Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '07), Hyderabad, India (2007)

Kernel Carpentry for Online Regression using Randomly Varying Coefficient Model

Narayanan U Edakunni [*] **Stefan Schaal** [†] **Sethu Vijayakumar** [*†]
[*] School of Informatics, University of Edinburgh, Edinburgh EH9 3JZ, UK
[†] Department of Computer Science, University Southern California, Los Angeles, CA 90089, USA
n.u.edakunni@sms.ed.ac.uk, sschaal@usc.edu, sethu.vijayakumar@ed.ac.uk

Abstract

We present a Bayesian formulation of locally weighted learning (LWL) using the novel concept of a randomly varying coefficient model. Based on this, we propose a mechanism for multivariate non-linear regression using spatially localised linear models that learns completely *independent* of each other, uses only *local* information and adapts the local model complexity in a data driven fashion. We derive *online* updates for the model parameters based on variational Bayesian EM. The evaluation of the proposed algorithm against other state-of-the-art methods reveal the excellent, robust generalization performance beside surprisingly efficient time and space complexity properties. This paper, for the first time, brings together the computational efficiency and the adaptability of 'non-competitive' locally weighted learning schemes and the modelling guarantees of the Bayesian formulation.

1 Introduction

Locally weighted projection regression (LWPR) [Vijayakumar *et al.*, 2005] is a prime example of recent developments in the area of localised learning schemes that have resulted in powerful non-linear regression algorithms capable of operating in real-time, high dimensional, online learning scenarios. They have been proven to work on many real world applications including, for e.g., supervised learning of sensorimotor dynamics in multiple degree of freedom anthropomorphic robotic systems [Vijayakumar *et al.*, 2002].

All locally weighted schemes (including LWPR) have to determine a region of validity of the local models, i.e., an adaptive local distance metric, in a data driven fashion. This is usually achieved by minimising some sort of cross validation cost on the fit using gradient descent methods. However, the initialization of the local complexity parameter or distance metric, the forgetting factor and the learning rates involved in the gradient method necessitate careful hand tuning of multiple open parameters in existing methods. This may not be trivially achieved in many real world problems with limited prior domain knowledge. Also, there exists no proper probabilistic formulation of the local weighted learning framework – a necessary development in order to exploit the model selection guarantees that Bayesian methods provide while retaining the flexibility provided by nonparametric localised learning.

One of the most attractive characteristics of LWPR-like localised learning schemes is its independent learning rules for each individual local model, which combines or blends the outputs only at the stage of prediction. In addition to avoiding *negative interference* [Schaal and Atkeson, 1998], this property also allows asynchronous learning of local models leading to improved efficiency. We preserve this property in our model by building a generative probabilistic model for each individual local model and derive corresponding learning rules. We show that our novel formulation performs robustly in estimating local model complexity, competes with the state-of-the-art methods in generalization capability, can be extended to learn in *truly* incremental fashion, i.e., without storing data and is surprisingly efficient in both computational complexity and space.

2 Randomly Varying Coefficient model

Modelling spatially localized linear models using a probabilistic framework involves deriving a formulation that allows to model the *fit*, in our case a linear fit, and the *bandwidth* at a particular location in the input space. Each of these local models can then be combined to provide a prediction for a novel data. Additionally, in order for the local models to be independent, each of them should be capable of modelling the entire data by learning the correct bandwidth that partitions the data into two parts – one which corresponds to the linear region of interest and the other which does not. In this paper, we accomplish this by formulating a probabilistic model called Randomly Varying Coefficient(RVC) model which builds upon the idea of a random coefficient model [Longford, 1993].

For a locally linear region centered around $\mathbf{x}_c$ a generative model for the data points can be written as:

$$y_i = \boldsymbol{\beta}_i^T \mathbf{x}_i + \epsilon \tag{1}$$

where $\mathbf{x}_i \equiv [(\mathbf{x}'_i - \mathbf{x}_c)^T, 1]^T$ represents the center subtracted, bias augmented input vector, $\boldsymbol{\beta}_i \equiv [\beta_i^{(1)} \dots \beta_i^{(d+1)}]^T$ represents the corresponding regression coefficient and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian mean zero noise with a standard deviation σ. The data is assumed to have been generated in an

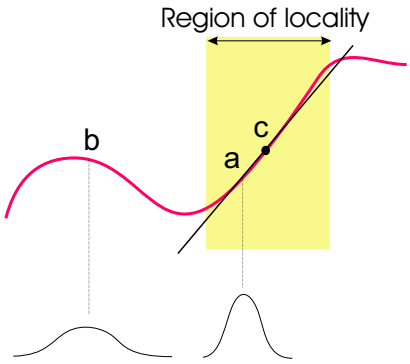


Figure 1: Variation of prior with the location of the input

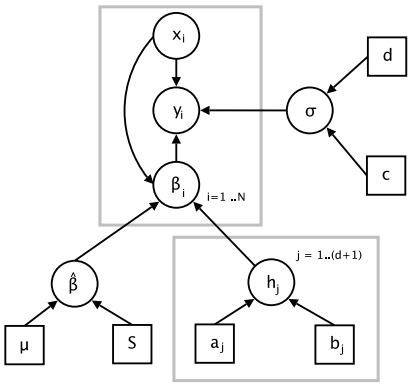


Figure 2: The 'local' generative model

IID fashion. Crucially, we allow the regression coefficient to be a random variable with a prior distribution given by:

$$\boldsymbol{\beta}_i \sim \mathcal{N}(\hat{\boldsymbol{\beta}}, \boldsymbol{C}_i) \tag{2}$$

where we have assumed that each $\boldsymbol{\beta}_i$ is generated from a Gaussian centered around $\hat{\boldsymbol{\beta}}$ with the confidence being represented by the covariance $\mathbf{C}_i$. The covariance itself is defined to be proportional to the distance of $\mathbf{x}_i'$ from the center. This has the effect that for points that lie close to the center, the distribution of $\boldsymbol{\beta}_i$ is peaked around $\hat{\boldsymbol{\beta}}$ resulting in a linear region around the center. This has been illustrated schematically in Fig. 1 where point *c* is the center of the local model: for a point *a* that lies close to *c* we assign a prior that is fairly tight around the mean whereas for a point *b* that lies away from *c* the prior is much broader. One can consider various distance functions to index the variation of the covariance matrix $\mathbf{C}$. Here, we restrict ourselves to a diagonal version, each diagonal element varying quadratically with $\mathbf{x}$ as:

$$\mathbf{C}_i(j,j) = ((\mathbf{x}_i' - \mathbf{x}_c)^T(\mathbf{x}_i' - \mathbf{x}_c) + 1)/h_j^2 = \mathbf{x}_i^T\mathbf{x}_i/h_j^2 \tag{3}$$

where h_j is the *bandwidth* parameter of the kernel, which defines the extent of the locality along the j-th dimension. This choice of the kernel parametrization allows us to use a conjugate Gamma prior over h_j. The higher values of h_j imply lesser variation amongst the coefficients $\boldsymbol{\beta}_i$ and hence, larger regions of linearity. Although the bandwidth modulates the bias-variance tradeoff, an unconstrained likelihood maximization will, in general favor large h_j since it implies a higher confidence over larger regions of the data. Therefore, we use a Gamma *regularizer* prior over the bandwidth parameters such that it favors relatively small values of h_j leading to more localised models:

$$h_j^2 \sim \mathit{Gamma}(a_j, b_j) \tag{4}$$

We shall further assign noninformative Normal prior $\mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$ for the parameter $\hat{\boldsymbol{\beta}}$ and a noninformative inverse Gamma prior with hyperparameters c and d for σ. We assume a uniform prior for the regularizer hyperparameters a_j and b_j. Fig. 2 summarizes the resultant probabilistic model for a *single* local model.

In this model, one can marginalize out the hidden variables $\boldsymbol{\beta}_i$ to obtain

$$\begin{aligned} &P(y_i|\hat{\boldsymbol{\beta}}, \sigma, h_1 \ldots h_{d+1}) = \int P(y_i|\boldsymbol{\beta}_i^T\mathbf{x}_i, \sigma^2)P(\boldsymbol{\beta}_i|\hat{\boldsymbol{\beta}}, \boldsymbol{C}_i)d\boldsymbol{\beta}_i \\ &\Rightarrow y_i \sim \mathcal{N}(\hat{\boldsymbol{\beta}}^T\mathbf{x}_i, \mathbf{x}_i^T\boldsymbol{C}_i\mathbf{x}_i + \sigma^2) \end{aligned} \tag{5}$$

It is interesting to note that the form of likelihood in Eq. (5) corresponds to a heteroscedastic regression and will be used in later sections for prediction. In the next section we deal with computing the parameter updates and the resultant ensemble posteriors in an efficient manner.

3 Learning

Our objective is to learn the posterior over the parameters $\hat{\boldsymbol{\beta}}$, h_j, σ and to obtain point estimates for the hyperparameters – a_j, b_j. The joint posterior is given by:

$$P(\mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma|\mathbf{y}, \mathbf{a}, \mathbf{b}, c, d, \boldsymbol{\mu}, \mathbf{S}) = \frac{P(\mathbf{y}, \hat{\boldsymbol{\beta}}, \mathbf{h}, \sigma, \mathbf{a}, \mathbf{b}, c, d, \boldsymbol{\mu}, \mathbf{S})}{P(\mathbf{y}, \mathbf{a}, \mathbf{b}, c, d, \boldsymbol{\mu}, \mathbf{S})} \tag{6}$$

where we have used $\mathbf{h}$ to denote the vector $[h_1^2 \ldots h_{d+1}^2]^T$ and $\mathbf{y}$ denotes the training data $[y_1 \ldots y_N]^T$, $\mathbf{a} \equiv [a_1 \ldots a_{d+1}]^T$ and $\mathbf{b} \equiv [b_1 \ldots b_{d+1}]^T$. However, the posterior over the parameters is rendered intractable due to the difficulty in evaluating the denominator of Eq. (6). This necessitates the use of variational Bayesian EM to evaluate the posterior $P(\mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma|\mathbf{y}, \mathbf{a}, \mathbf{b}, c, d, \boldsymbol{\mu}, \mathbf{S})$ and learn the regulariser hyperparameters $\mathbf{a}$ and $\mathbf{b}$.

3.1 Variational approximation

To learn the parameters of the model we can maximize the marginal log likelihood with respect to the parameters treating $\boldsymbol{\beta}_i$ as the hidden variables. The marginal log likelihood is given by:

$$\begin{aligned} \mathcal{L} &= \ln P(\mathbf{y}|\mathbf{a}, \mathbf{b}, c, d, \boldsymbol{\mu}, \mathbf{S}) \\ &= \ln \int P(\mathbf{y}, \boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma|\mathbf{a}, \mathbf{b}, \boldsymbol{\mu}, \mathbf{S}, c, d)d\boldsymbol{\beta}_1 \ldots d\boldsymbol{\beta}_N \\ &\quad d\mathbf{h}d\hat{\boldsymbol{\beta}}d\sigma \\ &= \ln \int \left[\prod_i P(y_i|\boldsymbol{\beta}_i, \sigma)P(\boldsymbol{\beta}_i|\hat{\boldsymbol{\beta}}, h_1, \ldots h_{d+1}) \right. \\ &\quad \left. \prod_j P(h_j^2|a_j, b_j)P(\hat{\boldsymbol{\beta}}|\boldsymbol{\mu}, \mathbf{S})P(\sigma^2|c, d)\right] d\boldsymbol{\beta}_1 \ldots d\boldsymbol{\beta}_N \\ &\quad dh_1 \ldots dh_{d+1}d\hat{\boldsymbol{\beta}}d\sigma \end{aligned} \tag{7}$$

Using Jensen's inequality, the objective function that lower bounds $\mathcal{L}$ is given by:

$$\mathcal{F} = \int \Big[Q(\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma^2) \ln \frac{P(\mathbf{y}, \boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma^2 | \mathbf{a}, \mathbf{b}, \boldsymbol{\mu}, \mathbf{S}, c, d)}{Q(\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma^2)} \Big] d\boldsymbol{\beta}_1 \dots d\boldsymbol{\beta}_N d\mathbf{h} d\hat{\boldsymbol{\beta}} d\sigma^2 \tag{8}$$

The optimal value for $Q(\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma)$ that makes the bound tight is given by the joint posterior $P(\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma | \mathbf{y})$ but since this posterior is intractable, we make an approximation by assuming that the posterior over the variables is independent and can be expressed as $Q(\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_N, \mathbf{h}, \hat{\boldsymbol{\beta}}, \sigma) = \prod_i Q(\boldsymbol{\beta}_i | \mathbf{y}) \prod_j Q(h_j^2 | \mathbf{y}) Q(\hat{\boldsymbol{\beta}} | \mathbf{y}) Q(\sigma^2 | \mathbf{y})$. This form of approximation is often called an *ensemble* variational approximation, details of which can found in [Beal, 2003]. Substituting the factorised approximation in Eq. (8) we get:

$$\begin{aligned} \mathcal{F}_{approx} = & \sum_i \Big[\langle \ln P(y_i | \boldsymbol{\beta}_i, \sigma) \rangle_{Q_{\beta_i}, Q_{\sigma^2}} \\ & + \Big\langle \ln P(\boldsymbol{\beta}_i | \hat{\boldsymbol{\beta}}, h_1 \dots h_{d+1}) \Big\rangle_{Q_{\beta_i}, Q_{h_1} \dots Q_{h_{d+1}}, Q_{\hat{\beta}}} \Big] \\ & + \sum_j \big\langle \ln P(h_j^2 | a_j, b_j) \big\rangle_{Q_{h_j}} + \Big\langle \ln P(\hat{\boldsymbol{\beta}} | \boldsymbol{\mu}, \mathbf{S}) \Big\rangle_{Q_{\hat{\beta}}} \\ & + \big\langle \ln P(\sigma^2 | c, d) \big\rangle_{Q_{\sigma^2}} - \sum_i \Big\langle \ln Q_{\boldsymbol{\beta}_i} \Big\rangle_{Q_{\beta_i}} \\ & - \sum_j \big\langle \ln Q_{h_j} \big\rangle_{Q_{h_j}} - \Big\langle \ln Q_{\hat{\boldsymbol{\beta}}} \Big\rangle_{Q_{\hat{\beta}}} - \langle \ln Q_{\sigma^2} \rangle_{Q_{\sigma^2}} \end{aligned} \tag{9}$$

where $\langle . \rangle_Q$ denotes the expectation with respect to the distribution Q. The optimal values of the posterior probabilities can be computed iteratively by maximizing the functional $\mathcal{F}_{approx}$ with respect to each individual posterior distribution keeping the other distributions fixed akin to an EM procedure. Such a procedure can be shown to improve our factorised approximation of the actual posterior in each iteration. Skipping the derivation, such a procedure yields the following posterior distributions:

$$Q(\boldsymbol{\beta}_i | \mathbf{y}) \sim \mathcal{N}(\boldsymbol{\nu}_i, \mathbf{G}_i) \tag{10}$$

$$Q(\hat{\boldsymbol{\beta}} | \mathbf{y}) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{S}}) \tag{11}$$

$$Q(h_j^2 | \mathbf{y}) \sim \textit{Gamma}(\tilde{a}_j, \tilde{b}_j) \tag{12}$$

$$Q(\sigma^2 | \mathbf{y}) \sim \textit{Inv-Gamma}(\tilde{c}, \tilde{d}) \tag{13}$$

where

$$\begin{aligned} \mathbf{G}_i &= (\mathbf{x}_i \mathbf{x}_i^T / \langle \sigma^2 \rangle + \langle \boldsymbol{C}_i \rangle^{-1})^{-1} \\ &= \langle \boldsymbol{C}_i \rangle - \frac{\langle \boldsymbol{C}_i \rangle \mathbf{x}_i \mathbf{x}_i^T \langle \boldsymbol{C}_i \rangle}{\langle \sigma^2 \rangle + \mathbf{x}_i^T \langle \boldsymbol{C}_i \rangle \mathbf{x}_i} \end{aligned} \tag{14}$$

where the second part has been derived by making use of the Sherman-Morrison Woodbury theorem. Here $\langle \mathbf{C}_i \rangle = diag(\mathbf{x}_i^T \mathbf{x}_i / \langle h_j^2 \rangle_{Q(h_j^2)})$ and $\langle \sigma^2 \rangle$ is the expectation with respect to Q_{σ^2}. Furthermore, using results from Eq. (14),

$$\begin{aligned} \boldsymbol{\nu}_i &= \mathbf{G}_i (y_i \mathbf{x}_i / \langle \sigma^2 \rangle + \langle \boldsymbol{C}_i \rangle^{-1} \tilde{\boldsymbol{\mu}}_i) \\ &= \frac{\langle \boldsymbol{C}_i \rangle \mathbf{x}_i}{(\langle \sigma^2 \rangle + \mathbf{x}_i^T \langle \boldsymbol{C}_i \rangle \mathbf{x}_i)} \Big(y_i - \mathbf{x}_i^T \tilde{\boldsymbol{\mu}}_i \Big) + \tilde{\boldsymbol{\mu}}_i \end{aligned} \tag{15}$$

$$\tilde{\mathbf{S}} = (\sum_i \langle \mathbf{C}_i \rangle^{-1} + \mathbf{S}^{-1})^{-1}, \tag{16}$$

$$\tilde{\boldsymbol{\mu}} = \tilde{\mathbf{S}} (\sum_i \langle \mathbf{C}_i \rangle^{-1} \boldsymbol{\nu}_i + \mathbf{S}^{-1} \boldsymbol{\mu}) \tag{17}$$

$$\tilde{a}_j = a_j + N/2 \tag{18}$$

$$\tilde{b}_j = b_j + \sum_i \Big[(\boldsymbol{\nu}_{i,j} - \tilde{\boldsymbol{\mu}}_{i,j})^2 + \mathbf{G}_{i,jj} + \tilde{\mathbf{S}}_{jj} \Big] / (2 \mathbf{x}_i^T \mathbf{x}_i) \tag{19}$$

Here, $\boldsymbol{\nu}_{i,j}$ and $\tilde{\boldsymbol{\mu}}_{i,j}$ denote the j-th element of the respective vectors and $\mathbf{G}_{i,jj}$ and $\tilde{\mathbf{S}}_{jj}$ denotes the j-th diagonal element.

$$\tilde{c} = c + N/2 \tag{20}$$

$$\tilde{d} = d + \sum_i \Big[(y_i - \boldsymbol{\nu}_i^T \mathbf{x}_i)^2 + \mathbf{x}_i^T \mathbf{G}_i \mathbf{x}_i \Big] / 2 \tag{21}$$

We also need to learn the point estimates for the regulariser hyperparameters a_j and b_j. Maximum likelihood value for the hyperparameters a_j and b_j can be found by maximizing the bound $\mathcal{F}_{approx}$ given by Eq. (9) with respect to these hyperparameters keeping the posterior distributions Q fixed. Considering only the terms involving the hyperparameters:

$$\mathcal{E} = \int Q(h_j^2 | \tilde{a}_j, \tilde{b}_j) \ln P(h_j^2 | a_j, b_j) dh_j^2$$

Maximising $\mathcal{E}$ with respect to the hyperparameters is equivalent to minimising the KL divergence between the distributions Q and P. Since the posterior Q and prior P share the same parametric form, KL divergence is minimised when the parameters of these distributions match. This leads to the simple update rule for the hyperparameters given by:

$$a_j = \tilde{a}_j, \quad b_j = \tilde{b}_j \tag{22}$$

The hyperparameters $\boldsymbol{\mu}$, $\mathbf{S}$, c and d are initialised such that the corresponding priors are non-informative. An initialisation of $\boldsymbol{\mu} = \mathbf{0}$, $\mathbf{S} = 10^{-3} \times \mathbf{I}$, $c = 10^{-3}$ and $d = 10^{-3}$ ensures such a condition. On the other hand the regulariser hyperparameters $\mathbf{a}$ and $\mathbf{b}$ are initialised such that it encourages small $\mathbf{h}$. A value of $\mathbf{a} = 1$ and a sufficiently large value for $\mathbf{b}$ ensures such a bias. These are the settings used by RVC for all the evaluations carried out in Sec. 4.

3.2 Prediction using the committee of local models

We have dealt so far with building a coherent probabilistic model for each local expert and have derived inference procedures to estimate the parameters of individual model. Given the ensemble of trained local experts, in order to predict the response y_q for a new query point $\mathbf{x}_q$, we take the normalised product of the *predictive distribution* of each local expert. This is close in spirit to the paradigm of Product of Experts [Hinton, 1999] and the Bayesian Committee Machines [Tresp, 2000]. The predictive distribution of each local expert is given by:

$$P(y_q | \mathbf{y}) = \int P(y_q | \hat{\boldsymbol{\beta}}, \sigma, \mathbf{h}) Q(\hat{\boldsymbol{\beta}} | \mathbf{y}) Q(\sigma^2 | \mathbf{y}) Q(\mathbf{h} | \mathbf{y}) d\mathbf{h} d\hat{\boldsymbol{\beta}} d\sigma^2 \tag{23}$$

where $P(y_q | \hat{\boldsymbol{\beta}}, \sigma, \mathbf{h})$ has the form given by Eq. (5). We can further integrate out $\hat{\boldsymbol{\beta}}$ from Eq. (23), but cannot do the same for σ^2 and $\mathbf{h}$. Hence, we approximate $Q(\sigma^2 | \mathbf{y})$ and $Q(\mathbf{h} | \mathbf{y})$ by a delta function at the mode which implies

$Q(\sigma^2|\mathbf{y}) \approx \delta_{\sigma^2_{mode}}$ and $Q(\mathbf{h}|\mathbf{y}) \approx \delta_{\mathbf{h}^2_{mode}}$. The final predictive distribution for the k-th local model is:

$$y_{q,k} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}^T \mathbf{x}_{q,k}, \mathbf{x}_{q,k}{}^T(\tilde{\mathbf{S}}_k + \boldsymbol{C}_{k\mathbf{h}_{mode}})\mathbf{x}_{q,k} + \sigma^2_{mode})$$

where $\mathbf{x}_{q,k}$ refers to the query point with the k-th center subtracted and augmented with bias. Blending the prediction of different experts by taking their product and normalising it results in a Normal distribution given by:

$$y_q \sim \mathcal{N}(\mu, \zeta^2) \quad \text{where} \quad \mu = \frac{\sum_k \alpha_k \tilde{\boldsymbol{\mu}}_k^T \mathbf{x}_{q,k}}{\sum_k \alpha_k}, \;\; \zeta^2 = \frac{1}{\sum_k \alpha_k}.$$

Here, μ is a sum of the means of each individual expert weighted by the confidence expressed by each expert in its own prediction α_k, ζ^2 is the variance and α_k is the precision of each expert:

$$\alpha_k = 1/(\mathbf{x}_{q,k}^T(\tilde{\mathbf{S}}_k + \boldsymbol{C}_k)\mathbf{x}_{q,k} + \sigma_k^2), \;\; \boldsymbol{C}_k = diag\{\mathbf{x}_{q,k}^T \mathbf{x}_{q,k}/h_{j,k}^2\}$$

3.3 Online updates

The iterative learning rules to estimate the posteriors over parameters given the appropriate prior and the data, represented by Eqs. (16)-(21), can be rewritten in the form of online updates by exploiting the Bayesian formalism. In a batch mode of posterior evaluation, we have

$$\textit{posterior}_N = \prod_i^N (\textit{likelihood}_i) \times \textit{prior}_0$$

The same can be expressed as a set of online updates:

$$\textit{posterior}_i = \textit{likelihood}_i \times prior_i; \quad \textit{prior}_{i+1} = \textit{posterior}_i$$

Therefore we can transform the batch updates that we had derived earlier into online updates given by :

$$\tilde{\mathbf{S}}_i = (\langle \mathbf{C}_i \rangle^{-1} + \mathbf{S}_i^{-1})^{-1} \tag{24}$$

$$\tilde{\boldsymbol{\mu}}_i = \tilde{\mathbf{S}}_i(\langle \mathbf{C}_i \rangle^{-1} \boldsymbol{\nu}_i + \mathbf{S}_i^{-1} \boldsymbol{\mu}_i) \tag{25}$$

$$\tilde{a}_{i,j} = a_{i,j} + 1/2 \tag{26}$$

$$\tilde{b}_{i,j} = b_{i,j} + \left[(\boldsymbol{\nu}_{i,j} - \tilde{\boldsymbol{\mu}}_{i,j})^2 + \mathbf{G}_{i,jj} + \tilde{\mathbf{S}}_{i,jj}\right]/(2\mathbf{x}_i^T \mathbf{x}_i) \tag{27}$$

$$\tilde{c}_i = c_i + 1/2 \tag{28}$$

$$\tilde{d}_i = d_i + \left[(y_i - \boldsymbol{\nu}_i^T \mathbf{x}_i)^2 + \mathbf{x}_i^T \mathbf{G}_i \mathbf{x}_i\right]/2 \tag{29}$$

We repeat above updates for a single data point $\{\mathbf{x}_i, y_i\}$ till the posteriors converge – here, $\tilde{\Theta}$ represents the posterior of Θ. For the $(i+1)$-th point, we then use posterior of i-th step as the prior as illustrated in Algorithm 1.

Addition/deletion of local models

The complexity of the learner is adapted by the addition and deletion of local models. When the predictive likelihood for a new data point is sufficiently low then one can conclude that the complexity of the learner needs to be increased by adding a new local model. This leads to the simple heuristic for the addition of a local model wherein a local model is added at a data point when the predictive probability for the particular training data is less than a fixed threshold. The data point serves as the center for the added local model.

When two local models have sufficient overlap in the region they model, then one of them is redundant and can be pruned. The overlap between two local models can be determined by the difference in the confidence expressed in their prediction for a common test point. The addition and deletion heuristics that have been used here is similar to the ones used in [Schaal and Atkeson, 1998].

Algorithm 1 Training a local model

1: Initialise hyperparameters: $\Theta_0 \equiv \{\boldsymbol{\mu}_0, \mathbf{S}_0, c_0, d_0, \mathbf{a}_0, \mathbf{b}_0\}$.
2: **for** $i = 1$ to N **do**
3: Input $\mathbf{x}_i$, y_i
4: **repeat**
5: Estimate posterior hyperparameters $\tilde{\Theta}_i$ using Θ_i and Eq. (14), (15) and Eqs. (24) - (29).
6: Estimate values of the hyperparameters $\mathbf{a}$ and $\mathbf{b}$ of the regulariser prior using Eq. (22).
7: **until** convergence of posteriors
8: $\Theta_{i+1} = \tilde{\Theta}_i$
9: **end for**

3.4 Complexity analysis

The time complexity of the algorithm is dominated by the computation of $\mathbf{G}_i$ in Eq. (14). The equations that use $\mathbf{G}_i$ are Eq. (27) and Eq. (29) and these can be rewritten to avoid explicit computation of $\mathbf{G}_i$. Eq. (27) requires only the diagonal elements of $\mathbf{G}_i$ which can be computed in $O(d)$ since

$$\mathbf{G}_i(j,j) = \mathbf{C}_i(j,j) - (\mathbf{C}_i(j,j)\mathbf{x}_i(j))^2/(\sigma^2 + \gamma_i) \quad \text{using Eq. (14)}$$

where $\gamma_i = \mathbf{x}_i^T \mathbf{C}_i \mathbf{x}_i$ which can also be computed in $O(d)$ due to the fact that $\mathbf{C}_i$ is diagonal. On the other hand, Eq. (29) requires the evaluation of $\mathbf{x}_i^T \mathbf{G}_i \mathbf{x}_i$ which in turn can be written down as:

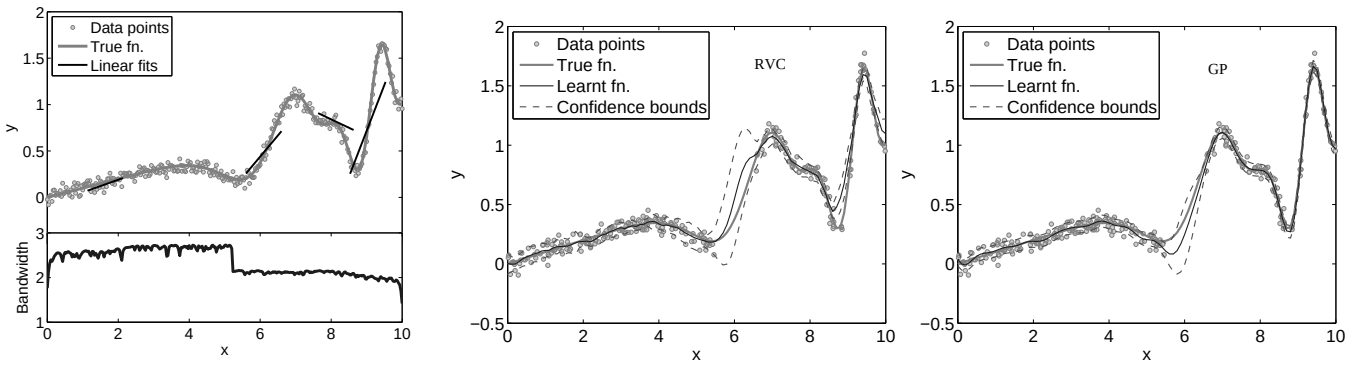
$$\mathbf{x}_i^T \mathbf{G}_i \mathbf{x}_i = \frac{\sigma^2 \gamma_i}{\sigma^2 + \gamma_i}$$

and can also be computed in $O(d)$. Furthermore, the matrix inverses in Eq. (24) and Eq. (25) can also be computed in $O(d)$ due to the fact that $\mathbf{S}_i$ and $\mathbf{C}_i$ are diagonal matrices. Therefore the overall time complexity per online update is $O(dM)$ where d is the number of dimensions and M the number of local models. The algorithm doesn't require any data points to be stored and hence, has a $O(M)$ space complexity for the sufficient statistics stored in the local models. The independence of the local models also means that we could bring down the effective time complexity to $O(d)$ if we had M parallel processors. The time complexity for prediction is $O(dM)$ including the evaluation of mean and the confidence bounds. We can see from this analysis that the algorithm is very efficient with respect to time and space (in fact it matches LWPR's efficiency) and hence, is a strong candidate for situations which require real time and online learning.

4 Evaluation

In this section, we demonstrate the salient aspects of the RVC model by looking at some empirical test results, compare the accuracy and robustness against state of the art methods and evaluate its performance on some benchmark datasets.

Fig. 3(a) shows the local linear fits (at selected test points) learned by RVC from noisy training data on a function with varying spatial complexity. Such functions are extremely hard to learn since models with high bias tends to oversmooth the nonlinear regions while more complex models tend to fit the noise. One can see that the linear fit roughly corresponds to the tangential line at the center of each local model as expected. A more significant result is the adaptation of the local



(a) (b) (c)

Figure 3: (a)Local fits and bandwidth adaptation. Fit and confidence bounds learned by (b) RVC model and by (c) GP model.

bandwidth. The bottom section of Fig. 3(a) plots the converged locality measure computed as *product* of the bandwidth parameters along each input dimension - nicely illustrating the ability to adapt the local complexity parameter in a data driven manner. Note that for this illustration, we have placed local centers in a dense, uniform grid in input space.

In the next evaluation, using the same $sinc$ function, we compare the fits and confidence bounds learned by RVC and Gaussian Processes (GP) [Williams, 1998] in Fig. 3(b) and (c). It is important to note that we have deliberately avoided using training data in [5.5,6.5] and the confidence bounds of RVC nicely reflect this. Our next experiment aims to illustrate the ability of RVC to model heteroscedastic data (i.e., data with varying noise levels). Fig. 4 illustrates the fit and the confidence interval learnt on the *motorcycle impact* data discussed in [Rasmussen and Gharamani, 2000]. Notice that the confidence interval correctly adapts to the varying amount of noise in the data as compared to the confidence interval learnt by a GP with squared exponential kernel shown in Fig. 5. This ability to model non-stationary functions is another advantage of RVC's localised learning. In these evaluations, we have used RVC in the batch mode using the updates that we derived in Sec. 3(c.f. Eqs. 14 - 21). The subsequent evaluations in this section make use of the online updates derived in Sec. 3.3.

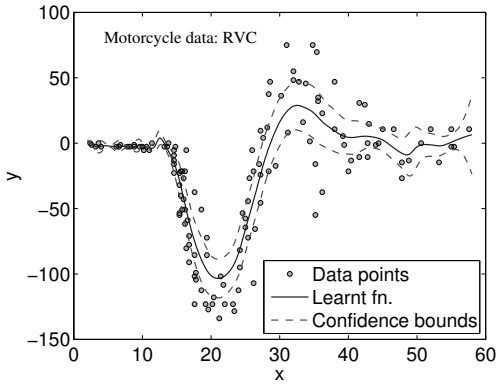


Figure 4: Fit and confidence bounds for the motorcycle dataset learned by the RVC model (local models were centered at 20 uniformly distributed points along the input)

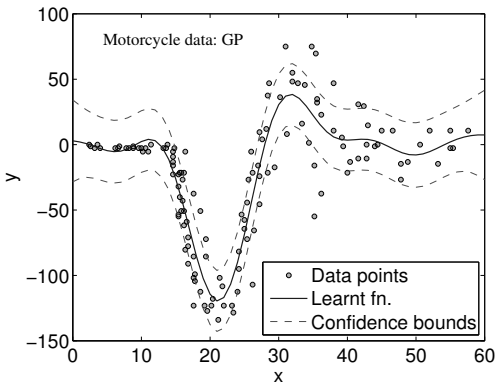


Figure 5: Fit and confidence bounds for the motorcycle dataset learned by the Gaussian Processes model

To compare the online learning characteristics, we trained the three candidate algorithms on 500 data points from the $sinc$ function corrupted with output noise: $\epsilon \sim \mathcal{N}(0, 0.05^2)$. After each training data was presented to the learner, the error in learning was measured using a set of 1000 uniformly distributed test points. The RVC model was allowed only a single EM iteration for each data point to ensure a fair comparison with LWPR. The resulting error dynamics is shown in Fig. 6(a). In this comparison, GP exhibits a sharply decreasing error curve which is not surprising considering that it is essentially a *batch* method and stores away all of the training data for prediction. When we compare RVC with LWPR, we find that RVC converges faster while using roughly similar number of local models. This can be attributed to the Bayesian learning rules of RVC that estimates the posterior over parameters rather than point estimates. Since the posterior is a product of likelihood and prior, in the event of sparse data (as in the initial stages of online learning), the prior ensures that the posterior distributions assigned to the parame-

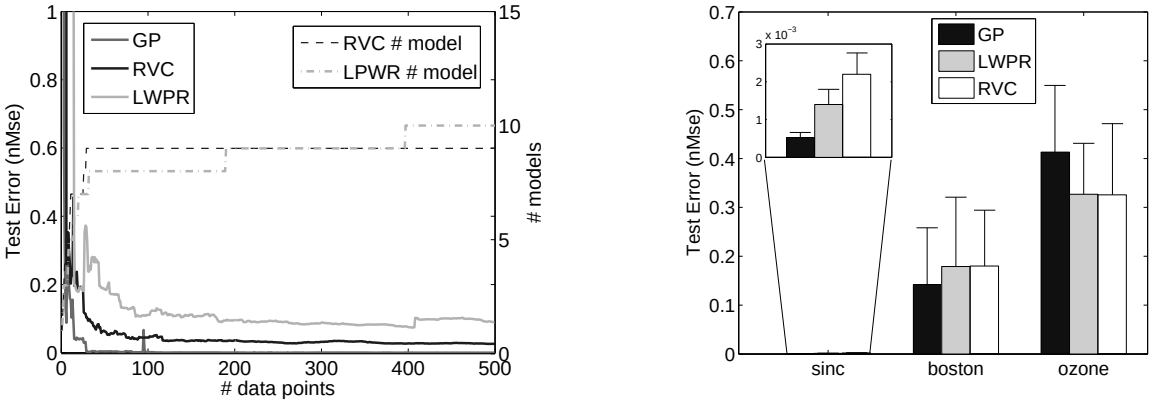


Figure 6: (a) Comparison of online learning dynamics for $sinc$ function (b) Comparison of generalization error

ters and in turn the predictions of the learner are reasonable. Also the optimization of the regularizer hyperparameters for every data point implies a faster adaptation and hence, a faster convergence.

In the next evaluation, we compare the generalization performance of the algorithms on artificial as well as real world datasets. The $sinc$ function, air dataset described in [Bruntz *et al.*, 1974] and the Boston housing dataset from the UCI repository were used as benchmark datasets. The air(ozone) dataset which is a three dimensional dataset with 111 data points was split into 83 training and 28 test points. The 13 dimensional Boston dataset was split into 404 training and 102 test points. The online learners namely RVC and LWPR were trained in epochs of repeated presentation of the training data, till convergence – LWPR required careful tuning of the distance metric initialization and learning rates to achieve the performance reported here as opposed to the uninformative priors used for RVC. The performance of GP, RVC and LWPR shown in Fig. 6(b) are statistics accumulated over 10 different train-test splits. All three methods perform very well on the $sinc$ data set, achieving $nMSE$ of less than 0.0025. For the ozone dataset which is highly nonlinear, RVC and LWPR performs better than GP. For the Boston dataset, we find that the performance of RVC is close to that of LWPR while slightly inferior to the GP results – although this difference is statistically insignificant.

5 Discussion

The major contribution of the paper is the development of a Bayesian formulation for independent spatially localised learners for multivariate nonlinear regression. We have used a novel formulation of data dependent priors in order to carve out locally linear regions while avoiding competition amongst local models. The 'non-competitive' behaviour of each local model allows independent, efficient learning while the Bayesian regularizer hyperpriors guard against the danger of overfitting or over-smoothing through automatic local bandwidth adaptation. We evaluated the RVC model against the state of the art in non-linear regression techniques on artificial as well as real world data sets. RVC matched the generalization performance of LWPR while avoiding cumbersome parameter tuning for initialization. It achieves competitive performance compared to GP – essentially a batch method, while being much more computationally efficient (linear in number of training data and input dimensionality as opposed to cubic in training data for GP). The space and computational efficiency of RVC coupled with the ability to grow model complexity in a data driven fashion makes it a strong candidate for practical online and real time learning scenarios.

References

[Beal, 2003] M.J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[Bruntz *et al.*, 1974] S.M. Bruntz, W.S. Cleveland, B. Kleiner, and J.L. Warner. The dependence of ambient ozone on solar radiation, temperature and mixing height. In *Symp. Atmospheric Diffusion and Air pollution*. MIT Press, 1974.

[Hinton, 1999] G.E. Hinton. Product of experts. In *Ninth International Conference on Artificial Neural Networks*, 1999.

[Longford, 1993] N.T. Longford. *Random coefficient models*. Clarendon Press, 1993.

[Rasmussen and Gharamani, 2000] C.E. Rasmussen and Z. Gharamani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2000.

[Schaal and Atkeson, 1998] S. Schaal and C.G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1998.

[Tresp, 2000] V. Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[Vijayakumar *et al.*, 2002] S. Vijayakumar, A. D'Souza, T. Shibata, J. Conradt, and S. Schaal. Statistical learning for humanoid robots. *Autonomous Robot*, 12(1):55–69, 2002.

[Vijayakumar *et al.*, 2005] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17(12), 2005.

[Williams, 1998] C.K.I. Williams. Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 599–621. Kluwer, 1998.