PROPAGATION OF UNCERTAINTY IN BAYESIAN KERNEL MODELS — APPLICATION TO MULTIPLE-STEP AHEAD FORECASTING

Joaquin Quiñonero Candela(1), *Agathe Girard*(2), *Jan Larsen*(1) & *Carl Edward Rasmussen*(3)

(1)Informatics and Math. Modelling
Technical University of Denmark
Richard Petersens Plads, Build. 321
2800 Kongens Lyngby, Denmark
{jqc,jl}@imm.dtu.dk

(2)Dept. of Computing Science
Glasgow University
17 Lilybank Gardens
Glasgow G12 8QQ, Scotland
agathe@dcs.gla.ac.uk

(3)Biological Cybernetics
Max Planck Institute
Spemann straße 38
72076 Tübingen, Germany
carl@tuebingen.mpg.de

ABSTRACT

The object of Bayesian modelling is the predictive distribution, which in a forecasting scenario enables evaluation of forecasted values and their uncertainties. In this paper we focus on reliably estimating the predictive mean and variance of forecasted values using Bayesian kernel based models such as the Gaussian Process and the Relevance Vector Machine. We derive novel analytic expressions for the predictive mean and variance for Gaussian kernel shapes under the assumption of a Gaussian input distribution in the static case, and of a recursive Gaussian predictive density in iterative forecasting. The capability of the method is demonstrated for forecasting of time-series and compared to approximate methods.

1. INTRODUCTION

The problem of nonlinear forecasting is relevant to numerous application domains e.g. in financial modelling and control. This paper focuses on providing better estimates of the forecasted value as well as its uncertainty. The object of interest in Bayesian modelling framework [1] is the predictive density which contains all information about the forecasted value given the history of known values. For many Bayesian models the predictive density can only be approximated using Monte-Carlo sampling, local expansions, or variational approaches. However, when using Bayesian Gaussian shaped kernel models such as the Gaussian Process (GP) with a Gaussian kernel [1, 2] or the Relevance Vector Machine (RVM) [3, 4] the predictive mean and variance are given by analytic expressions under mild assumptions. Moreover the Bayesian kernel methods have proven to be very efficient nonlinear models [2, 4], with flexible approximation capabilities and high generalization performance.

We focus on the nonlinear auto-regressive (NAR) model with Gaussian innovations although more flexible nonlinear time-series models [5] sometimes are more efficient. Multi-step ahead forecasting can be done as direct forecast or as iterative one-step ahead forecasting. In [6] it is concluded that iterative forecasting usually is superior to direct forecasting. Generally the complexity of the nonlinear mapping in direct forecasting increases with the forecast horizon and for a fixed length time-series the number of training examples decreases with the forecast horizon. In iterative forecasting the complexity of the nonlinear mapping is much lower than in the direct case, the number of training samples higher, but the performance is diminished by the uncertainty of the forecasted values. Consequently the involved effects provide a delicate trade-off. We restrict this work to iterative forecasting, which offers the additional advantage that multi-step ahead forecasts can be obtained with only one trained model.

In classical iterative forecasting only the predictive mean is iterated, here we consider an improvement to the methods suggested in [7] which iterate both the predictive mean and variance. This corresponds to using the model in recall/test phase under uncertain input. We do not consider training the model under uncertain inputs, which has been addressed for nonlinear model in [8] and for linear models in e.g, [9].

In section 2 we introduce the Bayesian modelling framework. In section 3 we consider the evaluation of the prediction density with uncertain inputs, which is formulated for time-series forecasting in section 4. Finally section 5 provides numerical experiments, that demonstrate the capability of the proposed method.

2. BAYESIAN KERNEL MODELLING

Consider a D-dimensional column input vector $\boldsymbol{x}$ and a single output y, then the nonlinear model is defined as[1]

$$y = f(\boldsymbol{x}) + \varepsilon, \tag{1}$$

where $f(\cdot)$ is a nonlinear function implemented as a GP or a RVM, and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is additive i.i.d. Gaussian noise with variance σ_ε^2. Suppose that the training data set is $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where N is the number of training samples. When using a GP [1, 2] or a RVM [3, 4], the predictive distribution of the output, y, is Gaussian [10],

$$p(y|\boldsymbol{x}, \mathcal{D}) = \mathcal{N}(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})), \tag{2}$$

where $\boldsymbol{x}$ is an arbitrary input at which we perform prediction. For a GP the mean and variance of the predictive distribution are given by

$$\mu(\boldsymbol{x}) = \boldsymbol{k}^\top(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{y}, \quad \sigma_{\mathrm{GP}}^2(\boldsymbol{x}) = 1 - \boldsymbol{k}^\top(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x}), \tag{3}$$

This work is supported by the Multi-Agent Control Research Training Network - EC TMR grant HPRN-CT-1999-00107. Roderick Murray-Smith is acknowledged for useful discussions.

[1] We tacitly assume that y has zero mean, although a bias term can be included, see further [10].

where $C_{\text{GP}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the kernel, which we set to the commonly used Gaussian form[2]. We have

$$\begin{aligned} C_{\text{GP}}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \exp[-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)/2], &(4)\\ \boldsymbol{\Lambda} &= \text{diag}[\lambda_1^2, \dots, \lambda_D^2], &(5)\\ \boldsymbol{K} &= \{K_{ij}\} = \{C_{\text{GP}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma_\varepsilon^2 \delta_{ij}\} &(6)\\ \boldsymbol{k}(\boldsymbol{x}) &= [C_{\text{GP}}(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, C_{\text{GP}}(\boldsymbol{x}, \boldsymbol{x}_N)]^\top, &(7)\\ \boldsymbol{y} &= [y_1, \cdots, y_N]^\top. &(8) \end{aligned}$$

The kernel width hyper-parameters, λ_p, are fitted by maximizing the evidence (ML-II) using conjugate gradient, see e.g. [2].

For the RVM, let $\{\phi_j(\boldsymbol{x})\}$ and $\{\alpha_j\}$ with $j = 1, 2, \cdots, M$ be respectively the basis functions and the weight hyper-parameters, where M is the number of relevance vectors. Since typically $M < N$, the RVM yields sparse kernels, spanned by a finite number of basis functions [3, 10]. For the RVM the predictive distribution (2) has mean and variance specified by

$$\mu(\boldsymbol{x}) = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{w}_{\text{MP}}, \quad \sigma^2_{\text{RVM}}(\boldsymbol{x}) = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi}(\boldsymbol{x}), \qquad (9)$$

where, choosing Gaussian basis functions, we have

$$\begin{aligned} \boldsymbol{w}_{\text{MP}} &= \sigma_\varepsilon^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top \boldsymbol{y}, &(10)\\ \boldsymbol{\Sigma} &= (\sigma_\varepsilon^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{A})^{-1}, &(11)\\ \boldsymbol{A} &= \text{diag}[\alpha_1, \cdots, \alpha_M], &(12)\\ \phi_j(\boldsymbol{x}) &= \exp[-(\boldsymbol{x}_j - \boldsymbol{x})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_j - \boldsymbol{x})/2], &(13)\\ \boldsymbol{\phi}(\boldsymbol{x}) &= [\phi_1(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x})]^\top, &(14)\\ \boldsymbol{\Phi} &= \{\Phi_{ij}\} = \{\phi_j(\boldsymbol{x}_i)\},\ i = [1; N],\ j = [1; M]. &(15) \end{aligned}$$

The details of training the RVM are described in [3, 4].

3. PREDICTION WITH UNCERTAIN INPUT

Assume that the test input $\boldsymbol{x}$ can not be observed directly and the uncertainty is modeled as $\boldsymbol{x} \sim p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{u}, \boldsymbol{S})$, with mean $\boldsymbol{u}$ and covariance matrix $\boldsymbol{S}$. The resulting predictive distribution is then obtained by marginalizing over the test input

$$p(y|\boldsymbol{u}, \boldsymbol{S}, \mathcal{D}) = \int p(y|\boldsymbol{x}, \mathcal{D})p(\boldsymbol{x})\, d\boldsymbol{x}. \qquad (16)$$

The principle is shown in Figure 1. The marginalization can in most cases only be carried out using Monte-Carlo numerical approximation techniques, however, in the case of Gaussian kernels[3] it is possible to obtain exact analytical expressions for the mean and variance of the marginalized predictive distribution:

$$\begin{aligned} m(\boldsymbol{u}, \boldsymbol{S}) &= \int y \cdot p(y|\boldsymbol{u}, \boldsymbol{S}, \mathcal{D})\, dy, \quad \text{and} &(17)\\ v(\boldsymbol{u}, \boldsymbol{S}) &= \int (y - m(\boldsymbol{u}, \boldsymbol{S}))^2 p(y|\boldsymbol{u}, \boldsymbol{S}, \mathcal{D})\, dy. &(18) \end{aligned}$$

The proposed method is an extension of the work presented in [7], which makes additional approximations, viz. Taylor series expansions of $\mu(\boldsymbol{x})$ and $\sigma^2(\boldsymbol{x})$ to first and second order around $\boldsymbol{u}$ and

[2]The exponential in equation (4) is usually multiplied by an additional hyperparameter whose value is fitted during training. We here set it to 1 for clarity, which requires normalizing the data to unit variance.

[3]Exact analytical results can also be obtained for polynomial kernels, $C(\boldsymbol{x}_p, \boldsymbol{x}_q) \propto |\boldsymbol{x}_p - \boldsymbol{x}_q|^r$, e.g. a linear model.

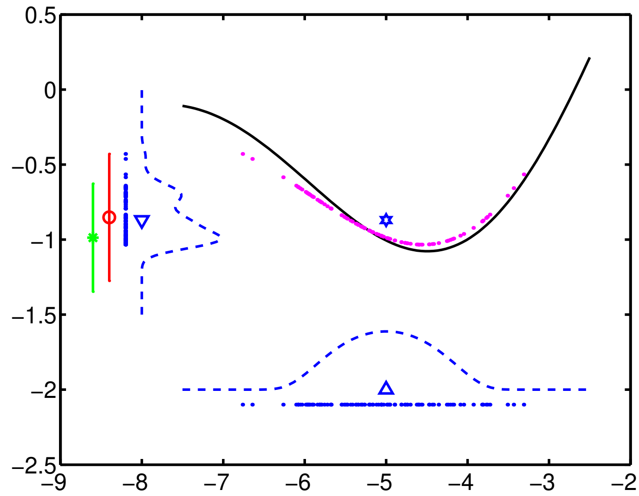

Fig. 1. Prediction with uncertain input. *On the x-axis*, the dashed line represents the Gaussian input distribution, with mean located by the triangle, from which we draw 100 samples (dots under it). *In the middle of the figure*, the solid line represents the true underlying function. We fit a model to it, and propagate the 100 input samples through the model (dots close to the true function). *On the y-axis* we project the 100 predicted values (dots) and use them to estimate the predictive density (dashed line), with mean located by the triangle. The error bar with a circle and the error bar with a star show the mean and 95% confidence interval of the Gaussian approximation with exact computation of mean and variance and of the method with Taylor expansion respectively.

$\boldsymbol{S}$. Using properties of the conditional mean and variance

$$\begin{aligned} m(\boldsymbol{u}, \boldsymbol{S}) &= E_{\boldsymbol{x}}[E_y[y|\boldsymbol{x}]] = E_{\boldsymbol{x}}[\mu(\boldsymbol{x})], &(19)\\ v(\boldsymbol{u}, \boldsymbol{S}) &= E_{\boldsymbol{x}}[V_y[y|\boldsymbol{x}]] + V_{\boldsymbol{x}}[E_y[y|\boldsymbol{x}]] \\ &= E_{\boldsymbol{x}}[\sigma^2(\boldsymbol{x})] + V_{\boldsymbol{x}}[\mu(\boldsymbol{x})], &(20) \end{aligned}$$

where $E_{\boldsymbol{x}}[\cdot]$, $V_{\boldsymbol{x}}[\cdot]$ denote the expectation and variance wrt. $\boldsymbol{x}$. When using Gaussian kernels in GPs and Gaussian basis functions in RVMs, the expressions for $\mu(\boldsymbol{x})$ in eq. (3) and (9) are Gaussian shaped functions of $\boldsymbol{x}$ and the expressions for $\sigma^2(\boldsymbol{x})$ are products of Gaussian shaped functions in $\boldsymbol{x}$. Therefore the integrants involved in determining $m(\boldsymbol{u}, \boldsymbol{S})$ and $v(\boldsymbol{u}, \boldsymbol{S})$ are products of Gaussian shaped functions, which allows an analytical calculation. In [10] it is shown that

$$m(\boldsymbol{u}, \boldsymbol{S}) = \boldsymbol{\beta}^\top \boldsymbol{l}. \qquad (21)$$

For the GP $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_N\} = \boldsymbol{K}^{-1}\boldsymbol{y}$ and for the RVM $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_M\} = \boldsymbol{w}_{\text{MP}}$. Vector $\boldsymbol{l} = \{l_1, \cdots, l_N\}$ is given by

$$\begin{aligned} l_j &= |\boldsymbol{\Lambda}^{-1}\boldsymbol{S} + \boldsymbol{I}|^{-\frac{1}{2}} \\ &\quad \cdot \exp\left[-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{x}_j)^\top(\boldsymbol{\Lambda} + \boldsymbol{S})^{-1}(\boldsymbol{u} - \boldsymbol{x}_j)\right], \qquad (22) \end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix. Note that if $\boldsymbol{S}$ is the zero matrix, then $\boldsymbol{l} = \boldsymbol{k}(\boldsymbol{u})$ and $m(\boldsymbol{u}, \boldsymbol{S}) = \mu(\boldsymbol{u})$ as would be expected.

Further, for the GP

$$\begin{aligned} v(\boldsymbol{u}, \boldsymbol{S}) &= \sigma^2_{\text{GP}}(\boldsymbol{u}) + \text{Tr}\left(\tilde{\boldsymbol{L}}(\boldsymbol{\beta}\boldsymbol{\beta}^\top - \boldsymbol{K}^{-1})\right) \\ &\quad + \text{Tr}\left((\boldsymbol{k}(\boldsymbol{u})\boldsymbol{k}(\boldsymbol{u})^\top - \boldsymbol{l}\boldsymbol{l}^\top)\boldsymbol{\beta}\boldsymbol{\beta}^\top\right), \qquad (23) \end{aligned}$$

where $\tilde{\boldsymbol{L}} = \boldsymbol{L} - \boldsymbol{k}(\boldsymbol{u})\boldsymbol{k}(\boldsymbol{u})^\top$ and the elements of matrix $\boldsymbol{L}$ are

$$L_{ij} = \boldsymbol{k}_i(\boldsymbol{u})\boldsymbol{k}_j(\boldsymbol{u}) \cdot |2\boldsymbol{\Lambda}^{-1}\boldsymbol{S} + \boldsymbol{I}|^{-\frac{1}{2}} \tag{24}$$
$$\cdot \exp\left[2(\boldsymbol{u} - \boldsymbol{x}_d)^\top \boldsymbol{\Lambda}^{-1}(2\boldsymbol{\Lambda}^{-1} + \boldsymbol{S}^{-1})^{-1}\boldsymbol{\Lambda}^{-1}(\boldsymbol{u} - \boldsymbol{x}_d)\right],$$

and where $\boldsymbol{x}_d = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_j)$. For the RVM

$$\begin{aligned} v(\boldsymbol{u}, \boldsymbol{S}) \;=\; & \sigma^2_{\mathrm{RVM}}(\boldsymbol{u}) + \mathrm{Tr}\left(\tilde{\boldsymbol{L}}(\beta\boldsymbol{\beta}^\top + \boldsymbol{\Sigma}^{-1})\right) \\ & + \mathrm{Tr}\left([\boldsymbol{k}(\boldsymbol{u})\boldsymbol{k}(\boldsymbol{u})^\top - \boldsymbol{l}\boldsymbol{l}^\top]\beta\boldsymbol{\beta}^\top\right). \end{aligned} \tag{25}$$

Notice that both for GPs and RVMs, when $\boldsymbol{S}$ is the zero matrix, $\tilde{\boldsymbol{L}}$ is also the zero matrix, again $\boldsymbol{l} = \boldsymbol{k}(\boldsymbol{u})$, and $v(\boldsymbol{u}, \boldsymbol{S}) = \sigma^2(\boldsymbol{u})$.

4. APPLICATION TO TIME-SERIES FORECASTING

Suppose that $\{y_t\}$ are the ordered samples of a time-series, where t is an integer time index. We wish to make time-series forecasting using a NAR model (1), where the inputs are formed by a collection of previous output values, $\boldsymbol{x}_t = [y_{t-1}, y_{t-2}, \ldots, y_{t-L}]$, where the integer L is the size of the lag space.

Given that we have observed the values $y^T \equiv \{y_t\}_{t=1}^T$, T being the number of observed samples, computing the predictive density of the value y_{T+1} is readily given by the model from (2) as

$$p(y_{T+1}|\boldsymbol{x}_{T+1}) = \mathcal{N}(\mu(\boldsymbol{x}_{T+1}), \sigma^2(\boldsymbol{x}_{T+1}))$$

The predictive density of the value y_{T+2} (two steps ahead) depends on $\boldsymbol{x}_{T+1}$, which now contains a stochastic element. In general, the predictive distribution of y_{T+k}, with $k \geq 2$, requires integrating out the uncertainty of the input:

$$p(y_{T+k}|y^T) = \int p(y_{T+k}|\boldsymbol{x}_{T+k})\, p(\boldsymbol{x}_{T+k}|y^T)\, d\boldsymbol{x}_{T+k}. \tag{26}$$

It is straightforward that this scheme leads to a recursive density estimation. The integral in (26) has no analytical solution. A naïve approach to the recursion is to ignore the uncertainty in the distribution of the input by setting $p(\boldsymbol{x}_{T+k}) = \delta(\boldsymbol{x} - [\mu(\boldsymbol{x}_{T+k-1}), \ldots, \mu(\boldsymbol{x}_{T+k-L})]^\top)$[4], thus propagating only the mean predictions. This method yields very poor error-bars, since it in some way only considers one step ahead predictions, treating the previous predicted values as exact, and is therefore overconfident, [7]. Alternatively, one can approximate the predictive density of y_{T+k} by a Gaussian density and compute only the mean and variance of $p(y_{T+k}|y^T)$. By doing this one ensures that the input distribution $p(\boldsymbol{x}_{T+k}|y^T)$ is always Gaussian, which allows to use the results described in section 3 for computing the mean and variance of y_{T+K}, see eq. (26). This can be done exactly (for Gaussian or polynomial kernels) or in an approximate fashion, [7]. The recursive mechanism works because the predictive distribution of y_{T+1} at the first step is Gaussian (26), and therefore the input distribution of $\boldsymbol{x}_{T+1}$ is also Gaussian. We call this procedure of recursively approximating the predictive density by a Gaussian the Recursive Gaussian Predictive Density (RGPD), and distinguish between exact-RGPD for the case of exact computation of mean and variance and approximate-RGPD for the case where the model is approximated by a Taylor expansion, [7].

[4] Where $\delta(x)$ is 1 for $x = 0$ and 0 otherwise. If $k < L$, we have simply $\mu(\boldsymbol{x}_n) = y_n$ for $n < T$.

In the RGPD scheme, the input distribution is given by[5]

$$p(\boldsymbol{x}_{T+k}|y^T) = \mathcal{N}(\mathbf{u}_{T+k}, \mathbf{S}_{T+k}), \tag{27}$$

where

$$\mathbf{u}_{T+k} = [m(\boldsymbol{u}_{T+k-1}, \boldsymbol{S}_{T+k-1}), \ldots, m(\boldsymbol{u}_{T+k-L}, \boldsymbol{S}_{T+k-L})],$$

and where $\mathbf{S}_{T+k}$ is iteratively computed by using the fact that its first column is given by

$$(\mathbf{S}_{T+k})_{1:L,1} = \mathrm{cov}(y_{T+k}, \boldsymbol{x}_{T+k}) = \sum_j \beta_j L_j(\mathbf{c}_j - \mathbf{u}_{T+k}), \tag{28}$$

where $\mathbf{c}_j = (\boldsymbol{\Lambda}^{-1} + \boldsymbol{S}^{-1})^{-1}(\boldsymbol{\Lambda}^{-1}\boldsymbol{x}_j + \boldsymbol{S}^{-1}\boldsymbol{u})$, refer to [10].

5. EXPERIMENTS

We examine the comparative performance of the exact and approximate-RGPD on a hard prediction problem, the Mackey-Glass chaotic time series [11], which is well-known for its strong non-linearity. In [4] we showed that non-linear models, in particular RVMs, have a prediction error four orders of magnitude lower than optimized linear models. The Mackey-Glass attractor is a non-linear chaotic system described by the following equation:

$$\frac{dz(t)}{dt} = -bz(t) + a\frac{z(t-\tau)}{1 + z(t-\tau)^{10}} \tag{29}$$

where the constants are set to $a = 0.2$, $b = 0.1$ and $\tau = 17$. The series is re-sampled with period 1 according to standard practice. The inputs are formed by $L = 16$ samples spaced 1 periods from each other $\mathbf{x}_k = [z_{k-1}, z_{k-2}, \ldots, z_{k-L}]$ and the targets are chosen to be $y_k = z_k$.

We train a GP model with Gaussian kernel on only 100 examples — enough to obtain a 1-step ahead normalized mean squared error on the order of 10^{-4}. Besides, we normalize the data and contaminate it with a small amount of Gaussian noise with variance 10^{-3}. Figure 2 shows the result of making 100 iterative predictions using a GP model, both for the exact-RGPD and the approximate-RGPD methods. By informal visual inspection, the error-bars of the exact-RGPD seem to be better than those of the approximate-RGPD. Consequently the exact-RGPD produces a better predictive density, which we show in figure 3. The mean value of the predictions seems also to be a slightly closer to the true target values for the exact-RGPD than for the approximate-RGPD.

In order to better evaluate the performance of the proposed methods, for a given prediction horizon, we compute the negative log predictive density, the squared error and the absolute error. While the two last measures only take into consideration the mean of the Gaussian predictive distribution, the first one also takes into account its variance. We average over 200 repetitions with different starting points (chosen at random from the series), and represent averages of the three loss measures for prediction horizons ranging from 1 to 100. Figure 3 shows the results. The means are slightly better for the exact-RGPD, but the predictive distribution is much improved. The better error-bars obtained by the exact-RGPD result in a lower value of the negative log predictive density for all values of the prediction horizon. The performance of the naïve iterative method is identical to that of the approximate-RGPD in terms of absolute and squared error. In terms of predictive density (since it produces unrealistic small error-bars) its performance is so poor that it is not worth reporting.

[5] If $k < L$, we have simply $\boldsymbol{\mu}(\boldsymbol{x}_n) = y_n$ for $n < T$.

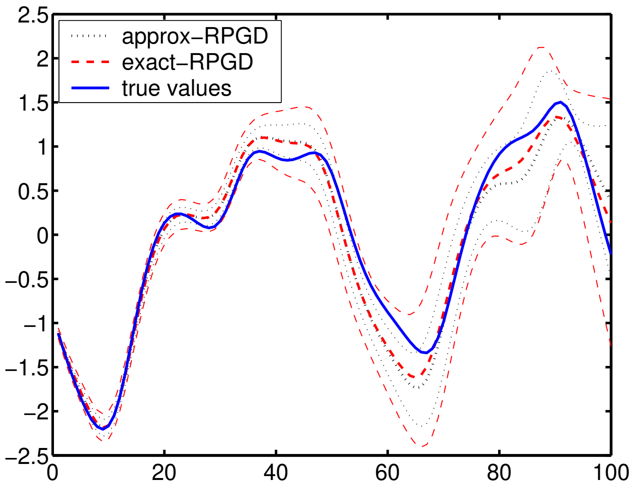

Fig. 2. 100 iterated predictions for the exact-RGPD (dashed) and approximate-RGPD (dotted): for each the thicker lines represent the mean of the predictive distributions and the two thinner lines around represent the upper and lower bounds of the 95% confidence interval of the Gaussian predictive distributions. The solid line shows the true target values.

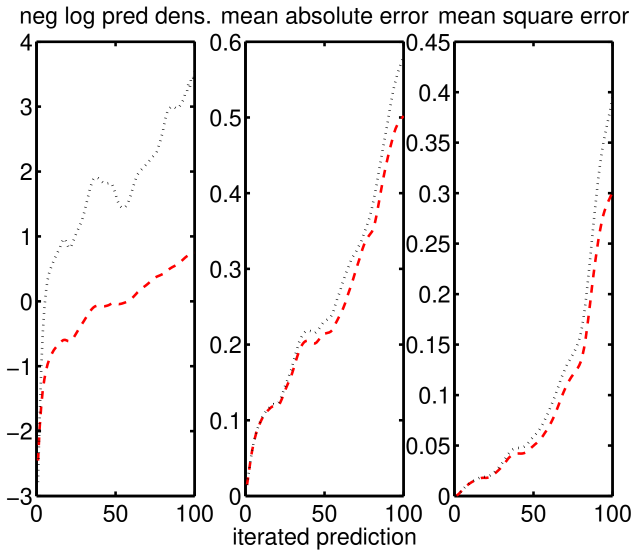

Fig. 3. Negative log predictive density, mean absolute error and mean squared error as a function of the iterative prediction horizon for the exact-RGPD method (dashed) and for the approximate-RGPD (dotted). Averages over 200 repetitions.

6. CONCLUSIONS

We have derived analytical expressions for the exact computation of the mean and variance of the marginalized predictive distribution for uncertain Gaussian test inputs. This analytical expressions are valid for Gaussian processes and the Relevance Vector Machine (extended linear models) with Gaussian or polynomial kernels or basis functions. Our results extend the approximate method presented in [7], where the mean prediction was unaffected by the input uncertainty. In our case the input uncertainty biases the mean prediction, by smoothing, which is interesting in itself for predictions on noisy inputs. Furthermore, in the context of iterated time-series forecasting, our exact-RGPD not only gives much better error-bars, but the mean predictions are closer to the true values, both in terms of absolute and squared error. We are currently investigating efficient Monte Carlo methods to avoid the Gaussian approximation of the recursive predictive density.

7. REFERENCES

[1] Radford M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, no. 118. Springer, New York, 1996.

[2] Carl E. Rasmussen, *Evaluation of Gaussian Processes and Other Methods for Non-linear Regression*, Ph.D. thesis, Dept. of Computer Science, University of Toronto, 1996.

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

[4] Joaquin Quiñonero-Candela and Lars Kai Hansen, "Time series prediction based on the relevance vector machine with adaptive kernels," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, pp. 985–988.

[5] I.J. Leontaritis and S.A. Billings, "Input-output parametric models for non-linear systems, part 1: Deterministic non-linear systems, part 2: Stochastic non-linear systems," *International Journal of Control*, vol. 41, pp. 303–344, 1985.

[6] J. Doyne Farmer and John J. Sidorowich, "Exploiting chaos to predict the future and reduce noise," Tech. Rep. LA-UR-88, Los Alamos National Laboratory, 1988.

[7] Agathe Girard, Carl Edward Rasmussen, and Roderick Murray-Smith, "Gaussian process with uncertain input - application to multiple-step ahead time-series forecasting," in *Advances in Neural Information Processing Systems 15*, Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, Eds. 2003, MIT Press.

[8] Volker Tresp, Subutai Ahmad, and Ralph Neuneier, "Training neural networks with deficient data," in *Advances in Neural Information Processing Systems 6*, Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, Eds. 1994, pp. 128–135, Morgan Kaufmann Publishers, Inc.

[9] Oscar Nestares, David J. Fleet, and David J. Heeger, "Likelihood functions and confidence bounds for total-least-squares problems," in *Proceedings of IEEE Conference on Computer Vision and Pattern Regcognition*, 2000, vol. 1, pp. 523–530.

[10] Joaquin Quiñonero-Candela and Agathe Girard, "Prediction at an uncertain input for gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting," Tech. Rep., IMM, DTU, 2002, http://isp.imm.dtu.dk/staff/jqc/prop_uncert.ps.gz.

[11] M.C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, pp. 287–289, July 1977.