Gaussian Processes: Prediction at a Noisy Input and Application to Iterative Multiple-Step Ahead Forecasting of Time-Series

Agathe Girard[1] and Roderick Murray-Smith[1,2]

[1] Department of Computing Science, University of Glasgow, 17 Lilybank Gardens,
Glasgow G12 8QQ, UK,
agathe@dcs.gla.ac.uk
[2] Hamilton Institute, Maynooth, Ireland

Abstract. With the Gaussian Process model, the predictive distribution of the output corresponding to a new given input is Gaussian. But if this input is uncertain or noisy, the predictive distribution becomes non-Gaussian. We present an analytical approach that consists of computing only the mean and variance of this new distribution (*Gaussian approximation*). We show how, depending on the form of the covariance function of the process, we can evaluate these moments exactly or approximately (within a Taylor approximation of the covariance function). We apply our results to the iterative multiple-step ahead prediction of non-linear dynamic systems with propagation of the uncertainty as we predict ahead in time. Finally, using numerical examples, we compare the *Gaussian approximation* to the numerical approximation of the true predictive distribution by simple Monte-Carlo.

1 Background

Given a set of observed data $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^D$ and $t_i = f(\mathbf{x}_i) + \epsilon, \in \mathcal{R}$ (ϵ is a white noise with variance v_t), we model the input/output relationship using a zero-mean Gaussian Process (GP) with covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$. For the moment, we do not specify the form of the covariance function and simply assume it is a valid one, generating a positive definite covariance matrix. We refer to [1, 2, 3, 4] for a review of GPs.

1.1 Prediction at a New x

With this model, given a new 'test' input $\mathbf{x}$, and based on the observed data, the predictive distribution of the corresponding output $y = f(\mathbf{x})$ is readily obtained. This distribution is Gaussian, $p(y|\mathcal{D}, \mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, with mean and variance respectively given by

R. Murray-Smith, R. Shorten (Eds.): Switching and Learning, LNCS 3355, pp. 158–184, 2005.

$$\begin{cases} \mu(\mathbf{x}) = \sum_{i=1}^{N} \beta_i C(\mathbf{x}, \mathbf{x}_i) \\ \sigma^2(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}) - \sum_{i,j=1}^{N} K_{ij}^{-1} C(\mathbf{x}, \mathbf{x}_i) C(\mathbf{x}, \mathbf{x}_j) \end{cases} \tag{1}$$

with $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{t}$, where $\mathbf{t}$ is the $N \times 1$ vector of observed noisy targets and $\mathbf{K}$ is the $N \times N$ data covariance matrix, such that $K_{ij} = C(\mathbf{x}_i, \mathbf{x}_j) + v_t \delta_{ij}$. The covariances between the new point and the training cases are given by $C(\mathbf{x}, \mathbf{x}_i)$, for $i = 1 \dots N$, and $C(\mathbf{x}, \mathbf{x})$ is the covariance between the test point and itself.

In practice, the predictive mean $\mu(\mathbf{x})$ is used as a point estimate for the function output, while the variance $\sigma^2(\mathbf{x})$ can be translated into uncertainty bounds (error-bars) on this estimate. Although this variance corresponds to the model's uncertainty (and therefore depends on the prior and on the local data complexity), it represents valuable information as it enables us to quantify the uncertainty attached to the prediction. Figure 1 shows the predictive means and their 2σ error-bars computed for 81 test inputs. A Gaussian Process with zero-mean and Gaussian covariance function (Eq. (22)) was trained using only $N = 10$ points. Near the data points, the predictive variance is small, increasing as the test inputs are far away from the training ones.

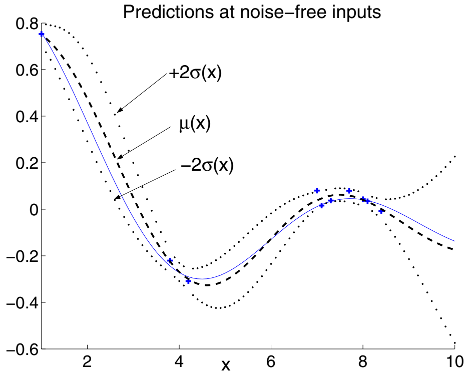

Fig. 1. Predictive means (dashed line) and 2σ error-bars (dotted lines) corresponding to 81 noise-free test inputs. A zero-mean GP was trained on 10 training points (crosses) to learn the underlying function (continuous line).

1.2 Motivation

We first motivate the necessity of being able to make a prediction at an uncertain or noisy input using a dynamic example.

Dynamic Case Let a time-series be known up to time t and assume a simple auto-regressive generative model of the form $y_{t+1} = f(y_t)$ where the input now corresponds to a delayed value of the time-series. Having formed a set of input/output pairs and trained a GP, we wish to predict the value of the time-series at, say, time $t+k$. With our one-step ahead model, we need to iterate predictions up to the desired horizon, i.e. we have $y_{t+k} = f(y_{t+k-1})$, $y_{t+k-1} = f(y_{t+k-2})$, so on, down to $y_{t+1} = f(y_t)$. Since y_t is known, the predictive distribution of y_{t+1} is simply Gaussian, $p(y_{t+1}|\mathcal{D}, y_t) = \mathcal{N}(\mu(y_t), \sigma^2(y_t))$, with mean and variance given by (1) evaluated at $\mathbf{x} = y_t$. For the next time-step, a naive approach consists in only using $\mu(y_t)$ as an estimate for y_{t+1}, $\hat{y}_{t+1} = \mu(y_t)$, and evaluate $p(y_{t+2}|\mathcal{D}, \hat{y}_{t+1}) = \mathcal{N}(\mu(\hat{y}_{t+1}), \sigma^2(\hat{y}_{t+1}))$. As we will see in our numerical examples, this approach is not advisable for two reasons: it is over-confident about the estimate (the variance $\sigma^2(\hat{y}_{t+1})$ will typically be very small) and it is also throwing away valuable information, namely, the uncertainty attached to the estimate $\hat{y}_{t+1}$, $\sigma(y_t)$. If we wish to account for this uncertainty, and thus *propagate* it as we predict ahead in time, we need to be able to evaluate $p(y_{t+2}|\mathcal{D}, y_{t+1})$ where $y_{t+1} \sim \mathcal{N}(\mu(y_t), \sigma^2(y_t))$. This means being able to evaluate the predictive distribution corresponding to an uncertain or noisy input, y_{t+1} here.

Static Case In real experiments and applications, we use sensors and detectors that can be corrupted by many different sources of disturbances. We might then only observe a noise corrupted version of the true input and the system senses the new input imperfectly. Again, if the model does not account for this 'extra' uncertainty (as opposed to the uncertainty usually acknowledged on the observed outputs), the model is too confident, which is misleading and could potentially be dangerous if, say, the model's output were to be used in a decision-making process of a critical application. Note that in this case, the approach we suggest assumes prior knowledge of the input noise variance.

In the next section, we present the problem of predicting at a noisy input when using a Gaussian Process model. We then suggest an analytical approximation and compute the mean and variance of the new predictive distribution (sections 3 and 4). In section 5, we return to the iterative forecasting of a nonlinear time-series to which we apply our results.

Although most of the material presented in this chapter has already been published [5, 6, 7], the present document aims at unifying and presenting the different results in a more principled manner.

2 Prediction at an Uncertain Input

Let the new test input be corrupted by some noise, $\boldsymbol{\epsilon}_{\mathbf{x}} \sim \mathcal{N}_{\boldsymbol{\epsilon}_{\mathbf{x}}}(\mathbf{0}, \boldsymbol{\Sigma}_x)$, such that $\mathbf{x} = \mathbf{u} + \boldsymbol{\epsilon}_{\mathbf{x}}$. That is, we wish to make a prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ and to do so, we need to integrate the predictive distribution $p(y|\mathcal{D}, \mathbf{x})$ over the input distribution[3]

$$p(y|\mathcal{D}, \mathbf{u}, \boldsymbol{\Sigma}_x) = \int p(y|\mathcal{D}, \mathbf{x}) p(\mathbf{x}|\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \ . \tag{2}$$

[3] When the bounds are not indicated, it means that the integrals are evaluated from $-\infty$ to $+\infty$.

For the GP, we have $p(y|\mathcal{D},\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2(\mathbf{x})}} \exp\left[-\frac{1}{2}\frac{(y-\mu(\mathbf{x}))^2}{\sigma^2(\mathbf{x})}\right]$, which is a nonlinear function of $\mathbf{x}$, such that this integral cannot be solved without resorting to approximations.

2.1 Possible Approximations

Many techniques are available to approximate intractable integrals of this kind. Approximation methods are divided into deterministic approximations and Monte-Carlo numerical methods. The most popular deterministic approaches are variational methods,[4] Laplace's method and Gaussian quadrature that consist of analytical approximations of the integral. Refer to [4] for a review of these methods.

Numerical methods relying on Markov-Chain Monte-Carlo sampling techniques evaluate the integral numerically, thus approximating the true distribution. In our case, the numerical approximation by simple Monte-Carlo is straightforward since we simply need to sample from a Gaussian distribution $\mathcal{N}_{\mathbf{x}}(\mathbf{u},\boldsymbol{\Sigma}_x)$. For each sample $\mathbf{x}^t$ from this distribution, $p(y|\mathcal{D},\mathbf{x}^t)$ is normal, with mean and variance given by Eqs. (1):

$$p(y|\mathcal{D},\mathbf{u},\boldsymbol{\Sigma}_x) \simeq \frac{1}{T}\sum_{t=1}^{T} p(y|\mathcal{D},\mathbf{x}^t) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}_y(\mu(\mathbf{x}^t),\sigma^2(\mathbf{x}^t)) \ . \tag{3}$$

The numerical approximation of $p(y|\mathcal{D},\mathbf{u},\boldsymbol{\Sigma}_x)$ is then a mixture of T Gaussians with identical mixing proportions. As the number of samples T grows, the approximate distribution will tend to the true distribution.

On Fig. 2, 100 predictive means with their corresponding uncertainties are plotted, corresponding to 100 samples x^t from $p(x)$, centered at the noisy observed input x (asterisks), with variance $v_x = 1$. The 'true' test inputs are $u = 2$ (left) and $u = 6$ (right). The histograms of the samples at which predictions are made are shown on Fig. 3. The circle and asterisk indicate the noise-free and noisy inputs (u and x respectively). After having computed the loss associated to each x^t[5], we find that for which the loss is minimum (triangle), which is close to the true value.

In the remaining of this document, we focus on an analytical approximation which consists of computing only the first two moments, the mean and variance, of $p(y|\mathcal{D},\mathbf{u},\boldsymbol{\Sigma}_x)$. As we will now see, approximate or exact moments are computed, depending on the form of the covariance function.

2.2 Analytical Approximation

To distinguish from $\mu(\mathbf{u})$ and $\sigma^2(\mathbf{u})$, the mean and variance of the Gaussian predictive distribution $p(y|\mathcal{D},\mathbf{u})$ corresponding to a noise-free $\mathbf{u}$, we denote by

[4] Many references can be found at http://www.gatsby.ucl.ac.uk/vbayes/

[5] We compute the squared error and the minus log-predictive density, see section 6.

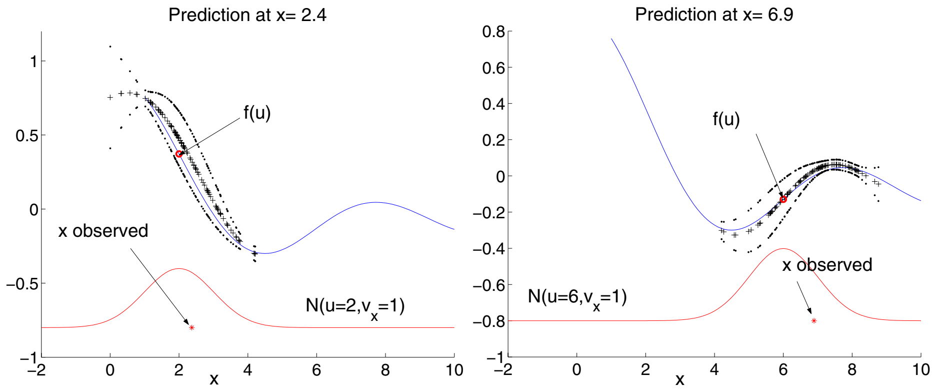

Fig. 2. Monte-Carlo approximation to the prediction at an observed noisy input x (asterisk). Predictive means $\mu(x^t)$ (crosses) with $2\sigma(x^t)$ error-bars (dots), computed for 100 samples x^t from $p(x)$, with mean x and variance v_x. The true input distribution is $x \sim \mathcal{N}_x(u, v_x)$, for $u = 2$ (left), $u = 6$ (right) and $v_x = 1$. The circle indicates the output corresponding to the noise-free input u.

$m(\mathbf{u}, \boldsymbol{\Sigma}_x)$ the mean and by $v(\mathbf{u}, \boldsymbol{\Sigma}_x)$ the variance of the non-Gaussian predictive distribution $p(y|\mathcal{D}, \mathbf{u}, \boldsymbol{\Sigma}_x)$, corresponding to $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$. This can be interpreted as a *Gaussian approximation*, such that

$$p(y|\mathcal{D}, \mathbf{u}, \boldsymbol{\Sigma}_x) \approx \mathcal{N}(m(\mathbf{u}, \boldsymbol{\Sigma}_x), v(\mathbf{u}, \boldsymbol{\Sigma}_x)) \ .$$

This mean and variance are respectively given by

$$\begin{aligned} m(\mathbf{u}, \boldsymbol{\Sigma}_x) &= \int y \left\{ \int p(y|\mathcal{D}, \mathbf{x}) p(\mathbf{x}|\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \right\} dy \\ v(\mathbf{u}, \boldsymbol{\Sigma}_x) &= \int y^2 \left\{ \int p(y|\mathcal{D}, \mathbf{x}) p(\mathbf{x}|\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \right\} dy - m(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \ . \end{aligned}$$

Using the law of iterated expectations and that of conditional variances,[6] we directly have

$$m(\mathbf{u}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})] \tag{4}$$

$$v(\mathbf{u}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + \mathrm{Var}_{\mathbf{x}}[\mu(\mathbf{x})] \ , \tag{5}$$

[6] Recall that $E[X] = E[E[X|Y]]$ and $\mathrm{Var}[X] = E[\mathrm{Var}[X|Y]] + \mathrm{Var}[E[X|Y]]$.

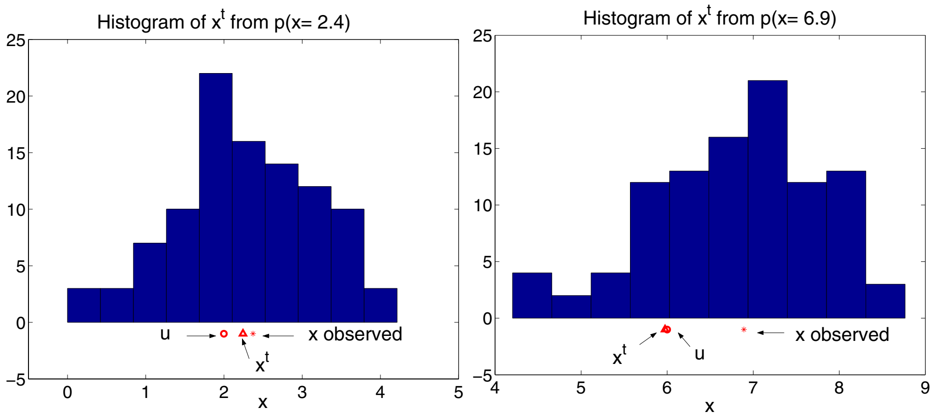

Fig. 3. Histogram of the samples x^t from $p(x)$ at which predictions were made, when the true input (circle) is $u = 2$ (left) and $u = 6$ (right). Also plotted, the observed noisy input (asterisk), taken as the mean of $p(x)$, and the sample x^t that leads to the minimum loss (triangle).

where $\mathrm{Var}_{\mathbf{x}}[\mu(\mathbf{x})] = E_{\mathbf{x}}[\mu(\mathbf{x})^2] - m(\mathbf{u}, \boldsymbol{\Sigma}_x)^2$. Replacing $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ by their expressions (Eqs. (1)), we finally have

$$\begin{cases} m(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i E_{\mathbf{x}}[C(\mathbf{x}, \mathbf{x}_i)] \\ v(\mathbf{u}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[C(\mathbf{x}, \mathbf{x})] - \sum_{i,j=1}^{N} (K_{ij}^{-1} - \beta_i \beta_j) E_{\mathbf{x}}[C(\mathbf{x}, \mathbf{x}_i) C(\mathbf{x}, \mathbf{x}_j)] - m(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \ . \end{cases} \quad (6)$$

Let

$$l = \int C(\mathbf{x}, \mathbf{x}) p(\mathbf{x}) d\mathbf{x} \quad (7)$$

$$l_i = \int C(\mathbf{x}, \mathbf{x}_i) p(\mathbf{x}) d\mathbf{x} \quad (8)$$

$$l_{ij} = \int C(\mathbf{x}, \mathbf{x}_i) C(\mathbf{x}, \mathbf{x}_j) p(\mathbf{x}) d\mathbf{x} \ . \quad (9)$$

How solvable integrals (7)-(9) are basically depends on the form of the covariance function.

1. If the covariance function is e.g. linear, Gaussian, polynomial (or a mixture of those), we can compute the integrals exactly and obtain the *exact* mean and variance. In section 4, we derive the ‘exact’ moments for the linear and Gaussian covariance functions.
2. Otherwise, we can again approximate (7)-(9) in a number of ways. Since we are mostly interested in closed form approximate solutions, we evaluate the

integrals within a Taylor approximation of the covariance function around the mean **u** of **x** and obtain the 'approximate' mean and variance.

Note that this second case might be required, if the form of the covariance function is definitely one for which one cannot solve the integrals exactly, or simply preferable, if the integrals are tractable but at the cost of long and tedious calculations (assuming one has access to software like Mathematica or Matlab's symbolic toolbox to compute the derivatives, the solutions obtained using the proposed approximations provide a suitable performance/implementation trade-off).

Figure 4 summarizes the different possible approximations and highlights the analytical one we take. We now turn to the evaluation of the mean and variance in the case of a 'general' the covariance function, that is when further approximations are needed to evaluate integrals (7)-(9) analytically.

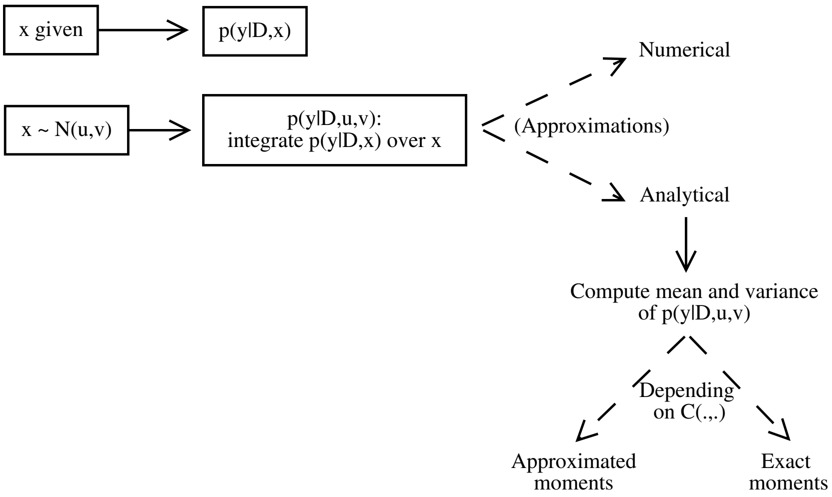

Fig. 4. Dealing with a noisy test input: With the GP model, the predictive distribution of the output corresponding to a new test input x is readily obtained, by conditioning on the training data D and on the new x. If x is noisy, such that $x \sim \mathcal{N}(u, v)$, the new predictive distribution is now obtained by integrating over the input distribution. Since $p(y|D, x)$ is nonlinear in x, the integral is analytically intractable. Although a numerical approximation of the integral is possible, we concentrate on an analytical approximation. We suggest to compute the mean and the variance of the new predictive distribution, which is done exactly or approximately, depending on the parametric form of the covariance function $C(.,.)$.

3 *Gaussian Approximation*: Approximate Moments

We use the Delta method (also called Moment Approximation), which consists of approximating the integrand by a Taylor polynomial. In the one-dimensional case, the Delta method is as follows [8, 9]: Let x be a random variable with mean $E_x[x] = u$ and variance $\mathrm{Var}_x[x] = v_x$, and $y = \phi(x)$. For sufficiently small $\sigma_x = \sqrt{v_x}$ and well-behaved ϕ we can write

$$E_x[y] \simeq \phi(u) + \frac{1}{2} v_x \phi''(u) \tag{10}$$

$$\mathrm{Var}_x[y] \simeq \phi'(u)^2 v_x \tag{11}$$

where ϕ' and ϕ'' are the first and second derivatives of ϕ evaluated at u.

These results are simply obtained by considering the expansion of $\phi(x)$ in Taylor series about u, up to the second order:

$$y = \phi(x) = \phi(u) + (x-u)\phi'(u) + \frac{1}{2}(x-u)^2 \phi''(u) + O([(x-u)^3]) \ . \tag{12}$$

By taking the expectation on both sides, we directly find the approximation (10). For the variance, we have $\mathrm{Var}[y] = E[y^2] - E[y]^2$ and the estimate given by (11) corresponds to an approximation of the second order estimate: Neglecting the term in v_x^2 for both $E[y^2]$ and $E[y]^2$, we have

$$\begin{aligned} E[y^2] &\approx \phi(u)^2 + v_x \phi'(u)^2 + \phi(u)\phi''(u) v_x \\ E[y]^2 &\approx \phi(u)^2 + \phi(u)\phi''(u) v_x \end{aligned}$$

leading to (11). This approximation is motivated by the fact that the Taylor approximation is useful for small standard deviations (if σ_x is small, by Chebychev's inequality $P(|x-u| > k\sigma_x) < \frac{1}{k^2}$), such that x will depart only a little from u except on rare occasions and therefore $(x-u)$ will be small.

There are obviously conditions which $\phi(x)$ should fulfill to make the Taylor series possible (in the neighborhood of u) and to avoid anomalies of behavior away from u. As in [8], we do not state such conditions and assume the covariance function to be such that the expressions are valid.

3.1 Approximate Mean

Let $m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ be the approximate mean, such that

$$m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i l_i^{ap}$$

with $l_i^{ap} = E_{\mathbf{x}}[C^{ap}(\mathbf{x}, \mathbf{x}_i)]$ and where $C^{ap}(\mathbf{x}, \mathbf{x}_i)$ corresponds to the second order Taylor polynomial of $C(\mathbf{x}, \mathbf{x}_i)$ around the mean $\mathbf{u}$ of $\mathbf{x}$,

$$C^{ap}(\mathbf{x}, \mathbf{x}_i) = C(\mathbf{u}, \mathbf{x}_i) + (\mathbf{x}-\mathbf{u})^T \mathbf{C}'(\mathbf{u}, \mathbf{x}_i) + \frac{1}{2}(\mathbf{x}-\mathbf{u})^T \mathbf{C}''(\mathbf{u}, \mathbf{x}_i)(\mathbf{x}-\mathbf{u}) \ .$$

We directly have

$$l_i^{ap} = C(\mathbf{u}, \mathbf{x}_i) + \frac{1}{2}\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)\boldsymbol{\Sigma}_x]$$

so that the approximate mean is

$$m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \mu(\mathbf{u}) + \frac{1}{2}\sum_{i=1}^{N}\beta_i \mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)\boldsymbol{\Sigma}_x] \tag{13}$$

where $\mu(\mathbf{u}) = \sum_{i=1}^{N}\beta_i C(\mathbf{u}, \mathbf{x}_i)$ is the noise-free predictive mean computed at $\mathbf{u}$.

3.2 Approximate Variance

Similarly, the approximate variance is

$$v^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = l^{ap} - \sum_{i,j=1}^{N}(K_{ij}^{-1} - \beta_i\beta_j)l_{ij}^{ap} - m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2$$

with $l^{ap} = E_{\mathbf{x}}[C^{ap}(\mathbf{x}, \mathbf{x})]$ and $l_{ij}^{ap} = E_{\mathbf{x}}[C^{ap}(\mathbf{x}, \mathbf{x}_i)C^{ap}(\mathbf{x}, \mathbf{x}_j)]$, where $C^{ap}(.,.)$ is again the second order Taylor approximation of $C(.,.)$. We have

$$l^{ap} = C(\mathbf{u}, \mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{u})\boldsymbol{\Sigma}_x]$$

and

$$\begin{aligned} l_{ij}^{ap} \approx & C(\mathbf{u}, \mathbf{x}_i)C(\mathbf{u}, \mathbf{x}_j) + \mathrm{Tr}[\mathbf{C}'(\mathbf{u}, \mathbf{x}_i)\mathbf{C}'(\mathbf{u}, \mathbf{x}_j)^T\boldsymbol{\Sigma}_x] + \frac{1}{2}C(\mathbf{u}, \mathbf{x}_i)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_j)\boldsymbol{\Sigma}_x] \\ & + \frac{1}{2}C(\mathbf{u}, \mathbf{x}_j)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)\boldsymbol{\Sigma}_x] \end{aligned}$$

where the approximation comes from discarding terms of higher order than $\boldsymbol{\Sigma}_x$ in $C^{ap}(\mathbf{x}, \mathbf{x}_i)C^{ap}(\mathbf{x}, \mathbf{x}_j)$, as discussed in the previous section. Similarly, approximating $m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2$ by

$$\begin{aligned} m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \approx \sum_{i,j=1}^{N}\beta_i\beta_j \Bigg(& C(\mathbf{u}, \mathbf{x}_i)C(\mathbf{u}, \mathbf{x}_j) + \frac{1}{2}C(\mathbf{u}, \mathbf{x}_i)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_j)\boldsymbol{\Sigma}_x] \\ & + \frac{1}{2}C(\mathbf{u}, \mathbf{x}_j)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)\boldsymbol{\Sigma}_x]\Bigg) \ , \end{aligned}$$

we find, after simplifications,

$$\begin{aligned} v^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = & \sigma^2(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{u})\boldsymbol{\Sigma}_x] - \sum_{i,j=1}^{N}(K_{ij}^{-1} - \beta_i\beta_j)\mathrm{Tr}[\mathbf{C}'(\mathbf{u}, \mathbf{x}_i)\mathbf{C}'(\mathbf{u}, \mathbf{x}_j)^T\boldsymbol{\Sigma}_x] \\ & - \frac{1}{2}\sum_{i,j=1}^{N}K_{ij}^{-1}(C(\mathbf{u}, \mathbf{x}_i)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_j)\boldsymbol{\Sigma}_x] + C(\mathbf{u}, \mathbf{x}_j)\mathrm{Tr}[\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)\boldsymbol{\Sigma}_x]) \end{aligned} \tag{14}$$

where $\sigma^2(\mathbf{u}) = C(\mathbf{u},\mathbf{u}) - \sum_{i,j=1}^N K_{ij}^{-1} C(\mathbf{u},\mathbf{x}_i)C(\mathbf{u},\mathbf{x}_j)$ is the noise-free predictive variance.

Note that these results might be more easily derived by approximating $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ directly in Eqs. (4) and (5), as done in [5, 7].[7] Applying (10) to $\mu(\mathbf{x})$, we have $E[\mu(\mathbf{x})] \simeq \mu(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\boldsymbol{\mu}''(\mathbf{u})\boldsymbol{\Sigma}_x]$, and replacing into (4) gives

$$m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \mu(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\boldsymbol{\mu}''(\mathbf{u})\boldsymbol{\Sigma}_x] \ .$$

Similarly, $E[\sigma^2(\mathbf{x})] \simeq \sigma^2(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\boldsymbol{\sigma}^{2''}(\mathbf{u})\boldsymbol{\Sigma}_x]$ and, using (11), $\mathrm{Var}[\mu(\mathbf{x})] \simeq \mathrm{Tr}[\boldsymbol{\mu}'(\mathbf{u})\boldsymbol{\mu}'(\mathbf{u})^T\boldsymbol{\Sigma}_x]$. Replacing into (5) we obtain

$$v^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sigma^2(\mathbf{u}) + \mathrm{Tr}\left[\left(\frac{1}{2}\boldsymbol{\sigma}^{2''}(\mathbf{u}) + \boldsymbol{\mu}'(\mathbf{u})\boldsymbol{\mu}'(\mathbf{u})^T\right)\boldsymbol{\Sigma}_x\right] \ .$$

Although, replacing the derivatives by their expressions, these results are obviously the same as those obtained when working with the covariance function, working directly with $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ lacks flexibility in that it is not clear that exact moments can be computed.

Both approximate mean and variance are composed of the noise-free predictive moments plus correction terms. Assuming $\boldsymbol{\Sigma}_x$ is diagonal, these correction terms consist of the sum of the derivatives of the covariance function in each input dimension, weighted by the variance of the new test input in the same direction. Figure 5 illustrates these results. On the x-axis, the asterisks indicate the observed noisy inputs and the distribution they come from ($p(x) = \mathcal{N}_x(u, v_x)$, for $u = 2, 6, 9.5$ and $v_x = 1$). The circles indicate the function output corresponding to the noise-free u's. The approximate means $m^{ap}(u, v_x)$ and associated uncertainties, $\pm 2\sqrt{v^{ap}}(u, v_x)$ are plotted as triangles and dotted lines. We can compare them to the *naive* (noise-free) means $\mu(u)$ with error-bars $\pm 2\sigma(u)$, which do not account for the noise on the input.

4 *Gaussian Approximation*: Exact Moments

We are now going to show that in the special cases of the linear and the Gaussian (squared exponential) covariance functions, we can evaluate integrals (7)-(9) exactly.

4.1 Case of the Linear Covariance Function

Let us write the linear covariance function as $C_L(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{L} \mathbf{x}_j$ where $\mathbf{L} = \mathrm{diag}[\alpha_1 \ldots \alpha_D]$. In the noise-free case, the prediction at $\mathbf{u}$ leads to a Gaussian distribution with mean and variance

[7] In [5, 7], we only considered a first order approximation for the mean $\mu(\mathbf{x})$.

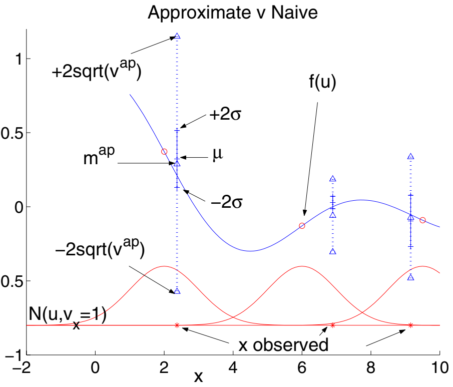

Fig. 5. *Gaussian approximation* to the prediction at x (asterisk), noisy version of the true $u = 2, 6, 9.5$, where the noise has variance and $v_x = 1$. The approximate mean and uncertainty $(m^{ap}(x) \pm 2\sqrt{v^{ap}(x)})$ are indicated by triangles and the noise-free moments $(\mu(x) \pm 2\sigma(x))$ by crosses. The circles show the function outputs corresponding to the noise-free u's.

$$\begin{cases} \mu_L(\mathbf{u}) = \sum_{i=1}^{N} \beta_i C_L(\mathbf{u}, \mathbf{x}_i) \\ \sigma_L^2(\mathbf{u}) = C_L(\mathbf{u}, \mathbf{u}) - \sum_{i,j=1}^{N} K_{ij}^{-1} C_L(\mathbf{u}, \mathbf{x}_i) C_L(\mathbf{u}, \mathbf{x}_j) \ . \end{cases} \tag{15}$$

When predicting at a noisy input, the predictive mean and variance are now given by

$$m^{ex_L}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i l_i^{ex_L} \tag{16}$$

$$v^{ex_L}(\mathbf{u}, \boldsymbol{\Sigma}_x) = l^{ex_L} - \sum_{i,j=1}^{N} (K_{ij}^{-1} - \beta_i \beta_j) l_{ij}^{ex_L} - m^{ex_L}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \tag{17}$$

so that we need to evaluate

$$\begin{aligned} l^{ex_L} &= E_{\mathbf{x}}[C_L(\mathbf{x}, \mathbf{x})] = \int \mathbf{x}^T \mathbf{L} \mathbf{x} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \\ l_i^{ex_L} &= E_{\mathbf{x}}[C_L(\mathbf{x}, \mathbf{x}_i)] = \int \mathbf{x}^T \mathbf{L} \mathbf{x}_i \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \\ l_{ij}^{ex_L} &= E_{\mathbf{x}}[C_L(\mathbf{x}, \mathbf{x}_i) C_L(\mathbf{x}, \mathbf{x}_j)] = \int \mathbf{x}^T \mathbf{L} \mathbf{x}_i \mathbf{x}^T \mathbf{L} \mathbf{x}_j \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \ . \end{aligned}$$

Using the formula giving the expectation of a quadratic form under a Gaussian[8] we directly obtain

[8]

$$\int_{\mathbf{x}} (\mathbf{x} - \mathbf{m})^T \mathbf{M}^{-1} (\mathbf{x} - \mathbf{m}) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} = (\mathbf{m} - \mathbf{u})^T \mathbf{M}^{-1} (\mathbf{m} - \mathbf{u}) + \mathrm{Tr}[\mathbf{M}^{-1} \boldsymbol{\Sigma}_x]$$

$$
\begin{aligned}
l^{ex_L} &= \mathbf{u}^T L\mathbf{u} + \mathrm{Tr}[\mathbf{L}\boldsymbol{\Sigma}_x] = C_L(\mathbf{u},\mathbf{u}) + \mathrm{Tr}[\mathbf{L}\boldsymbol{\Sigma}_x] \\
l_i^{ex_L} &= \mathbf{u}^T \mathbf{L}\mathbf{x}_i = C_L(\mathbf{u},\mathbf{x}_i) \\
l_{ij}^{ex_L} &= \mathbf{u}^T(\mathbf{L}\mathbf{x}_i\mathbf{x}_j^T\mathbf{L})\mathbf{u} + \mathrm{Tr}[\mathbf{L}\mathbf{x}_i\mathbf{x}_j^T\mathbf{L}\boldsymbol{\Sigma}_x] = C_L(\mathbf{u},\mathbf{x}_i)C_L(\mathbf{x}_j,\mathbf{u}) + \mathrm{Tr}[\mathbf{L}\mathbf{x}_i\mathbf{x}_j^T\mathbf{L}\boldsymbol{\Sigma}_x] \ .
\end{aligned}
$$

Therefore, the predictive mean is the same as the noise-free one, as we have

$$m^{ex_L}(\mathbf{u},\boldsymbol{\Sigma}_x) = \sum_{i=1}^N \beta_i C_L(\mathbf{u},\mathbf{x}_i) \ . \tag{18}$$

On the other hand, the variance becomes

$$
\begin{aligned}
v^{ex_L}(\mathbf{u},\boldsymbol{\Sigma}_x) = C_L(\mathbf{u},\mathbf{u}) + \mathrm{Tr}[\mathbf{L}\boldsymbol{\Sigma}_x] - \sum_{i,j=1}^N (K_{ij}^{-1} - \beta_i\beta_j)\mathrm{Tr}[\mathbf{L}\mathbf{x}_i\mathbf{x}_j^T\mathbf{L}\boldsymbol{\Sigma}_x]) \\
- \sum_{i,j=1}^N K_{ij}^{-1} C_L(\mathbf{u},\mathbf{x}_i)C_L(\mathbf{x}_j,\mathbf{u})
\end{aligned} \tag{19}
$$

after simplification of the $\beta_i\beta_j$ terms. Or, in terms of the noise-free variance,

$$v^{ex_L}(\mathbf{u},\boldsymbol{\Sigma}_x) = \sigma_L^2(\mathbf{u}) + \mathrm{Tr}[\mathbf{L}\boldsymbol{\Sigma}_x] - \sum_{i,j=1}^N (K_{ij}^{-1} - \beta_i\beta_j)\mathrm{Tr}[\mathbf{L}\mathbf{x}_i\mathbf{x}_j^T\mathbf{L}\boldsymbol{\Sigma}_x]) \ . \tag{20}$$

If we note that $\mathbf{C}_L'(\mathbf{u},\mathbf{x}_i) = \frac{\partial C_L(\mathbf{u},\mathbf{x}_i)}{\partial \mathbf{u}} = \mathbf{L}\mathbf{x}_i$ and $\mathbf{C}_L''(\mathbf{u},\mathbf{u}) = \frac{\partial^2 C_L(\mathbf{u},\mathbf{u})}{\partial\mathbf{u}\partial\mathbf{u}^T} = 2\mathbf{L}$, we can also write it as

$$
\begin{aligned}
v^{ex_L}(\mathbf{u},\boldsymbol{\Sigma}_x) = \sigma_L^2(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[\mathbf{C}_L''(\mathbf{u},\mathbf{u})\boldsymbol{\Sigma}_x] \\
- \sum_{i,j=1}^N (K_{ij}^{-1} - \beta_i\beta_j)\mathrm{Tr}[\mathbf{C}_L'(\mathbf{x},\mathbf{x}_i)\mathbf{C}_L'(\mathbf{x},\mathbf{x}_j)^T\boldsymbol{\Sigma}_x]) \ .
\end{aligned} \tag{21}
$$

As we would expect, the predictive mean and variance in the case of the linear covariance function correspond to the approximate moments we would obtain within a first order approximation of the covariance function.

4.2 Case of the Gaussian Covariance Function

The Gaussian (or squared exponential) covariance function became a popular choice especially after Rasmussen demonstrated that a GP with such a covariance function performed as well, if not better, than other models like neural networks [10]. It is usually expressed as

$$C_G(\mathbf{x}_i,\mathbf{x}_j) = v\exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{W}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right] \tag{22}$$

with $\mathbf{W}^{-1} = \mathrm{diag}[w_1 \dots w_D]$, where w_d is a roughness parameter, inversely proportional to the square of the correlation length in direction d ($w_d = 1/\lambda_d^2$), which represents the length along which successive values are strongly correlated (with a role similar to the Automatic Relevance Determination tool of Mackay and Neal [11, 12]). The parameter v controls the overall vertical scale relative to the zero mean of the process in the output space (the vertical amplitude of variation of a typical function).

We now denote by $\mu_G(\mathbf{u})$ and $\sigma^2_G(\mathbf{u})$ the corresponding noise-free predictive mean and variance,

$$\begin{cases} \mu_G(\mathbf{u}) = \sum_{i=1}^{N} \beta_i C_G(\mathbf{u}, \mathbf{x}_i) \\ \sigma^2_G(\mathbf{u}) = C_G(\mathbf{u}, \mathbf{u}) - \sum_{i,j=1}^{N} K^{-1}_{ij} C_G(\mathbf{u}, \mathbf{x}_i) C_G(\mathbf{u}, \mathbf{x}_j) \end{cases} \tag{23}$$

where $C_G(\mathbf{x}, \mathbf{x}) = v$. In this case, the predictive mean and variance, obtained for a prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$, are given by

$$m^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i l_i^{ex_G} \tag{24}$$

$$v^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x) = l^{ex_G} - \sum_{i,j=1}^{N} (K^{-1}_{ij} - \beta_i\beta_j) l^{ex_G}_{ij} - m^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \ . \tag{25}$$

We directly have $l^{ex_G} = E_{\mathbf{x}}[C_G(\mathbf{x}, \mathbf{x})] = v = C_G(\mathbf{u}, \mathbf{u})$, and we need to evaluate

$$\begin{aligned} l_i^{ex_G} &= E_{\mathbf{x}}[C_G(\mathbf{x}, \mathbf{x}_i)] = c \int N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \\ l_{ij}^{ex_G} &= E_{\mathbf{x}}[C_G(\mathbf{x}, \mathbf{x}_i) C_G(\mathbf{x}, \mathbf{x}_j)] = c^2 \int N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) \mathcal{N}_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) d\mathbf{x} \ , \end{aligned}$$

where, for notational convenience, we write the Gaussian covariance function as[9] $C_G(\mathbf{x}_i, \mathbf{x}_j) = c N_{\mathbf{x}_i}(\mathbf{x}_j, \mathbf{W})$, with $c = (2\pi)^{D/2} |\mathbf{W}|^{1/2} v$. Using the product of Gaussians formula,[10] we find

$$l_i^{ex_G} = c N_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x) \ . \tag{26}$$

And for $l_{ij}^{ex_G}$, using this product twice,

$$l_{ij}^{ex_G} = c^2 N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \boldsymbol{\Sigma}_x + \frac{\mathbf{W}}{2} \right) \ . \tag{27}$$

[9] Note that $N(.,.)$ is used to denote the parametric form of the function, it does not correspond to a normal probability distribution $\mathcal{N}(.,.)$.

[10] Recall that $\mathcal{N}_x(a, A)\mathcal{N}_x(b, B) = z\mathcal{N}_x(d, D)$ with $D = (A^{-1} + B^{-1})^{-1}$, $d = D(A^{-1}a + B^{-1}b)$ and $z = \mathcal{N}_a(b, A + B) = \mathcal{N}_b(a, A + B)$.

Exact Predictive Mean Replacing l_i^{exG} in $m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x)$, we have

$$m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i c N_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x) \tag{28}$$

and we can directly check that, as we would expect, $m(\mathbf{u}, \boldsymbol{\Sigma}_x = \mathbf{0}) = \mu_G(\mathbf{u})$.

It is useful to write $m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ as a *corrected* version of $\mu_G(\mathbf{u})$. Using the matrix inversion lemma, we have $(\mathbf{W} + \boldsymbol{\Sigma}_x)^{-1} = \mathbf{W}^{-1} - \mathbf{W}^{-1}(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1}\mathbf{W}^{-1}$, leading to

$$l_i^{exG} = C_G(\mathbf{u}, \mathbf{x}_i) C_{corr}(\mathbf{u}, \mathbf{x}_i) \tag{29}$$

with

$$C_{corr}(\mathbf{u}, \mathbf{x}_i) = |\mathbf{I} + \mathbf{W}^{-1}\boldsymbol{\Sigma}_x|^{-1/2} \exp\left[\frac{1}{2}(\mathbf{u} - \mathbf{x}_i)^T \Delta^{-1}(\mathbf{u} - \mathbf{x}_i)\right] \tag{30}$$

where $\Delta^{-1} = \mathbf{W}^{-1}(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1}\mathbf{W}^{-1}$. The predictive mean is then given by

$$m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sum_{i=1}^{N} \beta_i C_G(\mathbf{u}, \mathbf{x}_i) C_{corr}(\mathbf{u}, \mathbf{x}_i) \ . \tag{31}$$

Compared to the noise-free $\mu_G(\mathbf{u})$, the covariances between the new noisy input and the training inputs, formerly given by $C_G(\mathbf{u}, \mathbf{x}_i)$, are now weighted by $C_{corr}(\mathbf{u}, \mathbf{x}_i)$, thus accounting for the uncertainty associated to $\mathbf{u}$.

Exact Predictive Variance Replacing l_{ij}^{exG} by its expression, we have

$$\begin{aligned} v^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x) = C_G(\mathbf{u}, \mathbf{u}) - c^2 \sum_{i,j=1}^{N} (K_{ij}^{-1} - \beta_i\beta_j) N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \boldsymbol{\Sigma}_x + \frac{\mathbf{W}}{2}\right) \\ -m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 \end{aligned}$$

and again, we can show that for $\boldsymbol{\Sigma}_x = \mathbf{0}$, we have $v^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x = \mathbf{0}) = \sigma_G^2(\mathbf{u})$.[11]

[11] We have

$$\begin{aligned} v^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x = \mathbf{0}) = C_G(\mathbf{u}, \mathbf{u}) - c^2 \sum_{i,j=1}^{N} (K_{ij}^{-1} - \beta_i\beta_j) N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2}\right) \\ -m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x = \mathbf{0})^2 \end{aligned}$$

with $m^{exG}(\mathbf{u}, \boldsymbol{\Sigma}_x = \mathbf{0})^2 = c^2 \sum_{i,j=1}^{N} \beta_i\beta_j N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \frac{\mathbf{W}}{2}\right)$, to be compared to the noise-free predictive variance that we can write $\sigma_G^2(\mathbf{u}) = C_G(\mathbf{u}, \mathbf{u}) - c^2 \sum_{i,j=1}^{N} K_{ij}^{-1} N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \frac{\mathbf{W}}{2}\right)$, using $N_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{u}}(\mathbf{x}_j, \mathbf{W}) = N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \frac{\mathbf{W}}{2}\right)$.

As done for the predictive mean, we can find another form for $v^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ where the Gaussian covariance function appears weighted by a correction term. It can be shown that we can write $l_{ij}^{ex_G}$ as

$$l_{ij}^{ex_G} = C_G(\mathbf{u}, \mathbf{x}_i) C_G(\mathbf{u}, \mathbf{x}_j) C_{corr_2}(\mathbf{u}, \bar{\mathbf{x}})$$

where $\bar{\mathbf{x}} = \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$ and

$$C_{corr_2}(\mathbf{u}, \bar{\mathbf{x}}) = \left| \left(\frac{\mathbf{W}}{2} \right)^{-1} \boldsymbol{\Sigma}_x + \mathbf{I} \right|^{-1/2} \exp\left[\frac{1}{2} (\mathbf{u} - \bar{\mathbf{x}})^T \Lambda^{-1} (\mathbf{u} - \bar{\mathbf{x}}) \right] \tag{32}$$

with $\Lambda^{-1} = \left(\frac{\mathbf{W}}{2}\right)^{-1} \left(\left(\frac{\mathbf{W}}{2}\right)^{-1} + \boldsymbol{\Sigma}_x^{-1} \right)^{-1} \left(\frac{\mathbf{W}}{2}\right)^{-1}$.

In terms of $\sigma_G^2(\mathbf{u})$, we can then write

$$\begin{aligned} v^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \sigma_G^2(\mathbf{u}) + \sum_{i,j=1}^{N} K_{ij}^{-1} C_G(\mathbf{u}, \mathbf{x}_i) C_G(\mathbf{u}, \mathbf{x}_j) (1 - C_{corr_2}(\mathbf{u}, \bar{\mathbf{x}})) \\ + \sum_{i,j=1}^{N} \beta_i \beta_j C_G(\mathbf{u}, \mathbf{x}_i) C_G(\mathbf{u}, \mathbf{x}_j) (C_{corr_2}(\mathbf{u}, \bar{\mathbf{x}}) - C_{corr}(\mathbf{u}, \mathbf{x}_i) C_{corr}(\mathbf{u}, \mathbf{x}_j)) \;, \end{aligned} \tag{33}$$

where we have used
$m^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x)^2 = \sum_{i,j=1}^{N} \beta_i \beta_j C_G(\mathbf{u}, \mathbf{x}_i) C_{corr}(\mathbf{u}, \mathbf{x}_i) C_G(\mathbf{u}, \mathbf{x}_j) C_{corr}(\mathbf{u}, \mathbf{x}_j)$.

Although we will not give the details of the calculations here, it can be shown that these predictive mean and variance tend to the approximate mean and variance presented in section 3 when $\boldsymbol{\Sigma}_x$ tends to zero (so that we can approximate e^x by $1 + x$). As Figure 5 for the approximate moments, Figure 6 shows the exact predictive mean and error-bars (triangles) obtained when predicting at noisy inputs (asterisks).

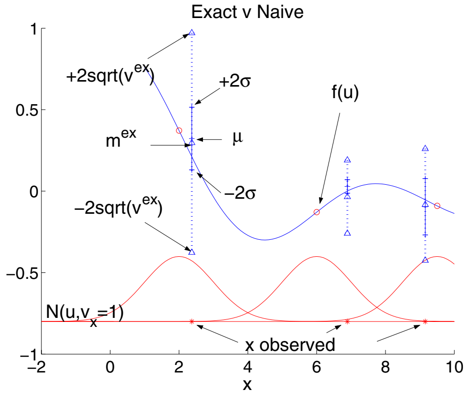

Fig. 6. As in Fig. 5, the triangles now indicate the exact predictive means with their error-bars, accounting for the uncertainty on the noisy inputs (asterisks).

5 Iterative *k*-step Ahead Prediction

Using the results derived in the previous sections, we now derive an algorithm for propagating the uncertainty as we predict ahead in time the output of a nonlinear dynamic system, represented by on one-step ahead non-linear auto-regressive (NAR) model.

At this point, it might be useful to recall the different notations used, depending on the situation, as done in Table 1. It is important not to forget that the predictive distribution corresponding to a noise-free $\mathbf{u}$ is Gaussian but it is not when predicting at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$. We only compute its mean and variance, which is done exactly when the covariance function is e.g. Gaussian or linear, or approximately, in the *general* case.

Table 1. Notation used, depending on the type of covariance function (left column) and whether the prediction is at a noise-free or a noisy input. ('Where' in the document the corresponding equations can be found is indicated in small fonts.)

Covariance function	Prediction at $\mathbf{u}$	Prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$
General	$\mu(\mathbf{u})$, $\sigma^2(\mathbf{u})$ Eqs. (1), at $\mathbf{u}$	$m^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (13) $v^{ap}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (14)
Linear	$\mu_L(\mathbf{u})$, $\sigma_L^2(\mathbf{u})$ Eqs. (15)	$m^{ex_L}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (18) $v^{ex_L}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (21)
Gaussian	$\mu_G(\mathbf{u})$, $\sigma_G^2(\mathbf{u})$ Eqs. (23)	$m^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (31) $v^{ex_G}(\mathbf{u}, \boldsymbol{\Sigma}_x)$ Eq. (33)

5.1 Background

Given a discrete one-dimensional time-series $y_1, \ldots, y_t$, we wish to predict its value at, say, time $t + k$. Viewing the observed time-series as a projection of the dynamics of the underlying system, which lie in a higher dimensional space [13], we consider the following non-linear auto-regressive (NAR) model

$$y_{t+1} = f(\mathbf{x}_t) \quad \text{with} \quad \mathbf{x}_t = [y_t, y_{t-1}, \ldots, y_{t-L}]^T \ , \tag{34}$$

whose order, L, corresponds to the dimension of the reconstructed space (number of delayed outputs, called *lag* or *embedding dimension*). The state (or input) at time t is $\mathbf{x}_t$ and y_{t+1} is the corresponding output. Note that in practise, y_{t+1} is alone considered as noisy ($y_{t+1} = f(\mathbf{x}_t) + \epsilon_{t+1}$). Here, we simply assume that ϵ_{t+1} is a white noise but colored noise models can also be considered, as in [14].

Using this one-step ahead model, the iterative k-step ahead prediction task can be thought of as a missing or noisy data modelling problem[12] since what

[12] The missing variables can be seen as noisy variables for complete noise.

we want is to predict y_{t+k}, when y_{t+k-1} down to y_t are missing, provided the time-series is known up to time t. This problem has been the scope of much research (see e.g. [15, 16]) but has not yet been addressed for the GP model. A naive way of solving the iterative multiple-step ahead prediction task is simply to substitute a single value to the missing value (say the value of the time-series at another time-step, or a maximum likelihood estimate) but this approach has been shown not to be optimal and to lead to biased predictions [17, 15]. In [18], long-term predictions are improved by eliminating the systematic errors induced by each successive short term prediction, by considering a function of the estimates.

Using our approximation for the prediction at a noisy input, we suggest to incorporate the uncertainty about intermediate regressor values as we predict ahead in time. This results in an update of the uncertainty on the current prediction and therefore an improvement of each successive predictions.

5.2 Propagation of Uncertainty Algorithm

We assume that a zero-mean GP model was trained to minimize the one-step ahead predictions of a time-series known up to time t. By propagating the uncertainty as we predict ahead in time, we mean that for $y_{t+k} = f(y_{t+k-1}, \dots, y_{t+k-L})$, we consider the delayed $y_{t+k-1}, \dots, y_{t+k-L}$ as Gaussian random variables, with mean $m(.,.)$ and variance $v(.,.)$, computed either approximately or exactly, depending on the covariance function of the process.

Here is a sketch of how we proceed:

- Time $t+1$, $\mathbf{x}_{t+1} = [y_t, \dots, y_{t-L}]^T$: Since the state is formed on known values of the time-series, we simply have $y_{t+1} \sim \mathcal{N}(\mu(\mathbf{x}_{t+1}, \sigma^2(\mathbf{x}_{t+1}))$.
- Time $t+2$, $\mathbf{x}_{t+2} = [y_{t+1}, y_t, \dots, y_{t+1-L}]^T \sim \mathcal{N}(\mathbf{u}_{t+2}, \boldsymbol{\Sigma}_{t+2})$ with

$$\mathbf{u}_{t+2} = \begin{bmatrix} \mu(\mathbf{x}_{t+1}) \\ y_t \\ \vdots \\ y_{t+1-L} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_{t+2} = \begin{bmatrix} \sigma^2(\mathbf{x}_{t+1}) & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} .$$

  Within our analytical approximation, we only compute the mean and variance of y_{t+2} and consider $y_{t+2} \sim \mathcal{N}(m(\mathbf{u}_{t+2}, \boldsymbol{\Sigma}_{t+2}), v(\mathbf{u}_{t+2}, \boldsymbol{\Sigma}_{t+2}))$.
- Time $t+3$, $\mathbf{x}_{t+3} = [y_{t+2}, y_{t+1}, \dots, y_{t+2-L}]^T \sim \mathcal{N}(\mathbf{u}_{t+3}, \boldsymbol{\Sigma}_{t+3})$ with

$$\mathbf{u}_{t+3} = \begin{bmatrix} m(x_{t+2}) \\ \mu(x_{t+1}) \\ y_t \\ \vdots \\ y_{t+2-L} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_{t+3} = \begin{bmatrix} v(x_{t+2}) & \mathrm{Cov}[y_{t+2}, y_{t+1}] & 0 & \dots & 0 \\ \mathrm{Cov}[y_{t+1}, y_{t+2}] & \sigma^2(x_{t+1}) & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} .$$

  Compute $y_{t+3} \sim \mathcal{N}(m(\mathbf{u}_{t+3}, \boldsymbol{\Sigma}_{t+3}), v(\mathbf{u}_{t+3}, \boldsymbol{\Sigma}_{t+3}))$.

Repeating this procedure up to the desired horizon k, and assuming $k > L$, at $t+k$, we have $\mathbf{x}_{t+k} = [y_{t+k-1}, y_{t+k-2}, \dots, y_{t+k-L}]^T \sim \mathcal{N}(\mathbf{u}_{t+k}, \boldsymbol{\Sigma}_{t+k})$ and compute $y_{t+k} \sim \mathcal{N}(m(\mathbf{u}_{t+k}, \boldsymbol{\Sigma}_{t+k}), v(\mathbf{u}_{t+k}, \boldsymbol{\Sigma}_{t+k}))$. The input mean is then given by

$$\mathbf{u}_{t+k} = [m(x_{t+k-1}), m(x_{t+k-2}), \dots, m(x_{t+k-L})]^T$$

and the input covariance matrix is

$$\boldsymbol{\Sigma}_{t+k} = \begin{bmatrix} v(x_{t+k-1}) & \mathrm{Cov}[y_{t+k-1}, y_{t+k-2}] & \dots & \mathrm{Cov}[y_{t+k-1}, y_{t+k-L}] \\ \mathrm{Cov}[y_{t+k-2}, y_{t+k-1}] & v(x_{t+k-2}) & \dots & \mathrm{Cov}[y_{t+k-2}, y_{t+k-L}] \\ \dots & \dots & \dots & \dots \\ \mathrm{Cov}[y_{t+k-L}, y_{t+k-1}] & \mathrm{Cov}[y_{t+k-L}, y_{t+k-2}] & \dots & v(x_{t+k-L}) \end{bmatrix} .$$

We now need to compute the cross-covariance terms: In general, at time $t+l$, we have the random input vector $\mathbf{x}_{t+l} = [y_{t+l-1}, \dots, y_{t+l-L}]^T \sim \mathcal{N}(\mathbf{u}_{t+l}, \boldsymbol{\Sigma}_{t+l})$. The $L \times L$ covariance matrix $\boldsymbol{\Sigma}_{t+l}$ has the delayed predictive variances on its diagonal and the cross-covariance terms correspond to the covariances between y_{t+l-i} and y_{t+l-j}, for $i, j = 1 \dots L$ with $i \neq j$. Discarding the last (*oldest*) element of $\mathbf{x}_{t+l}$, we need to compute $\mathrm{Cov}[y_{t+l-i}, y_{t+l-j}] = \mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}]$, that is

$$\mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}] = E[y_{t+l}\mathbf{x}_{t+l}] - E[y_{t+l}]E[\mathbf{x}_{t+l}] \tag{35}$$

where $E[y_{t+l}] = m(\mathbf{u}_{t+l}, \boldsymbol{\Sigma}_{t+l})$ and $E[\mathbf{x}_{t+l}] = \mathbf{u}_{t+l}$. For the expectation of the product, we have

$$\begin{aligned} E[y_{t+l}\mathbf{x}_{t+l}] &= \int\int y_{t+l}\mathbf{x}_{t+l} p(y_{t+l}, \mathbf{x}_{t+l}) dy_{t+l} d\mathbf{x}_{t+l} \\ &= \int\int y_{t+l}\mathbf{x}_{t+l} p(y_{t+l}|\mathbf{x}_{t+l}) p(\mathbf{x}_{t+l}) dy_{t+l} d\mathbf{x}_{t+l} \end{aligned}$$

and since $\int y_{t+l} p(y_{t+l}|\mathbf{x}_{t+l}) dy_{t+l} = \mu(\mathbf{x}_{t+l})$, we can write

$$E[y_{t+l}\mathbf{x}_{t+l}] = \int \mathbf{x}_{t+l}\mu(\mathbf{x}_{t+l}) p(\mathbf{x}_{t+l}) d\mathbf{x}_{t+l} \ .$$

Replacing $\mu(\mathbf{x}_{t+l})$ by its expression, we have

$$E[y_{t+l}\mathbf{x}_{t+l}] = \sum_i \beta_i \int \mathbf{x}_{t+l} C(\mathbf{x}_{t+l}, \mathbf{x}_i) p(\mathbf{x}_{t+l}) d\mathbf{x}_{t+l} \ . \tag{36}$$

Depending on the form of $C(.,.)$, we evaluate this integral exactly or approximately. Denoting $\mathbf{x}_{t+l}$ by $\mathbf{x}$ for notational convenience, let $I_i = \int \mathbf{x} C(\mathbf{x}, \mathbf{x}_i) p(\mathbf{x}) d\mathbf{x}$ be the integral we wish to solve.

Gaussian Case In the case of the Gaussian covariance function, we have $m(.,.) = m^{ex_G}(.,.)$ and $v(.,.) = v^{ex_G}(.,.)$, as given by Eqs. (31) and (33).

Using a similar notation as in section 4.2, we need to solve

$$I_i^{ex_G} = c \int \mathbf{x} N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) p(\mathbf{x}) d\mathbf{x}$$

where $C(\mathbf{x}, \mathbf{x}_i) = cN_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})$, with $c = (2\pi)^{D/2}|\mathbf{W}|^{1/2}v$. As before, using the product of Gaussians, we find

$$I_i^{ex_G} = cN_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x)[(\mathbf{I} + \mathbf{W}\boldsymbol{\Sigma}_x^{-1})^{-1}\mathbf{x}_i + (\mathbf{I} + \boldsymbol{\Sigma}_x\mathbf{W}^{-1})^{-1}\mathbf{u}]$$

where $cN_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x) = C(\mathbf{u}, \mathbf{x}_i)C_{corr}(\mathbf{u}, \mathbf{x}_i)$, with $C_{corr}(\mathbf{u}, \mathbf{x}_i)$ given by (30).

We can then write

$$E[y_{t+l}\mathbf{x}_{t+l}] = \sum_i \beta_i C(\mathbf{u}_{t+l}, \mathbf{x}_i)C_{corr}(\mathbf{u}_{t+l}, \mathbf{x}_i)[(\mathbf{I}+\mathbf{W}\boldsymbol{\Sigma}_{t+l)}^{-1})^{-1}\mathbf{x}_i + (\mathbf{I}+\boldsymbol{\Sigma}_{t+l}\mathbf{W}^{-1})^{-1}\mathbf{u}_{t+l}] \, .$$

After simplifications, the cross-covariance terms are given by

$$\mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}] = \sum_i \beta_i C(\mathbf{u}_{t+l}, \mathbf{x}_i)C_{corr}(\mathbf{u}_{t+l}, \mathbf{x}_i)(\mathbf{I} + \mathbf{W}\boldsymbol{\Sigma}_{t+l)}^{-1})^{-1}\mathbf{x}_i \, . \qquad (37)$$

General Case When the covariance function is such that approximations are needed, the predictive mean and variance corresponding to a noisy input are given by $m(.,.) = m^{ap}(.,.)$, using Eq. (13) and $v(.,.) = v^{ap}(.,.)$, using (14).

For the computation of the cross-covariances, we resort to a second order Taylor approximation of the covariance function, as in section 3. We then have[13]

$$I_i^{ap} \approx \mathbf{u}^T C(\mathbf{u}, \mathbf{x}_i) + \mathbf{C}'(\mathbf{u}, \mathbf{x}_i)^T\boldsymbol{\Sigma}_x + \frac{1}{2}\mathbf{u}^T \mathrm{Tr}[\boldsymbol{\Sigma}_x\mathbf{C}''(\mathbf{u}, \mathbf{x}_i)] \, .$$

After simplifications, we obtain the following expression for the cross-covariance terms

$$\mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}] = \sum_i \beta_i \mathbf{C}'(\mathbf{u}_{t+l}, \mathbf{x}_i)^T\boldsymbol{\Sigma}_{t+l}. \qquad (38)$$

6 Numerical Examples

For clarity, we will denote the different approaches as follows:

- *MC*, for the Monte-Carlo approximation to the true predictive distribution corresponding to a noisy input;
- *A*, for the *Gaussian approximation* that computes only the mean and variance of this distribution, and specifically A_{ap} when these moments are computed using the Taylor approximation, and A_{ex} when they are computed exactly;

[13] This result was obtained by extending the one-dimensional case to L-dimensions. In 1D, we have

$$\begin{aligned} I_i^{ap} &\approx \int x \left(C(u, x_i) + (x-u)C'(u, x_i) + \frac{1}{2}(x-u)^2 C''(u, x_i) \right) p(x)dx \\ &\approx uC(u, x_i) + v_x C'(u, x_i) + \frac{1}{2}uv_x C''(u, x_i) \end{aligned}$$

where we have used $\int x^2 p(x)dx = v_x + u^2$ and $\int x^3 p(x)dx = 3uv_x + u^3$.

– N, for the *naive* predictive mean and variances that do not account for the noise on the input.

We assess the performance of the different methods by computing the average squared error (E_1), over the test set, and average minus log predictive density (E_2), which measures the density of the actual true test output under the Gaussian predictive distribution and use its negative log as a measure of loss. To assess the performance of the Monte-Carlo approximation, we compute the squared error and minus log-likelihood loss for the predictions given by each sample and average over the number of samples. We also compute the average predictive mean (sample mean) and average predictive variance (sample variance) and compute the associated losses.

6.1 A Simple Comparison on a Static Example

On the static example previously used, we compare the different approaches for the prediction at a noisy input, when the true noise-free input is 2 (left) and 6 (right) and the input noise variance is 1. Figure 7 shows the predictive distribution given by MC (continuous), N (dashed), A_{ap} (dots) and A_{ex} (asterisks). Note how the naive approach leads to a narrow distribution (N), peaked around its mean value, since it does not account for the uncertainty on the input. The Monte-Carlo approximation to the true distribution highlights how the true distribution is non-Gaussian.

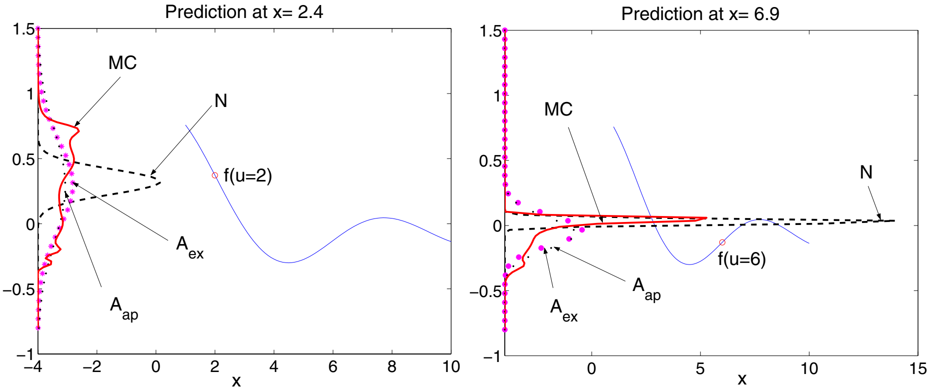

Fig. 7. Predictive distributions (on the y-axis) obtained when predicting at a noisy input: MC is the numerical approximation by simple Monte-Carlo, A_{ex} and A_{ap} correspond to the *Gaussian approximation* with moments computed exactly and approximately. N is the *naive* predictive distribution that does not account for the noise on the input.

For both the prediction at $x = 2.4$ (left) and $x = 6.9$ (right), Figure 8 shows the histogram of the losses (squared error E_1 on the left and minus log predictive density E_2 on the right) computed for each of the 100 samples given by the Monte-Carlo approximation. The minus-log predictive density loss is a very useful quantitative measure to assess the 'goodness' or quality of an approach as, unlike the squared error loss, it also accounts for the variance (or uncertainty) attached to the mean predictions. Table 2 summarizes the average losses obtained for each method (average over three test points). In this table, the losses reported for MC correspond to those obtained using the average sample mean and variance (average over 100 samples). We can also compute the losses associated to each sample and average those. We then obtain $E_1 = 0.42$ and $E_2 = 25.09$.

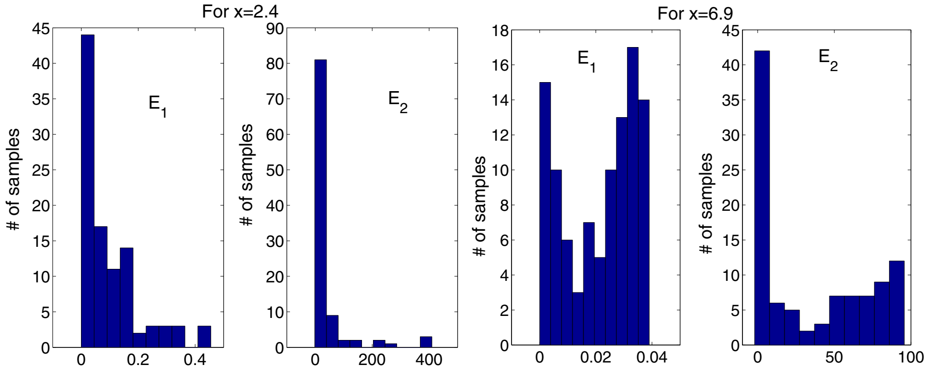

Fig. 8. Squared error (E_1) and minus log-likelihood (E_2) computed for 100 samples of the Monte-Carlo approximation (for the observed noisy $x = 2.4$, left and $x = 6.9$, right).

Table 2. Average squared error E_1 and minus log-predictive density E_2 over three test points obtained for the different approaches.

Loss	N	A_{ap}	A_{ex}	MC
E_1	0.009	0.004	0.005	0.004
E_2	7.685	−0.53	−0.635	−0.58

From this simple static example, for which the input noise variance is assumed to be known, we can conclude that our *Gaussian approximation* leads to results comparable to those obtained by simple Monte-Carlo, which approximates the true distribution.

6.2 Dynamic Case

The Mackey-Glass chaotic time-series constitutes a well-known challenging benchmark for the multiple-step ahead prediction task, due to its strong non-linearity [19]. We consider $\frac{dy(t)}{dt} = -by(t) + a\frac{y(t-\tau)}{1+y(t-\tau)^{10}}$, with $a = 0.2$, $b = 0.1$ and $\tau = 17$. The series is re-sampled with period 1 and normalized. We then assume the following NAR model $y_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-L})$, where $L = 16$ and we corrupt the output y_{t+1} by a white noise with variance 0.001. Having formed the input/output pairs, we train a zero-mean Gaussian Process with a Gaussian covariance function[14] on 100 points (taken at random). We first validate the model on one-step ahead predictions: We obtain $E_1 = 4.41\ 10^{-4}$, $E_2 = -2.16$ where the average is taken over $N_t = 1000$ test points. After performing a simulation of the test set (i.e. N_t-steps ahead prediction, where N_t is the length of the test set), we decide to make $k = 100$ steps ahead predictions (which corresponds to the horizon up to which predictions are 'reasonably good').

This example is intended to illustrate the propagation of uncertainty algorithm, described in section 5.2. We assess the quality of the predictions obtained using the approximate moments, given by the *Gaussian approximation*, by comparing them to the exact ones. We also compare the 'exact predictions' to those given by the naive approach, that feeds back only the predictive means as we predict ahead in time. Let t be the time up to which the time-series is known. Fig. 9 (top plots) shows the mean predictions (left) with their associated uncertainties (right) from $t + 1$ to $t + 100$. The crosses indicate the exact moments given by the *Gaussian approximation* (A_{ex}), the circles indicate the approximate moments (A_{ap}) and the dots the naive moments (N) that ignore the uncertainty induced by each successive prediction. We can see that up to around 60 steps ahead, the predictive means given by the different approaches are very similar. The uncertainty bars given by naive approach are very tight and the model is overly confident about its mean predictions. On the other hand, both the exact and approximate error-bars reflect well the fact that, as we predict ahead in time, less information is available and the estimates (predictive means) become more and more uncertain. On Fig. 9, the bottom left figure shows the 100-step ahead predictive means with their uncertainty. The upper plot shows the predictive means given by the naive approach, with their 2σ error-bars which are so tight that one cannot distinguish them from the means. The middle and bottom plots show respectively the approximate and exact 100-step ahead means where the shaded area corresponds to the uncertainty interval. On the right, we can see the evolution of the average squared error (left) and minus log-predictive density (right, on a log-scale), as the number of steps increases from one to 100. In this case, both losses clearly indicate that as the number of steps increases, the naive approach leads to poor predictions. These plots also show that, although not as good as A_{ex}, the predictions given by the approximate moments A_{ap} are quite encouraging.

[14] The covariance function is that given by Eq. (22) with $v = 1$.

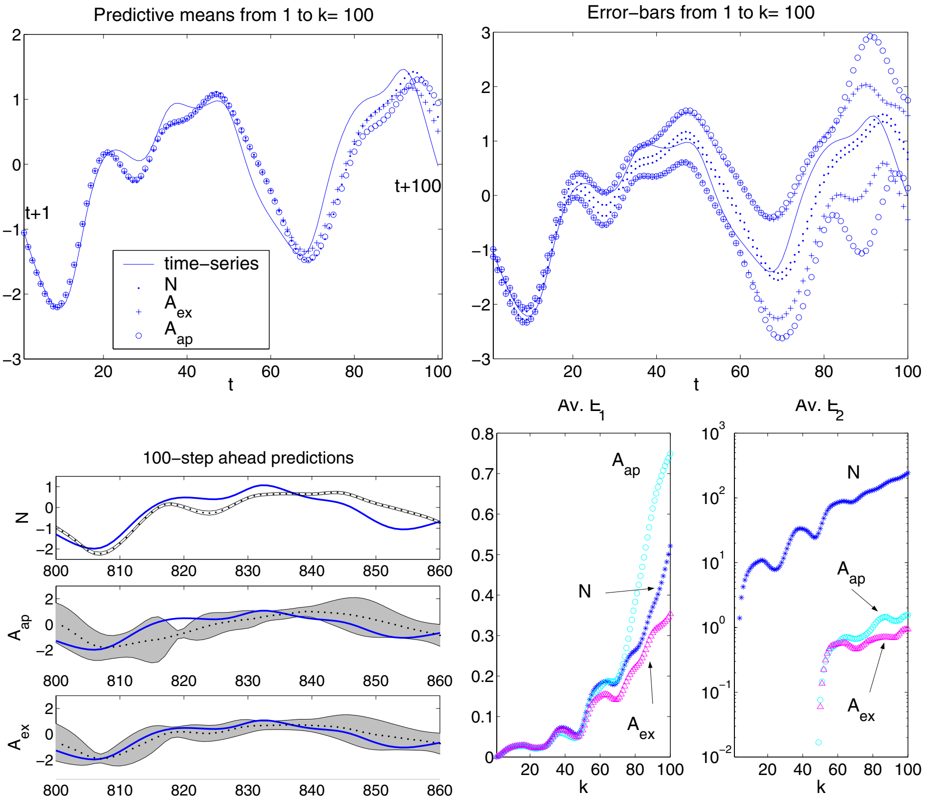

Fig. 9. Top plots: Iterative method in action on the Mackey-Glass time-series. Mean predictions (left) and uncertainty error bars (right) from 1 to 100 steps ahead, given by the exact moments A_{ap} (crosses), the approximate ones A_{ap} (circles) and the naive ones (dots). Bottom plots: 100-step ahead prediction of a portion of the time-series (left). From top to bottom: naive, approximate and exact means with the uncertainty region shaded. Right: Evolution of the average losses as the number of steps ahead increases from one to 100 (E_1 is the average squared error and E_2 the minus log-predictive density, on a log-scale)

We now turn to comparing the *Gaussian approximation* (exact moments) to the approximation of the true distribution by Monte-Carlo (MC). The Monte-Carlo approximation for the 100-step ahead prediction is done as follows: At $t+1$, compute $p(y_{t+1}|\mathcal{D}, \mathbf{x}_{t+1}) = \mathcal{N}(\mu(\mathbf{x}_{t+1}), \sigma^2(\mathbf{x}_{t+1}))$ where $\mathbf{x}_{t+1} = [y_t, y_{t-1}, \ldots, y_{t-16}]$. At $t+2$, draw a sample y^s_{t+1} from $p(y_{t+1}|\mathcal{D}, \mathbf{x}_{t+1})$, form the state $\mathbf{x}_{t+2} = [y^s_{t+1}, y_t, \ldots, y_{t-15}]$ and compute $p(y_{t+2}|\mathcal{D}, \mathbf{x}_{t+2}) = \mathcal{N}(\mu(\mathbf{x}_{t+2}), \sigma^2(\mathbf{x}_{t+2}))$. So on, up to $t+100$. Then, go back to $t+1$ and repeat the whole process. We repeat this $S = 1000$ times ($s = 1 \ldots S$), so that we finally obtain 1000 samples for each time-step. Finally, we do so for 100 different 'starting times t' (i.e. 100 test inputs), resulting in a $100 \times S \times k$ matrix of predictive means and variances, where S is the number of samples and k is the prediction horizon ($k = 100$). Fig. 10 shows the predictive uncertainties from $t + 1$ to $t + 100$.

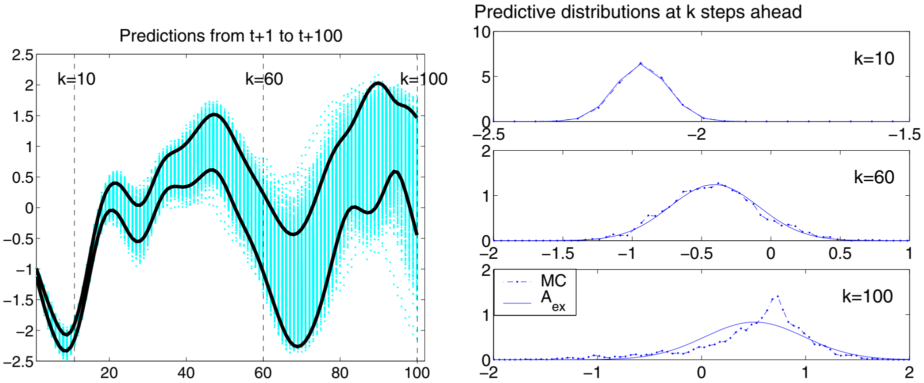

Fig. 10. Left: 1000 predictive error-bars from the Monte-Carlo approximation, from $t + 1$ to $t + k$, where $k = 100$ steps ahead. Also plotted, the predictive uncertainties given by the exact method A_{ex} (continuous lines). At $t + k$, for $k = 10, 60, 100$, we plot the corresponding predictive distribution(right plot), as it is approximated numerically by Monte-Carlo (dotted line) and analytically, by the Gaussian with exact moments (continuous).

This experiment clearly validates our analytical approximation of the true predictive distribution as we can see that the error bars given by the exact moments encompass those of the samples from the Monte-Carlo approximation. It is interesting noting how the approximation to the true distribution is long-tailed at 100 step-ahead.

Table 3, reports the average losses computed for the different approaches. (Note that since the Monte-Carlo approach uses only 100 test points, all losses are averaged over 100 points only.) The reported losses for MC correspond to the those computed using the average sample mean and variance. We can also compute the losses given by each single prediction and average them. We then obtain $E_1 = 0.72$ and $E_2 = 340.27$. These results for the Monte-Carlo approximation might look surprising but one should keep in mind that estimating the quality of this approximation with these losses is not really representative (since the distribution is not normal).

7 Conclusions

We have presented an original solution to the problem of iterative multiple-step ahead prediction of nonlinear dynamic systems within a NAR representation.

Table 3. Average (over 100 test points) squared error (E_1) and minus log predictive density (E_2) for the $k = 100$ step ahead predictions.

	N	A_{ap}	A_{ex}	MC
E_1	0.52	0.75	0.35	0.38
E_2	243.46	1.55	0.94	172.51

We do so by first showing how predicting at an uncertain or noisy input can be done within an analytical approximation of the predictive distribution of the Gaussian Process model (note that this approach is valid for other kernel-based models like the Relevance Vector Machines, see [20]). In experiments on simulated dynamic systems, we show that this analytical approach 1, performs as well as a numerical Monte-Carlo approximation of the true distribution and 2, propagating the uncertainty as we predict ahead in time improves the multiple-step ahead prediction task, achieving more realistic prediction variances than a method that uses only output estimates and thus ignores the uncertainty on current state.

In the derivation of the mean and variance of the predictive distribution, we show how exact or approximate moments are obtained, depending on the form of the covariance function. In the case of the Gaussian covariance function, for which exact moments are available, a numerical example proves that the approximate moments, computed using the Gaussian covariance function, lead to almost similar results as those obtained using the exact moments, which is encouraging for using the approximation.

Explicitly using the predictive variance has been recently successfully used in a control context [21] and also the propagation of uncertainty methodology, in a model predictive control framework where knowledge of the accuracy of the model predictions over the whole prediction horizon is required (see [22]).

In this chapter, we do not address the problem of learning in the presence of noisy inputs (we have assumed that the training inputs were noise-free). This is the subject of ongoing research. We suggest an approximation similar to that presented here: Assuming the input noise is white, the new non-Gaussian process can be approximated by a GP. We then derive its covariance function that accounts for the input noise variance, which is then learnt as an extra parameter.

Acknowledgements Thanks to Carl Edward Rasmussen who initiated this work. Many thanks to Joaquin Quiñonero-Candela for his feedback on this chapter and to Professor Mike Titterington for useful discussions on the subject. The authors gratefully acknowledge the support of the *Multi-Agent Control* Research Training Network - EC TMR grant HPRN-CT-1999-00107 and RMS is grateful for EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01.

References

[1] Williams, C.K.I., Rasmussen, C.E.: Gaussian Processes for Regression. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8., MIT Press (1996) 514–520

[2] Williams, C.K.I.: Prediction with Gaussian Processes: From linear regression to linear prediction and beyond. Technical Report NCRG-97-012, Dept of Computer Science and Applied Mathematics. Aston University. (1997)

[3] Williams, C.K.I.: Gaussian Processes. The handbook of Brain Theory and Neural Networks, Second edition, MIT Press (2002)

[4] Mackay, D.J.C.: Information theory, Inference and Learning Algorithms. Cambridge University Press (2003)

[5] Girard, A., Rasmussen, C., Quinonero-Candela, J., Murray-Smith, R.: Gaussian Process Priors With Uncertain Inputs – Application to Multiple-Step Ahead Time Series Forecasting. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems. Volume 15., MIT Press (2003) 545–552

[6] Quinonero-Candela, J., Girard, A., Larsen, J., Rasmussen, C.E.: Propagation of Uncertainty in Bayesian Kernels Models – Application to Multiple-Step Ahead Forecasting. In: IEEE International Conference on Acoustics, Speech and Signal Processing. Volume 2. (2003) 701–4

[7] Girard, A., Rasmussen, C., Murray-Smith, R.: Gaussian Process Priors with Uncertain Inputs: Multiple-Step Ahead Prediction. Technical Report TR-2002-119, Computing Science Department, University of Glasgow (2002)

[8] Lindley, D.V.: Introduction to Probability and Statistics from a Bayesian viewpoint. Cambridge University Press (1969)

[9] Papoulis, A.: Probability, random variables, and stochastic processes. McGraw-Hill (1991)

[10] Rasmussen, C.E.: Evaluation of Gaussian Processes and other methods for non-linear regresion. PhD thesis, University of Toronto (1996)

[11] Neal, R.M.: Bayesian learning for neural networks. PhD thesis, University of Toronto (1995)

[12] MacKay, D.J.C.: Bayesian methods for backpropagation networks. In Domany, E., van Hemmen, J.L., Schulten, K., eds.: Models of Neural Networks III. Springer-Verlag, New York (1994) 211–254

[13] Takens, F.: Detecting strange attractors in turbulence. In Rand, D., Young, L., eds.: Dynamical Systems and Turbulence. Volume 898., Springer-Verlag (1981) 366–381

[14] Murray-Smith, R., Girard, A.: Gaussian Process priors with ARMA noise models. In: Irish Signals and Systems Conference, Maynooth. (2001) 147–152

[15] Tresp, V., Hofmann, R.: Missing and Noisy Data in Nonlinear Time-Series Prediction. In S. F. Girosi, J. Mahoul, E.M., Wilson, E., eds.: Neural Networks for Signal Processing. Volume 24 of IEEE Signal Processing Society, New York. (1995) 1–10

[16] Tresp, V., Hofmann, R.: Nonlinear Time-Series Prediction with Missing and Noisy Data. Neural Computation **10** (1998) 731–747

[17] Ahmad, S., Tresp, V.: Some Solutions to the Missing Feature Problem in Vision. In Hanson, S.J., Cowan, J.D., Giles, C.L., eds.: Advances in Neural Information Processing Systems. Volume 5., Morgan Kaufmann, San Mateo, CA (1993) 393–400

[18] Judd, K., Small, M.: Towards long-term prediction. Physica D **136** (2000) 31–44

[19] Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science **197** (1977) 287–289

[20] Quinonero-Candela, J., Girard, A.: Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines – Application to Multiple-Step Ahead Time-Series Forecasting. Technical report, Informatics and Mathematical Modelling, Technical Univesity of Denmark (2002)

[21] Murray-Smith, R., Sbarbaro, D.: Nonlinear adaptive control using non-parametric Gaussian process prior models. In: 15th IFAC Triennial World Congress. International Federation of Automatic Control. (2002)

[22] Murray-Smith, R., Sbarbaro, D., Rasmussen, C.E., Girard, A.: Adaptive, Cautious, Predictive control with Gaussian Process priors. In: IFAC International Symposium on System Identification, Rotterdam. (2003)