RADIAL BASIS FUNCTION REGRESSION USING TRANS-DIMENSIONAL SEQUENTIAL MONTE CARLO

J. Vermaak, S. J. Godsill, A. Doucet

ABSTRACT

We consider the general problem of sampling from a sequence of distributions that is defined on a union of subspaces. We will illustrate the general approach on the problem of sequential radial basis function (RBF) regression where the number of kernels is variable and unknown. Our approach, which we term Trans-Dimensional Sequential Monte Carlo (TD-SMC), is based on a generalisation of importance sampling to spaces of variable dimension. In the spirit of [1] we augment the target parameter space at the current time step with an auxiliary space corresponding to the parameters at the previous time step. This facilitates the design of efficient proposal distributions, which can then be formulated as moves from the auxiliary parameter space to the target parameter space, lending our algorithm its sequential character. These proposals are very general, and may include within model moves to update parameters, and trans-dimensional birth or death moves to add or remove parameters when appropriate. From this perspective our approach is reminiscent of the Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) algorithm [2].

1. RBF REGRESSION

Regression with radial basis functions is concerned with fitting a mixture of local kernels to some unknown function, based on noisy samples from the target function. In most applications of interest the number of kernels required is unknown. Existing strategies to estimate the number of kernels alongside the other parameters are batch algorithms. Here we will develop a sequential strategy that achieves the same purpose by a single pass through the data.

The RBF regression function for the uncorrupted data takes the form

$$z_t = \beta_0 + \sum_{i=1}^{k} \beta_i K(\mathbf{x}_t, \boldsymbol{\mu}_i),$$

where $\mathbf{x}_t \in \mathbb{R}^d$ and $z_t \in \mathbb{R}$ denote the input variables and uncorrupted output, respectively, $\boldsymbol{\beta} = (\beta_0 \cdots \beta_k) \in \mathbb{R}^{k+1}$ is the vector of regression coefficients, and $K(\cdot, \boldsymbol{\mu})$ is a local kernel function centred on $\boldsymbol{\mu} \in \mathbb{R}^d$. The data is assumed to be corrupted with *i.i.d.* Gaussian noise, *i.e.*

$$y_t = z_t + v_t, \quad v_t \sim \mathcal{N}(0, \sigma_y^2),$$

where σ_y^2 is the observation noise variance. For the estimation we assume the availability of a static data set of input/output pairs $\{\mathbf{x}_t, y_t\}_{t=1}^T$. Note that the ordering of the data points may be arbitrary.

The unknown parameters to be estimated are the regression coefficients and the kernel centres, *i.e.* $\boldsymbol{\theta}_{1:k} = (\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_k)$, with $\boldsymbol{\theta}_i = (\beta_i, \boldsymbol{\mu}_i)$. As is common in RBF regression problems we assume the support for the kernel centres at time t to be the input data points available at time t. The support for $\boldsymbol{\theta}_{1:k}$ can thus be written as $\boldsymbol{\Theta}_{k,t} = (\mathbb{R} \times \{\mathbf{x}_1 \cdots \mathbf{x}_t\})^k$. We define the prior distribution for the unknown parameters as

$$p_t(k, \boldsymbol{\theta}_{1:k}) = p(k)p(\boldsymbol{\beta}_0) \prod_{i=1}^{k} p(\beta_i)p_t(\boldsymbol{\mu}_i), \qquad (1)$$

with

$$\begin{aligned} p(k) &\propto \lambda^k \exp(-\lambda)/k!, && k \in \{1 \cdots k_{\max}\} \\ p(\beta_i) &= \mathcal{N}(\beta_i | 0, \sigma_\beta^2), && i = 0 \cdots k \\ p_t(\boldsymbol{\mu}_i) &= \sum_{s=1}^{t} \delta_{\mathbf{x}_s}(\boldsymbol{\mu}_i)/t, && i = 1 \cdots k. \end{aligned}$$

The prior parameters $(\lambda, \sigma_\beta^2)$ are here assumed to be fixed and known. It is, however, possible to define further hyperpriors on these parameters and estimate them alongside the other unknowns.

Given the conditional independence assumption on the data points and the definition of the prior it is straightforward to obtain an expression for the full posterior $p_t(k, \boldsymbol{\theta}_{1:k} | \mathbf{y}_{1:t})$. Due to conjugacy the posterior can be marginalised over the regression coefficients. The resulting marginal posterior over the kernel centres can be written as

$$p_t(k, \boldsymbol{\mu}_{1:k} | \mathbf{y}_{1:t}) \propto \frac{|\mathbf{B}|^{1/2} \exp(-\mathbf{y}^{\mathrm{T}}\mathbf{P}\mathbf{y}/2\sigma_y^2) p(k) p_t(\boldsymbol{\mu}_{1:k})}{(2\pi\sigma_y^2)^{t/2} (\sigma_\beta^2)^{k+1/2}}, \qquad (2)$$

with $\mathbf{B} = (\mathbf{K}^{\mathrm{T}}\mathbf{K}/\sigma_y^2 + \mathbf{I}_{k+1}/\sigma_\beta^2)^{-1}$ and $\mathbf{P} = \mathbf{I}_t - \mathbf{K}\mathbf{B}\mathbf{K}^{\mathrm{T}}/\sigma_y^2$. In the above $\mathbf{y} \in \mathbb{R}^t$ is the column vector comprising the t output data points, and $\mathbf{K} \in \mathbb{R}^{t \times (k+1)}$ denotes the kernel matrix, with row s given by $\mathbf{K}_s = (1, K(\mathbf{x}_s, \boldsymbol{\mu}_1) \cdots K(\mathbf{x}_s, \boldsymbol{\mu}_k))$. For any estimate of the kernel centres an estimate of the clean output data points can be obtained as $\widehat{\mathbf{z}} = \mathbf{K}\mathbf{B}\mathbf{K}^{\mathrm{T}}/\sigma_y^2$, without the need to explicitly compute the regression coefficients.

Note that even though we consider the problem of sequential RBF regression here, our algorithm is applicable to the sequential estimation of any distribution $p_t(k, \boldsymbol{\theta}_{1:k}|\mathbf{y}_{1:t})$ that can be evaluated up to a normalising constant. The space over which the parameters are defined is allowed to be of variable dimension and evolve over time.

2. TRANS-DIMENSIONAL SEQUENTIAL MONTE CARLO

Our aim is to generate samples for a Monte Carlo approximation to the target posterior $p_t(k, \boldsymbol{\theta}_{1:k}|\mathbf{y}_{1:t})$. Designing an efficient proposal distribution to generate samples directly in the target parameter space is difficult. This is mostly due to the fact that the dimension of the parameter space is generally high and variable. To circumvent these problems we augment the target parameter space with an auxiliary parameter space, which we will later associate with the parameters at the previous time step. The target distribution over the resulting joint space is defined as

$$\pi_t(k, \boldsymbol{\theta}_{1:k}; k', \boldsymbol{\theta}'_{1:k'}) = p_t(k, \boldsymbol{\theta}_{1:k}|\mathbf{y}_{1:t}) q'_t(k', \boldsymbol{\theta}'_{1:k'}|k, \boldsymbol{\theta}_{1:k}). \quad (3)$$

This joint clearly admits the desired target distribution as a marginal. Apart from some weak assumptions, which we will discuss shortly, the distribution q'_t is entirely arbitrary, and may depend on the data and the time step. In fact, in the application to RBF regression we consider here we will set it to $q'_t(k', \boldsymbol{\theta}'_{1:k'}|k, \boldsymbol{\theta}_{1:k}) = \delta_{(k,\boldsymbol{\theta}_{1:k})}(k', \boldsymbol{\theta}'_{1:k'})$, so that it effectively disappears from the expression above. Note also that k and k' are not constrained in any sense, so that the target and auxiliary parameter spaces may be of different dimension. A similar strategy of augmenting the space to simplify the importance sampling procedure has been exploited before in [1] to develop efficient SMC samplers for a wide range of models. To generate samples in this joint space we define our proposal to be of the form

$$Q_t(k, \boldsymbol{\theta}_{1:k}; k', \boldsymbol{\theta}'_{1:k'}) = p_{t-1}(k', \boldsymbol{\theta}'_{1:k'}|\mathbf{y}_{1:t-1}) \times q_t(k, \boldsymbol{\theta}_{1:k}|k', \boldsymbol{\theta}'_{1:k'}), \quad (4)$$

where q_t may again depend on the data and the time step. This proposal embodies the sequential character of our algorithm. Similar to SMC methods [3] it generates samples for the parameters at the current time step by incrementally refining the posterior at the previous time step through the distribution q_t. Designing efficient incremental proposals is much easier than constructing proposals that generate samples directly in the target parameter space, since the posterior is unlikely to undergo dramatic changes over consecutive time steps. To compensate for the discrepancy between the proposal in (4) and the joint posterior in (3) the importance weight takes the form

$$W_t = \frac{p_t(k, \boldsymbol{\theta}_{1:k}|\mathbf{y}_{1:t}) q'_t(k', \boldsymbol{\theta}'_{1:k'}|k, \boldsymbol{\theta}_{1:k})}{p_{t-1}(k', \boldsymbol{\theta}'_{1:k'}|\mathbf{y}_{1:t-1}) q_t(k, \boldsymbol{\theta}_{1:k}|k', \boldsymbol{\theta}'_{1:k'})}. \quad (5)$$

Due to the construction of the joint in (3) marginal samples in the target parameter space associated with this weighting will indeed be distributed according to the target posterior $p_t(k, \boldsymbol{\theta}_{1:k}|\mathbf{y}_{1:t})$. As might be expected the importance weight in (5) is similar in form to the acceptance ratio for the RJ-MCMC algorithm [2]. One notable difference is that the reversibility condition is not required, so that for a given q_t, q'_t may be arbitrary, as long as the ratio in (5) is well-defined.

In practice it is often necessary to design a number of candidate moves to obtain an efficient algorithm. Examples include update moves to refine the model parameters in the light of the new data, birth moves to add new parameters to better explain the new data, death moves to remove redundant or erroneous parameters, and many more. We will denote the set of candidate moves at time t by $\{\alpha_{t,i}, q_{t,i}, q'_{t,i}\}_{i=1}^{M}$, where $\alpha_{t,i}$ is the probability of choosing move i, with $\sum_{i=1}^{M} \alpha_{t,i} = 1$. For each move i the importance weight $W_{t,i}$ is computed by substituting the corresponding $q_{t,i}$ and $q'_{t,i}$ into (5). Note that the probability of choosing a particular move may depend on the old state and the time step, so that moves may be included or excluded as appropriate.

3. EXPERIMENTS AND RESULTS

3.1. Algorithmic Details

We will consider three kinds of moves: a zero move $q_{t,1}$, a birth move $q_{t,2}$, and a death move $q_{t,3}$. The zero move leaves the model parameters unchanged. The birth move adds a new kernel at a uniformly randomly chosen location over the grid of unoccupied input data points. The death move removes a uniformly randomly chosen kernel. For $k = 0$ only the birth move is possible, whereas the birth move is impossible for $k = k_{\max}$ or $k = t$. Similar to [2] we set the move probabilities to

$$\begin{aligned} \alpha_{t,2} &= c \min\{1, p(k+1)/p(k)\} \\ \alpha_{t,3} &= c \min\{1, p(k-1)/p(k)\} \\ \alpha_{t,1} &= 1 - \alpha_{t,2} - \alpha_{t,3} \end{aligned}$$

in all other cases. In the above $c \in (0, 1)$ is a parameter that tunes the relative frequency of the dimension changing moves to the zero move. Given the expressions for the prior and posterior in (1) and (2), respectively, the generic

expression for the importance weight in (5) becomes

$$W_{t,i} \propto \frac{|\mathbf{B}|^{1/2}\exp(-(\mathbf{y}^{\mathrm{T}}\mathbf{P}\mathbf{y} - \mathbf{y}'^{\mathrm{T}}\mathbf{P}'\mathbf{y}')/2\sigma_y^2)}{|\mathbf{B}'|^{1/2}(2\pi\sigma_y^2)^{1/2}(\sigma_\beta^2)^{k-k'/2}} \times \frac{\lambda^{k-k'}(t-1)(k'-1)!}{t(k-1)!q_{t,i}(k,\boldsymbol{\mu}_{1:k}|k',\boldsymbol{\mu}'_{1:k'})},$$

where the primed variables are those corresponding to the posterior at time $t-1$. Note that for each move we have set the arbitrary distribution $q'_{t,i}$ to the Dirac delta mass centred on the values for the centres at the previous time step, so that this term effectively disappears from the expression for the importance weight. For the zero move the parameters are left unchanged, so that the expression for $q_{t,1}$ in the importance weight becomes unity. This is often a good move to choose, and captures the notion that the posterior rarely changes dramatically over consecutive time steps. For the birth move one new kernel is added, so that $k = k' + 1$. The centre for this kernel is uniformly randomly chosen from the grid of unoccupied input data points. This means that the expression for $q_{t,2}$ in the importance weight reduces to $1/(t-k')$, since there are $t-k'$ such data points. Similarly, the death move removes a uniformly randomly chosen kernel, so that $k = k' - 1$. In this case the expression for $q_{t,3}$ in the importance weight reduces to $1/k'$.

In the following sections we will evaluate the performance of the sequential RBF regression strategy on two standard benchmark data sets: the sinc data and the Boston housing data.

3.2. Sinc Data

This experiment is described in [4]. The training data is taken to be the sinc function, *i.e.* $\mathrm{sinc}(x) = \sin(x)/x$, corrupted by additive Gaussian noise of standard deviation $\sigma_y = 0.1$, for 50 evenly spaced points in the interval $x \in [-10, 10]$. In all the runs we presented these points to the sequential estimation algorithm in random order. For the test data we used 1000 points over the same interval. We used a Gaussian kernel of width 1.6, and set the fixed parameters of the model to[1] $(\lambda, k_{\max}) = (1, 50)$. The fraction of dimension change moves was set to $c = 0.25$.

The left side of Figure 1 shows the test error as a function of the number of samples N. These results were obtained by averaging over 25 random generations of the training data for each value of N. As expected, the error decreases with an increase in the number of samples. No significant decrease is obtained beyond $N = 250$, and we adopted this value for subsequent comparisons. A typical minimum mean square error (MMSE) estimate of the clean data is shown in the bottom of Figure 1.

[1]To perform the estimation we have, in fact, defined non-informative inverted gamma priors over σ_β^2 and σ_y^2, and estimated these alongside the kernel centres. Details can be found in [5].

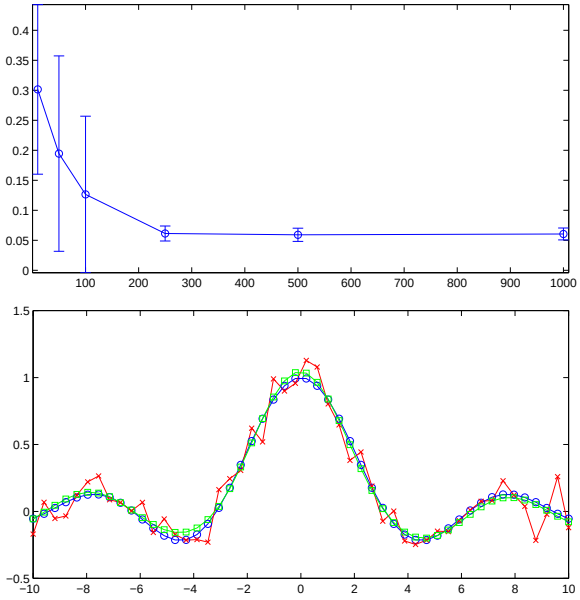

Fig. 1. **Results for the sinc experiment**. Test error as a function of the number of samples (top), and example fit (bottom), showing the uncorrupted data (blue circles), noisy data (red crosses) and MMSE estimate (green squares). For this example the test error was 0.0309 and an average of 6.18 kernels were used.

In Table 1 we compare the results of the TD-SMC algorithm with a number of batch strategies for the support vector machine (SVM) [6] and relevance vector machine (RVM) [7, 8]. The results for the batch algorithms are duplicated from [4, 9]. The error for the TD-SMC algorithm is slightly higher. This is due to the stochastic nature of the algorithm, and the fact that it uses only very simple moves that take no account of the characteristics of the data during the move proposition. This increase should be offset against the algorithm simplicity and efficiency. The error could be further decreased by designing more complex moves.

Method	Test Error	# Kernels
Figueiredo [10]	0.0455	7.0
SVM [6]	0.0519	28.0
RVM [7, 8]	0.0494	6.9
Variational RVM [4]	0.0494	7.4
MCMC [9]	0.0468	6.5
TD-SMC	0.0591	4.5

Table 1. **Comparative performance results for the sinc data**. The batch results are reproduced from [4, 9].

3.2.1. Boston Housing Data

We also applied our algorithm to the popular Boston housing data set. We considered random train / test partitions of the data of size 300 / 206. We again used a Gaussian kernel, and set the width parameter to 5. For the model and algorithm parameters we used values similar to those for the sinc experiment, except for setting $\lambda = 5$ to allow a larger number of kernels. The results are summarised in Table 2. These were obtained by averaging over 10 random partitions of the data, and setting the number of samples to $N = 250$. The test error is comparable to those for the batch strategies, but far fewer kernels are required.

Method	Test Error	# Kernels
SVM [6]	8.04	142.8
RVM [7, 8]	7.46	39.0
TD-SMC	7.18	25.29

Table 2. **Comparative performance results for the Boston housing data**. The batch results are reproduced from [7].

4. CONCLUSIONS

In this paper we introduced the TD-SMC algorithm for sampling from a sequence of distributions defined on a union of subspaces, and applied it to the problem of sequential RBF regression. Our algorithm is based on a generalisation of importance sampling, and incrementally updates a Monte Carlo representation of the target posterior distribution as more data points become available. It achieves this through simple and intuitive model moves, reminiscent of the RJ-MCMC algorithm. It is further non-iterative, and requires only a single pass over the data, thus overcoming some of the computational difficulties associated with batch estimation strategies for RBF regression. Our algorithm is more general than the RBF regression problem considered here. Its application extends to any model for which the posterior can be evaluated up to a normalising constant. Initial experiments on two standard regression data sets showed our algorithm to compare favourably with existing batch estimation strategies for RBF regression.

5. REFERENCES

[1] P. Del Moral and A. Doucet, "Sequential Monte Carlo samplers," Tech. Rep. CUED/F-INFENG/TR.443, Signal Processing Group, Cambridge University Engineering Department, 2002.

[2] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[3] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[4] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, C. Boutilier and M. Goldszmidt, Eds. 2000, pp. 46–53, Morgan Kaufmann.

[5] J. Vermaak, S. J. Godsill, and A. Doucet, "Sequential Bayesian kernel regression," 2003, Submitted to Advances in Neural Information Processing Systems.

[6] V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, 1998.

[7] M. E. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. R. Müller, Eds. MIT Press, 2000, vol. 12, pp. 652–658.

[8] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[9] S. S. Tham, A. Doucet, and R. Kotagiri, "Sparse Bayesian learning for regression and classification using Markov chain Monte Carlo," in *Proceedings of the International Conference on Machine Learning*, 2002, pp. 634–643.

[10] M. Figueiredo, "Adaptive sparseness using Jeffreys prior," in *Advances in Neural Information Processing Systems*, 2001.