP) model score (highest log joint probability of the training data out of all particles and/or samples) is shown per number of particles/samples for both estimation procedures. The error bars (2 standard deviations) indicate how sensitive the MAP score is to initialization. The overall conclusion we draw from this figure is that incremental estimation produces similar results to batch estimation.

It is worth noting here that the number of particles used to estimate the model is a direct proxy for computational cost. While no "Bayesian" would be happy with a posterior estimate consisting of a single sample/particle as it does not provide a reasonable way of averaging over models, it is still possible to run the particle filter with a single particle. For practical applications where computational capacity is at a premium, doing so may not necessarily be the proper "Bayesian" thing to do, but it still may produce a reasonable first approximation to the MAP model. In Fig. 1 we see evidence that even with a very small number of particles, the incremental estimation approach may produce models that are reasonable for use in, for instance, resource constrained robotic systems.

The score of the MAP sample is a somewhat impoverished demonstration of the efficacy of incremental estimation of this model. For this reason in Figure 2 we plot more detailed results on this data set, for both incremental and batch estimation. In the top row the black symbols are the training input/output observations. The symbols indicate which expert that training observation was assigned to in the MAP model. The MAP model from incremental estimation has three experts whereas the MAP model from batch estimation has four. The vertical gray dashed lines indicate the regions of the input space that are gated to a particular expert. The solid black prediction line and accompanying dotted confidence intervals are created by predicting the output for 100 inputs equally spaced across the input domain. Examining the MAP sample alone is insufficient to demonstrate that the two distributions are substantially similar. The bottom row shows samples drawn from the entire posterior at the same test points. It is apparent from these output samples that the posterior distribution estimated by both models is similar. Regardless of the estimator, the model captures the gen-

	LWPR	IMGPE
Boston	73.1 ± 22.3	68.5 ± 17.9
Cross	0.017	0.004
Synthetic	92.9	22.2

Table 1. Comparison of LWPR and IMGPE on three datasets. Shown are average mean squared error of predictions on held-out data.

eral characteristics of the mapping, including both the upper and lower branches of the overlapping region.

5.2 Comparing incremental IMGPE to LWPR

Directly comparing an IMGPE model to an LWPR model on the same data is somewhat difficult, due in large part to the fact that the IMGPE model is a probabilistic model, whereas the LWPR model is not. To facilitate the comparison we ignore many of the probabilistic modeling benefits of the IMGPE approach and compare only the MAP model to the model produced by LWPR. Because we are no longer in a probabilistic paradigm, we can not use Bayes factors to compare the models, so we instead use mean square error between actual and predicted outputs for known, held-out input/output pairs. Another difficulty is that each algorithm has a number of free parameters. When possible we chose LWPR parameters that have appeared in the literature; however, when these were not available finite differencing gradient search methods were used to search for optimal LWPR parameters. IMGPE parameters were, in general, chosen such that they matched the scale of the problem but were otherwise chosen to be as uninformative as possible.

We perform three experiments comparing LWPR to IMGPE modeling; the results from which are shown in Table 1. In Figure 4 we compare an LWPR model of the synthetic data described in the previous section to the incremental IMGPE model estimate shown in Fig. 2. The large black dots correspond to the center of the LWPR receptive fields while the black line is the LWPR predicted output for an evenly spaced set of input points. The dashed line is the incremental IMGPE MAP sample prediction (same as in Fig. 2). When generating Fig. 4 the parameters (D^* and $w_{\texttt{gen}}$) of the LWPR model were selected by finite difference-based gradient ascent. The objective maximized was a modified mean squared error metric, defined as usual except in the area of overlap. There, error was measured with respect to the *closest* correct ouput value. We chose this metric because, as previously stated, we take correct behavior in the overlapping region to be a prediction belonging to *either* function. We report the results in Table 1 (Synthetic).

In Figure 3 we compare an LWPR model of a "cross" mapping (Fig. 3(a)) to which Gaussian noise $\mathcal{N}(0, .1)$ is added to an incremental IMGPE model of the same. LWPR was intialized as in [3] with parameters $D^* = 30$ and $w_{gen} = 0.2$
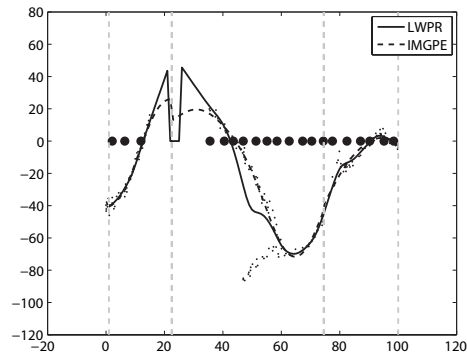
Fig. 4. Best LWPR model and MAP IMGPE model. Dots represent LWPR receptive field centers.

and produced a model with 27 receptive fields. The IMGPE model was initialized with $\mu_0 = 0$, $\Lambda_0 = 0.01\mathbf{I}$, $k_0 = 0.01$, $v_0 = 3$, $v_0 = 1$, $v_1 = .1$, $w = 0.15$, and $\alpha = 0.5$, and the particle filter was run with 500 particles, producing a model with two experts. The mean squared errors of output predictions for a test grid of 1681 novel input points were computed for both models; results are shown in Table 1. Although from Figure 3 one could assume that the incrementally estimated IMGPE model is overfitting, it should be noted that the prediction error achieved by the IMGPE model is nearly as good at the best LWPR prediction error published for this data. Moreover, in order to achieve this accuracy LWPR needed to see nearly ten thousand training examples and use over forty experts [3].

Finally, we compare LWPR prediction results [3] on the Boston housing data set (a standard regression dataset from the UCI machine learning dataset repository [19]) to that of an incremental IMGPE model on the same. All of the experiments thus far have had either one or two dimensional input spaces, but both LWPR and IMGPE models are capable of handling data with much higher input dimensionality. The results for the Boston dataset, which has an 13 dimensional input space, are shown in Table 1.

All the results presented in this section demonstrate that our new IMGPE-based incremental regression algorithm can perform comparably to, or better than LWPR on a variety of modeling tasks.

6 Discussion

The primary contribution of this paper is the development of an incremental approach to IMGPE model estimation. We have demonstrated that our proposed incremental estimation algorithm is valid by showing that the models estimated by this incremental procedure are equivalent to those arrived at through batch estimation. We also showed that for various kinds of data IMGPE models can

outperform LWPR models with respect to prediction error, giving credence to the argument that our model might be viewed as a starting point for developing new computationally efficient incremental, local Bayesian regression algorithms.

Unfortunately, there is still a wide gulf in computational cost between the incremental IMGPE model we propose and LWPR. This is mostly because the local experts in the IMGPE model are Gaussian processes, and each expert must retain all of the training datapoints assigned to it. Furthermore, although the GP is constructed incrementally via the partioned inverse equations, the space and time requirements of each GP expert are fundamentally those of a GP ($O(m_k{}^2)$ space and $O(m_k{}^3)$ time). This stands in stark contrast to the incremental PLS experts of LWPR which retain no training data and instead keep only projection directions and offsets. These GP computational costs were not a problem in the modeling tasks we considered, but could be a problem when modeling much larger datasets. There is, however, nothing in our approach that limits us to using GP experts. Any probabilistic incremental regressor will work as an expert in our model. It remains an open question whether to employ simple or complex local experts; the general success of LWPR indicates that simpler local experts may be sufficient. It may also not be necessary to discard GP experts in order to achieve computational efficiency. Simply limiting the amount of training data each expert can hold by employing sparse online Gaussian processes [6] may be sufficient to yield a practical competitor to LWPR.

Bibliography

[1] Atkeson, C.G., Schaal, S.: Robot learning from demonstration. In Fisher, D.H., ed.: International Conference on Machine Learning, Nashville, TN (1997) 12–20

[2] Nicolescu, M., Mataric, M.J.: Natural methods for robot task learning: Instructive demonstration, generalization and practice. In: International Joint Conference on Autonomous Agents and MultiAgent Systems, Melbourne, AUSTRALIA (2003) 241–248

[3] Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation **17** (2005) 1–33

[4] Wold, H.: Estimation of principle components and related models by iterative least squares. In Krishnaiaah, P., ed.: Multivariate Analysis. Academic Press, New York, NY (1966) 391–420

[5] Engel, Y., Mannor, S., Meir, R.: Sparse online greedy support vector regression (2002)

[6] Csató, L., Opper., M.: Sparse online Gaussian processes. Neural Computation (2002) 641–668

[7] Grollman, D.H., Jenkins, O.C.: Sparse incremental learning for interactive robot control policy estimation. In: IEEE International Conference on Robotics and Automation, Pasadena, CA, USA (2008)

[8] Meeds, E., Osindero, S.: An alternative infinite mixture of Gaussian process experts. In Weiss, Y., Schölkopf, B., Platt, J., eds.: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA (2006) 883–890

[9] Rasmussen, C., Ghahramani, Z.: Infinite mixtures of Gaussian process experts. In: Advances in Neural Information Processing Systems, Cambridge, MA, MIT Press (2002)

[10] Vijayakumar, S.: (`http://homepages.inf.ed.ac.uk/∼svijayak/software/LWPR/`)

[11] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation (1991) 79–87

[12] Rasmussen, C.: The infinite Gaussian mixture model. In: Advances in Neural Information Processing Systems 12. MIT Press, Cambridge, MA (2000)

[13] Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)

[14] Pitman, J.: Combinatorial stochastic processes (2002) Notes for Saint Flour Summer School.

[15] Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian data analysis. Chapman & Hall, New York (1995)

[16] Barnett, S.: Matrix Methods for Engineers and Scientists. McGraw-Hill (1979)

[17] Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer (2001)

[18] Fearnhead, P.: Particle filters for mixture models with an unknown number of components. Journal of Statistics and Computing **14** (2004) 11–21

[19] Asuncion, A., Newman, D.: UCI machine learning repository (2007)