Multiple-step Time Series Forecasting with Sparse Gaussian Processes

Perry Groot [a][b] Peter Lucas [a] Paul van den Bosch [b]

[a] *Radboud University, Model-Based Systems Development,*
Heyendaalseweg 135, 6525 AJ Nijmegen
[b] *Technical University Eindhoven, Department of Electrical Engineering,*
Control Systems, Potentiaal 4.28, 5600 MB Eindhoven

Abstract

Forecasting of non-linear time series is a relevant problem in control. Furthermore, an estimate of the uncertainty of the prediction is useful for constructing robust controllers. Multiple-step ahead forecasting has recently been addressed using Gaussian processes, but direct implementations are restricted to small data sets. In this paper we consider multiple-step forecasting for sparse Gaussian processes to alleviate this problem. We derive analytical expressions for multiple-step ahead prediction using the FITC approximation. On several benchmarks we compare the FITC approximation with a Gaussian process trained on a large portion of randomly drawn training samples. As a consequence of being able to handle larger data sets, we show a mean prediction that is closer to the true system response with less uncertainty.

1 Introduction

In this paper we consider non-linear time series of the form

$$x_{t+1} = f(x_t, u_t), \qquad y_t = x_t + \epsilon_t \tag{1}$$

with t a time index, x the system state, u a controllable input, y the observed state, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ a noise term, and f some non-linear transition function. In particular we consider the problem of forecasting future states, which is a relevant problem in control. Forecasting allows one to choose controls u in order to optimize a predefined economic criteria, for example, the throughput and amount of power used by a printer while print quality remains high. In this paper we focus on forecasting using a Gaussian process (GP), a flexible Bayesian framework that places a prior distribution over functions. The GP provides both a mean and a variance for each prediction, which is useful for constructing robust controllers [7, 12].

Multiple-step forecasting can be done using direct forecasting or using iterative one-step predictions with the latter usually being superior [2]. A naive way of repeatedly doing one-step ahead predictions is to feed back the mean of the prediction into the system state, but this has been shown to severely underestimate the uncertainty of the model predictions. A better approach is to also take the uncertainty of the input of the ith step into account to compute the $(i+1)$th step predictive distribution. The details using a Gaussian process (GP) approach have been worked out in [3, 4, 8, 9]. Initially by using Taylor approximations [3], but later extended to exact expressions for the first and second moments of the predictive distribution [8, 9], which can be solved analytically for a number of covariance functions [9, 5, 1]. Although for non-linear dynamical systems, a Gaussian input distribution does not guarantee the predictive distribution to be Gaussian, a GP allows an analytic Gaussian approximation that uses exact moment matching which has been shown to correspond well to the gold standard of MCMC sampling.

A direct implementation of GPs, however, limits their applicability to small data sets since the kernel matrix needs to be stored, costing $\mathcal{O}(N^2)$, and inverted, costing $\mathcal{O}(N^3)$, with N the number of data points. In recent years several approaches have addressed this problem often selecting a subset of the training data (the active set) of size M reducing the computational complexity to $\mathcal{O}(M^2N)$ for $M \ll N$. In this paper we focus on the FITC approximation [13], one of the leading methods for modeling sparse GPs, which relaxes the constraint that the active set has to come from the training data. This allows the M pseudo-inputs and addtional hyperparameters to be optimized at the same time.

The contribution of this paper are analytical expressions for multiple-step ahead prediction using the FITC approximation. We obtain a method with a predictive model that executes several magnitudes faster than the standard GP. On several benchmarks we compare the FITC approximation with a GP trained on a large portion of randomly drawn training samples. As a consequence of being able to handle larger data sets, we also show a mean prediction that is closer to the true system response with less uncertainty. The rest of this paper is structured as follows. Section 2 describes background information on Gaussian process regression and the FITC approximation for sparse GPs. Section 3 describes iterative one-step ahead time series forecasting using uncertainty propagation. Section 4 derives the analytical equations for time series forecasting with the FITC approximation. Section 5 presents two illustrative examples. Section 6 summarizes our conclusions.

2 Sparse Gaussian process regression

We denote vectors $\boldsymbol{x}$ and matrices $\boldsymbol{K}$ with bold-face type and their components with regular type, i.e., x_i, K_{ij} . With $\boldsymbol{x}^T$ we denote the transpose of the vector $\boldsymbol{x}$. $\mathcal{N}(\boldsymbol{x}|\boldsymbol{m}, \boldsymbol{V})$ is a Gaussian distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{V}$.

2.1 Gaussian processes

In this section we briefly describe the Gaussian process model for regression [11]. Let $\mathcal{D}$ be a data set consisting of N observed D-dimensional input vectors $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ and corresponding real-valued outputs $\boldsymbol{y} = \{y_n\}_{n=1}^N$. We assume that function observations follow from a latent function f that are corrupted by additive iid zero mean Gaussian noise, i.e., $y = f(\boldsymbol{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We place a GP prior on the latent function f giving a multi-variate Gaussian distribution on any finite subset of latent variables, i.e., the values of the function $f(\boldsymbol{x})$ at location $\boldsymbol{x}$. The GP is completely specified by a mean function (typically taken to be zero) and covariance function. In particular we have $p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}_{NN})$ where the covariance matrix $\boldsymbol{K}_{NN}$ is constructed from the covariance function $[\boldsymbol{K}_{NN}]_{nn'} = K(x_n, x_{n'})$. Typically the covariance function depends on additional hyperparameters. In this paper we will make use of the squared exponential (SE) covariance function with automatic relevance determination (ARD) hyperparameters

$$K(x_n, x_{n'}) = v \exp\left(-\frac{1}{2}(x_n - x_{n'})^T \boldsymbol{W}^{-1}(x_n - x_{n'})\right) \tag{2}$$

where $\boldsymbol{W} = \text{diag}(w_1^2, \ldots, w_D^2)$ allows for different length scales along each input dimension and v specifies the signal variance. By integrating out the latent function values we obtain the marginal likelihood

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{K}_{NN} + \sigma^2\boldsymbol{I}) \tag{3}$$

with $\boldsymbol{\Theta} = \{v, w_1, \ldots, w_D, \sigma^2\}$ all hyperparameters which we usually omit for readability. Maximum likelihood estimates for $\boldsymbol{\Theta}$ are typically obtained by minimizing the negative log marginal likelihood (e.g., using gradient descent), which can be evaluated exactly in the case of GP regression and is given by [11]

$$-\log p(\boldsymbol{y}|\boldsymbol{X}) = \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K}_{NN} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y} + \frac{1}{2}\log|\boldsymbol{K}_{NN} + \sigma^2\boldsymbol{I}| + \frac{N}{2}\log(2\pi). \tag{4}$$

The predictive distribution is obtained by considering a new point $\boldsymbol{x}_*$ and conditioning on the hyperparameters $\boldsymbol{\Theta}$ and data $\mathcal{D}$, which gives $p(y|\boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\Theta}) = \mathcal{N}(y|\mu_*, \sigma_*^2)$ with

$$\mu_* = \boldsymbol{K}_{*N}(\boldsymbol{K}_{NN} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}, \qquad \sigma_*^2 = K_{**} - \boldsymbol{K}_{*N}(\boldsymbol{K}_{NN} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_{N*} + \sigma^2 \tag{5}$$

Gaussian processes do not scale well for large data sets since training requires $\mathcal{O}(N^3)$ time because of the inversion of the covariance matrix. Once the inversion is done, computing the predictive mean is $\mathcal{O}(N)$ and computing the predictive variance is $\mathcal{O}(N^2)$ per new test case.

2.2 Sparse Gaussian processes

In this section we briefly describe the sparse pseudo-input Gaussian process [13], which was later renamed to Fully Independent Training Conditional (FITC) model to fit in the systematic framework of [10]. The FITC model approximates the full GP using M *pseudo*-inputs $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_m\}_{m=1}^M$ that are not restricted to the data inputs, but are rather hyperparameters that can be learned. Given a base covariance function K, the FITC covariance function has the following form:

$$K^{FITC}(\boldsymbol{x}_n, \boldsymbol{x}_{n'}) = \boldsymbol{K}_{nM}\boldsymbol{K}_{MM}^{-1}\boldsymbol{K}_{Mn'} + \lambda_n\delta_{nn'}, \qquad \lambda_n = K_{nn} - \boldsymbol{K}_{nM}\boldsymbol{K}_{MM}^{-1}\boldsymbol{K}_{Mn} \tag{6}$$

where $\boldsymbol{K}_{nM}$ is the matrix with elements $K(\boldsymbol{x}_n, \bar{\boldsymbol{x}}_m)$ of covariances between data inputs and pseudo-inputs and $\boldsymbol{K}_{MM}$ is the matrix with elements $K(\bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m'})$ of covariances between pseudo-inputs. The marginal likelihood is similar to Eq. (3)

$$p(\boldsymbol{y}|\boldsymbol{X}, \bar{\boldsymbol{X}}, \boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{K}_{NN}^{FITC} + \sigma^2\boldsymbol{I}) \tag{7}$$

which is used analogous to Eq. (4) to learn hyperparameters and pseudo-inputs of the FITC model jointly by minimizing the negative log likelihood using gradient descent. The predictive distribution is computed just as the standard GP model by considering a new point $\boldsymbol{x}_*$ and conditioning on the data $\mathcal{D}$ giving $p(y|\boldsymbol{x}_*, \mathcal{D}, \bar{\boldsymbol{X}}, \boldsymbol{\Theta}) = \mathcal{N}(y|\mu_*, \sigma_*^2)$ with

$$\mu_* = \boldsymbol{K}_{*M}\boldsymbol{Q}^{-1}\boldsymbol{K}_{MN}(\boldsymbol{\Lambda} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}, \qquad \sigma_*^2 = K_{**} - \boldsymbol{K}_{*M}(\boldsymbol{K}_{MM}^{-1} - \boldsymbol{Q}^{-1})\boldsymbol{K}_{M*} + \sigma^2 \tag{8}$$

where $\boldsymbol{Q} = \boldsymbol{K}_{MM} + \boldsymbol{K}_{MN}(\boldsymbol{\Lambda} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_{NM}$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$.

Training the FITC model is computationally more efficient since the covariance matrix $\boldsymbol{K}_{NN}^{FITC}$ (cf. Eq. (6)) consists of a sum of a low rank part and a diagonal part and can therefore be inverted in $\mathcal{O}(M^2N)$ rather than $\mathcal{O}(N^3)$. After some precomputations, computing the predictive mean is $\mathcal{O}(M)$ and computing the predictive variance is $\mathcal{O}(M^2)$ per new test case.

3 Iterative time-series forecasting with uncertainty propagation

In this section we briefly review iterative one-step ahead prediction with uncertainty propagation [3, 4, 8, 9]. We assume an autoregressive Gaussian process model with L lagged outputs given by

$$\boldsymbol{x}_k = [y_{k-L}, \dots, y_{k-1}]^T, \qquad y_k = f(x_k) + \epsilon \tag{9}$$

with f some non-linear function and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ a white noise term. Assuming the time series is known up to time T, denoted by Y_T, predicting the system k-steps ahead at time $T+k$ can be achieved by repeatedly doing one-step ahead predictions. When the input is a random variable $\boldsymbol{x}_*$ with an input distribution given by $p(\boldsymbol{x}_*|\boldsymbol{u}, \boldsymbol{S}) \sim \mathcal{N}(\boldsymbol{u}, \boldsymbol{S})$, the predictive distribution of the function value $p(f_*)$ is obtained by integrating over the input distribution:

$$p(f_*|\boldsymbol{u}, \boldsymbol{S}, \mathcal{D}) = \int p(f_*|\boldsymbol{x}_*, \mathcal{D}) p(\boldsymbol{x}_*|\boldsymbol{u}, \boldsymbol{S}) \, \mathrm{d}\boldsymbol{x}_* \tag{10}$$

The idea of iterative one-step ahead prediction with uncertainty propagation is to approximate at each step t the predictive distribution with a Gaussian distribution $\mathcal{N}(m(\boldsymbol{u}_t, \boldsymbol{S}_t), v(\boldsymbol{u}_t, \boldsymbol{S}_t))$. It has been shown that the first and second moment of the predictive distribution can be computed analytically for the SE kernel (cf. Eq. (2)) giving a prediction $p(y_t|Y_T) \sim \mathcal{N}(m(\boldsymbol{u}_t, \boldsymbol{S}_t), v(\boldsymbol{u}_t, \boldsymbol{S}_t) + \sigma^2)$, which is then fed back into the system state and represented by $\mathcal{N}(m(\boldsymbol{u}_{t+1}, \boldsymbol{S}_{t+1}), v(\boldsymbol{u}_{t+1}, \boldsymbol{S}_{t+1}))$. In particular, at time $T+1$ we have

$$\boldsymbol{u}_{T+1} = \begin{bmatrix} y_{T+1-L} \\ \vdots \\ y_T \end{bmatrix} \quad \text{and} \quad \boldsymbol{S}_{T+1} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix} \tag{11}$$

Since $p(y_{T+1}|Y_T) \sim \mathcal{N}(m(\boldsymbol{u}_{T+1}, \boldsymbol{S}_{T+1}), v(\boldsymbol{u}_{T+1}, \boldsymbol{S}_{T+1}) + \sigma^2)$ we have at time $T+2$:

$$\boldsymbol{u}_{T+2} = \begin{bmatrix} y_{T+2-L} \\ \vdots \\ m(\boldsymbol{u}_{T+1}, \boldsymbol{S}_{T+1}) \end{bmatrix} \quad \text{and} \quad \boldsymbol{S}_{T+2} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & v(\boldsymbol{u}_{T+1}, \boldsymbol{S}_{T+1}) + \sigma^2 \end{bmatrix} \tag{12}$$

This is repeated iteratively until at time $T + k$ we have

$$\begin{aligned}
\boldsymbol{u}_{T+k} &= \begin{bmatrix} m(\boldsymbol{u}_{T+k-L}, \boldsymbol{S}_{T+k-L}) \\ \vdots \\ m(\boldsymbol{u}_{T+k-1}, \boldsymbol{S}_{T+k-1}) \end{bmatrix} \quad \text{and} \\
\boldsymbol{S}_{T+k} &= \begin{bmatrix} v(\boldsymbol{u}_{T+k-L}, \boldsymbol{S}_{T+k-L}) + \sigma^2 & \dots & \operatorname{cov}(y_{T+k-L}, y_{T+k-1}) \\ \vdots & & \vdots \\ \operatorname{cov}(y_{T+k-1}, y_{T+k-L}) & \dots & v(\boldsymbol{u}_{T+k-1}, \boldsymbol{S}_{T+k-1}) + \sigma^2 \end{bmatrix}
\end{aligned} \tag{13}$$

4 Time series forecasting with sparse GPs

In this section we will derive the mathematical equations necessary for multiple-step ahead prediction with the FITC covariance function described in Section 2.2. In order to use the multiple-step ahead prediction approach described in Section 3, we need expressions for the mean $m(\boldsymbol{u}_t, \boldsymbol{S}_t)$, variance $v(\boldsymbol{u}_t, \boldsymbol{S}_t)$ and covariance $\operatorname{cov}(y_t, y_{t'})$. Let $\boldsymbol{B}^{(1)} = \boldsymbol{Q}^{-1}\boldsymbol{K}_{MN}(\boldsymbol{\Lambda} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$ and $\boldsymbol{B}^{(2)} = \boldsymbol{K}_{MM}^{-1} - \boldsymbol{Q}^{-1}$. The predictive equations using the FITC Gaussian process model are then given by $\mu(\boldsymbol{x}_*) = \boldsymbol{K}_{*M}\boldsymbol{B}^{(1)}$ and $\sigma_*^2 = K_{**} - \boldsymbol{K}_{*M}\boldsymbol{B}^{(2)}\boldsymbol{K}_{M*} + \sigma^2$. Furthermore, we define the following quantities for a kernel K:

$$\begin{aligned}
L_i^{(1)} &= \textstyle\int K(\boldsymbol{x}_*, \boldsymbol{x}_i) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_* & L_{ij}^{(2)} &= \textstyle\int K(\boldsymbol{x}_*, \boldsymbol{x}_i) K(\boldsymbol{x}_*, \boldsymbol{x}_j) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_* \\
L_i^{(3)} &= \textstyle\int \boldsymbol{x}_* K(\boldsymbol{x}_*, \boldsymbol{x}_i) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_* & L^{(4)} &= \textstyle\int K(\boldsymbol{x}_*, \boldsymbol{x}_*) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_*
\end{aligned} \tag{14}$$

Thus $\boldsymbol{L}^{(1)}$ is a $N \times 1$ vector, $\boldsymbol{L}^{(2)}$ a $N \times N$ matrix, $\boldsymbol{L}^{(3)}$ a $N \times D$ matrix, and $\boldsymbol{L}^{(4)}$ a scalar with N the number of training data points and D the dimension of the data. In [9] it has been shown that for the SE kernel (cf. Eq. (2)) the expressions $m(\boldsymbol{u}_t, \boldsymbol{S}_t)$, $v(\boldsymbol{u}_t, \boldsymbol{S}_t)$, and $\operatorname{cov}(y_t, y_{t'})$ can be stated in terms of $\boldsymbol{L}^{(1)}, \dots, \boldsymbol{L}^{(4)}$ which can be solved analytically:

$$\begin{aligned}
L_i^{(1)} &= v|\boldsymbol{W}^{-1}\boldsymbol{S} + I|^{-1/2} \exp\left(-\tfrac{1}{2}(\boldsymbol{u} - \boldsymbol{x}_i)^T(\boldsymbol{S} + \boldsymbol{W})^{-1}(\boldsymbol{u} - \boldsymbol{x}_i)\right) \\
L_{ij}^{(2)} &= v^2|2\boldsymbol{W}^{-1}\boldsymbol{S} + \boldsymbol{I}|^{-1/2} \\
&\quad \exp\left(-\tfrac{1}{2}\left[(\boldsymbol{u} - \boldsymbol{x}_d)^T(\tfrac{\boldsymbol{W}}{2} + \boldsymbol{S})^{-1}(\boldsymbol{u} - \boldsymbol{x}_d) + (\boldsymbol{x}_i - \boldsymbol{x}_j)^T(2\boldsymbol{W})^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]\right) \\
L_i^{(3)} &= L_i^{(1)} c_i \\
L^{(4)} &= v
\end{aligned} \tag{15}$$

where $\boldsymbol{x}_d = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_j)$ and $c_i = (\boldsymbol{W}^{-1} + \boldsymbol{S}^{-1})^{-1}(\boldsymbol{W}^{-1}\boldsymbol{x}_i + \boldsymbol{S}^{-1}\boldsymbol{u})$.

In the remainder we derive analytical expressions for the mean $m(\boldsymbol{u}_t, \boldsymbol{S}_t)$, variance $v(\boldsymbol{u}_t, \boldsymbol{S}_t)$ and covariance terms $\operatorname{cov}(y_t, y_{t'})$ for the FITC kernel function with respect to some base kernel K. We omit the time index t for brevity.

Computing the mean. Since the predictive mean is given by $\mu(\boldsymbol{x}_*) = \boldsymbol{K}_{*M}\boldsymbol{B}^{(1)}$ it holds that

$$m(\boldsymbol{u}, \boldsymbol{S}) = E_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)] = \int \mu(\boldsymbol{x}_*) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_* = \sum_m \boldsymbol{B}_m^{(1)} \int K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_m) p(\boldsymbol{x}_*)\, \mathrm{d}x_* \tag{16}$$

Thus $m(\boldsymbol{u}, \boldsymbol{S}) = \sum_m \boldsymbol{B}_m^{(1)} L_m^{(1)} = (\boldsymbol{B}^{(1)})^T \boldsymbol{L}^{(1)}$ where $\boldsymbol{L}^{(1)}$ is the required integral for the base kernel K with respect to the pseudo-inputs.

Computing the variance. Since $v(\boldsymbol{u}, \boldsymbol{S}) = E_{\boldsymbol{x}_*}[\sigma^2(\boldsymbol{x}_*)] + E_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)^2] - E^2_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)]$. These three terms give respectively

$$\begin{aligned}
E_{\boldsymbol{x}_*}[\sigma^2(\boldsymbol{x}_*)] &= \int \Big(K(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{K}_{*M} \boldsymbol{B}^{(2)} \boldsymbol{K}_{M*} \Big) p(\boldsymbol{x}_*) \, \mathrm{d}\boldsymbol{x}_* \\
&= L^{(4)} - \sum_i \sum_j \int K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_i) \boldsymbol{B}^{(2)}_{ij} K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_j) p(\boldsymbol{x}_*) \, \mathrm{d}\boldsymbol{x}_* = L^{(4)} - \sum_i \sum_j \boldsymbol{B}^{(2)}_{ij} L^{(2)}_{ij} \\
E_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)^2] &= \int (\sum_i \boldsymbol{B}^{(1)}_i K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_i))^2 p(\boldsymbol{x}_*) \, \mathrm{d}\boldsymbol{x}_* \\
&= \sum_i \sum_j \boldsymbol{B}^{(1)}_i \boldsymbol{B}^{(1)}_j \int K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_i) K(\boldsymbol{x}_*, \bar{\boldsymbol{x}}_j) p(\boldsymbol{x}_*) \, \mathrm{d}\boldsymbol{x}_* = \sum_i \sum_j \boldsymbol{B}^{(1)}_i \boldsymbol{B}^{(1)}_j L^{(2)}_{ij} \\
E^2_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)] &= (\sum_i \boldsymbol{B}^{(1)}_i L^{(1)}_i)^2
\end{aligned} \tag{17}$$

Thus we obtain the following expression for the variance

$$\begin{aligned}
v(\boldsymbol{u}, \boldsymbol{S}) &= L^{(4)} - \sum_i \sum_j (\boldsymbol{B}^{(2)}_{ij} - \boldsymbol{B}^{(1)}_i \boldsymbol{B}^{(1)}_j) L^{(2)}_{ij} - (\sum_i \boldsymbol{B}^{(1)}_i L^{(1)}_i)^2 \\
&= L^{(4)} - \mathrm{Tr}\Big((\boldsymbol{B}^{(2)} - \boldsymbol{B}^{(1)} (\boldsymbol{B}^{(1)})^T) \boldsymbol{L}^{(2)} \Big) - \mathrm{Tr}\Big(\boldsymbol{B}^{(1)} (\boldsymbol{B}^{(1)})^T \boldsymbol{L}^{(1)} (\boldsymbol{L}^{(1)})^T \Big)
\end{aligned} \tag{18}$$

Computing the covariances. For the covariance terms we compute $\mathrm{cov}(y_t, \boldsymbol{x}_t)$. The needed covariances can then be obtained by removing the oldest element of the state vector $\boldsymbol{x}_t$. Omitting the time index, it holds that $\mathrm{cov}(y, \boldsymbol{x}) = E[y\boldsymbol{x}] - E[y]E[\boldsymbol{x}]$ where $E[y] = m(\boldsymbol{u}, \boldsymbol{S})$ and $E[\boldsymbol{x}] = \boldsymbol{u}$. Since

$$\begin{aligned}
E[y\boldsymbol{x}] &= \int \boldsymbol{x} \mu(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\
&= \sum_m \boldsymbol{B}^{(1)}_m \int \boldsymbol{x} K(\boldsymbol{x}, \bar{\boldsymbol{x}}_m) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \sum_m \boldsymbol{B}^{(1)}_m L^{(3)}_m
\end{aligned} \tag{19}$$

we obtain the following expression for the covariance terms

$$\mathrm{cov}(y, \boldsymbol{x}) = \sum_m \boldsymbol{B}^{(1)}_m L^{(3)}_m - (\sum_m \boldsymbol{B}^{(1)}_m L^{(1)}_m) \boldsymbol{u} \tag{20}$$

In the next section we present experimental results for multiple-step ahead prediction for two benchmark problems leading to a better predictive distribution when using the FITC approximation compared to a standard GP. At the same time computing the predictive distribution is more efficient since the expressions $\boldsymbol{L}^{(1)}, \ldots, \boldsymbol{L}^{(4)}$ are computed for the base kernel K with respect to the pseudo-inputs $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_m\}_{m=1}^M$. Thus $\boldsymbol{L}^{(1)}$ is a $M \times 1$ vector, $\boldsymbol{L}^{(2)}$ a $M \times M$ matrix, $\boldsymbol{L}^{(3)}$ a $M \times D$ matrix, and $\boldsymbol{L}^{(4)}$ a scalar with M the number of pseudo-inputs and D the dimension of the data.

5 Experiments

In this section we compare the FITC model with a standard GP model on two benchmark problems. We report the mean together with their 2σ error bars. To evaluate the goodness of fit of the model we also report the root mean squared error (RMSE) and the absolute mean error (MAE). The RMSE is defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (y_i - \hat{y}_i)^2}, \tag{21}$$

where y_i is the target value, $\hat{y}_i$ predicted value, and N the number of test examples. Since the RMSE can be dominated by a few large residuals we also report the MAE which is defined as

$$MAE = \frac{1}{N} \sum_i^N |y_i - \hat{y}_i|, \tag{22}$$

in which the influence of individuals is linear.

Example 1. The first benchmark that we consider is a simple dynamical system with one system state and one control input given by

$$x_{k+1} = 0.95 * \tanh(x_k) + \sin(u_k), \qquad y_k = x_k + \epsilon_k \tag{23}$$

where at time k, x_k is the system state, u_k is the known control input, and $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ is a noise term.

We generated time series data of 1000 points used for training. Given values y_k, u_k for $k = 1, \dots, 1000$ we generated a training data set of the form

$$X = \begin{bmatrix} y_1 & u_1 \\ y_2 & u_2 \\ \vdots & \vdots \\ y_{999} & u_{999} \end{bmatrix} \qquad Y = \begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_{1000} \end{bmatrix} \tag{24}$$

We trained a GP model with a squared exponential kernel as in Eq. (2) on $N = 200$ points randomly drawn training data points. The sparse GP model used a squared exponential base kernel and was trained using all training data and $M = 20$ pseudo-inputs. Figure 1 shows the mean prediction and 2σ error bars of both models from $k = 1$ to $k = 100$ steps ahead on a separate test set. In the training data the control signal ranged between -2 and 2 whereas in the test set the control signal ranged between -3 and 3 explaining the higher amount of uncertainty whenever $|u_k| > 2$ occurs. The sparse GP model shows a mean prediction that is closer to the true model with less uncertainty. The RMSE and MAE are reported in Table 1 showing slightly better values for the FITC model. The FITC model is, however, much more efficient for multiple-step ahead prediction since $M \ll N$.

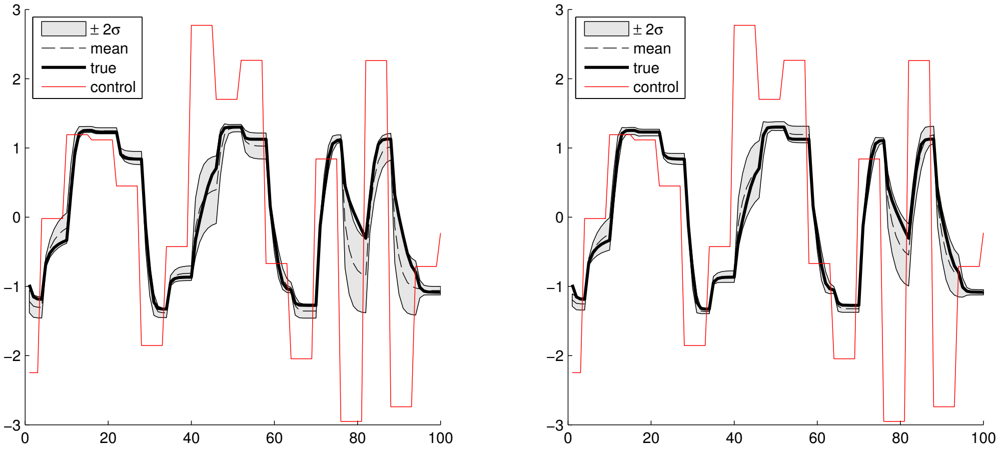

Figure 1: Dynamic system 100-step ahead prediction benchmark. Left: standard GP using 200 randomly drawn training inputs. Right: FITC with 20 pseudo-inputs.

Table 1: Comparison for two multiple-step ahead prediction benchmarks.

	Example 1		Example 2	
	GP	FITC	GP	FITC
RMSE	0.55	0.54	0.34	0.09
MAE	0.37	0.35	0.26	0.06

Example 2. A well-known benchmark for multiple-step ahead prediction is the Mackey-Glass chaotic time series, defined as $\frac{dz(t)}{dt} = -bz(t) + a\frac{z(t-\tau)}{1+z(t-\tau)^{10}}$, which shows strong non-linear behaviour [6]. In our experiments we used $a = 0.2$, $b = 0.1$, and $\tau = 17$ and the resulting time series was re-sampled with period 1 and normalized. For the models we used 16 lagged outputs in the state vector $\boldsymbol{x}_k = [y_{k-16}, \dots, y_{k-1}]$ where y_k is the observed system state $\boldsymbol{x}_k$ corrupted by white noise with variance 0.001.

We trained a GP model with a squared exponential kernel as in Eq. (2) on $N = 200$ points randomly drawn from a time series of 1200 points. The sparse GP model used a squared exponential base kernel and was trained using the 1200 points and $M = 40$ pseudo-inputs. Figure 2 shows the mean prediction and two standard deviations of both models from $k = 1$ to $k = 100$ steps ahead on a separate test set. The sparse GP model clearly shows a mean prediction that is closer to the true model with less uncertainty. At the same time the k-step ahead prediction is much more efficient since $M \ll N$. The RMSE and MAE are reported in Table 1, showing much beter values for the FITC model.

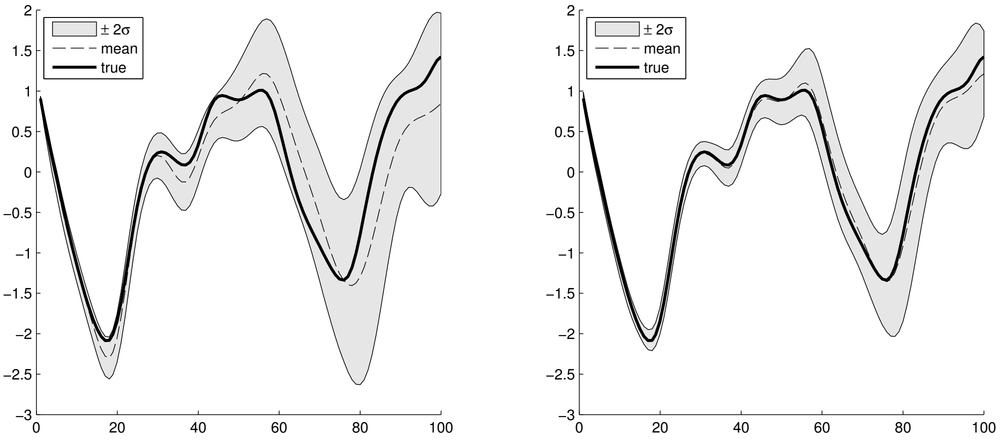

Figure 2: Mackey-Glass 100-step ahead prediction benchmark. Left: standard GP using 200 randomly drawn training inputs. Right: FITC with 40 pseudo-inputs.

6 Conclusions

In this paper we looked at the problem of non-linear time series forecasting in the context of iterative one-step predictions. A naive approach that only propagates the mean of the predictive distribution typically underestimates the uncertainty of the model predictions. This can be improved by also taking the uncertainty of the input into account. Details of such an approach has been worked out earlier using Gaussian processes. Naive implementations of Gaussian processes, however, limits their applicability to small data sets since inverting the kernel matrix costs $\mathcal{O}(N^3)$ with N the number of data points.

In this paper we have derived analytical equations for multiple-step ahead forecasting using sparse Gaussian processes. We reported results on two dynamical systems examples between a standard GP and a sparse GP. The sparse GP approach produced results several magnitudes faster than the standard GP approach. As a consequence of being able to handle larger data sets this also resulted in better predictive distributions.

Further research will focus on extending the presented framework for forecasting using sparse GPs with dimensionality reduction [14] and control optimization [12]. Other interesting directions are to combine the problem of predicting from noisy inputs with the problem of training a model from noisy inputs.

Acknowledgement

This work has been carried out as part of the OCTOPUS project with Océ Technologies B.V. under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Bsik program.

References

[1] K. Ažman and J. Kocijan. Non-linear model predictive control for models with local information and uncertainties. *Transactions of the Institute of Measurement and Control*, 30(5):371–396, 2008.

[2] J.D. Farmer and J.J. Sidorowich. Exploiting chaos to predict the future and reduce noise. Technical Report LA-UR-88, Los Alamos National Laboratory, 1985.

[3] A. Girard, C.E. Rasmussen, and R. Murray-Smith. Gaussian process priors with uncertain inputs: Multiple-step ahead prediction. Technical Report 119, Dept. of Computing Science, University of Glasgow, 2002.

[4] A. Girard, C.E. Rasmussen, J. Quiñonero-Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. In *Neural Information Processing Systems*, 2003.

[5] J. Kocijan, A. Girard, and D.J. Leith. Incorporating linear local models in Gaussian process model. Technical Report IJS report DP-8895, Jozef Stefan Institute, 2004.

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

[7] R. Murray-Smith and D. Sbarbaro. Nonlinear adaptive control using non-parametric Gaussian process prior models. In *15th IFAC World Congress on Automatic Control*, 2002.

[8] J. Quiñonero-Candela, A. Girard, J. Larsen, and C.E. Rasmussen. Propagation of uncertainty in Bayesian kernel models - application to multiple-step ahead forecasting. In *International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 701–704, Hong-Kong, 2003. IEEE.

[9] J. Quiñonero-Candela, A. Girard, and C.E. Rasmussen. Prediction at an uncertain input for Gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting. Technical report, IMM, Danish Technical University, 2002.

[10] J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[11] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, 2006.

[12] D. Sbarbaro and R. Murray-Smith. Self-tuning control of non-linear systems using gaussian process prior models. *Lecture Notes in Computer Science*, 3355:140–157, 2005.

[13] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*. MIT Press, 2006.

[14] E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the 22nd Annual Conference on Uncertainty in AI*. AUAI Press, 2006.