State Space approximation of Gaussian Processes for time series forecasting

Alessio Benavoli[1] and Giorgio Corani[2]

[1] School of Computer Science and Statistics (SCSS),
Trinity College Dublin, Ireland. alessio.benavoli@tcd.ie
[2] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
USI - SUPSI Lugano, Switzerland. giorgio.corani@idsia.ch

Abstract. Gaussian Processes (GPs), with a complex enough additive kernel, provide competitive results in time series forecasting compared to state-of-the-art approaches (arima, ETS) provided that: (i) during training the unnecessary components of the kernel are made irrelevant by automatic relevance determination; (ii) priors are assigned to each hyperparameter. However, GPs computational complexity grows cubically in time and quadratically in memory with the number of observations. The state space (SS) approximation of GPs allows to compute GPs based inferences with linear complexity. In this paper, we apply the SS representation to time series forecasting showing that SS models provide a performance comparable with that of full GP and better than state-of-the-art models (arima, ETS). Moreover, the SS representation allows us to derive new models by, for instance, combining ETS with kernels.

Keywords: time series forecasting · Gaussian Process · State Space approximation.

1 Introduction

Gaussian Processes (GPs) [15] are a powerful tool for modeling correlated observations, including time series. GPs have been used for the analysis of astronomical time series (see [4] and the references therein), forecasting of electric load [12] and analysis of correlated and irregularly-sampled time series [16].

A kernel composition specific for time series has been recently proposed [3]. It contains linear trend, periodic patterns, and other flexible kernel for modeling the non-linear trend. By setting priors on the hyperparameters, which keep the inference within a reasonable range even on short time series, the GP yields very accurate forecasts, outperforming the traditional time series models.

Note that the above GP based model is a type of Generalised Additive Model (GAM) [26]. However, contrarily to traditional GAMs, it uses different nonparametric components for the periodic and non-linear terms, and it is estimated in a fully Bayesian way (that is, without backfitting).

Yet, GPs have computational complexity $O(n^3)$ and storage demands of $O(n^2)$; hence, they are not suitable for large datasets. Several approximations

have been proposed to reduce their computational complexity to $O(n)$, such as sparse approximations based on inducing points [14, 20, 24, 6, 7, 1, 19], which however add additional hyperparameters.

In the case of time series, it is possible to represent the full GP as a State Space model, without the need for any additional hyperparameter [18, 22, 17, 11, 13, 2] and with $O(n)$ complexity.

We focus on the SS representation of the GP and we provide the following contributions. We discuss how to represent the model of [3] as a SS model, obtaining almost identical results on the time series of the M3 competition.

We also apply the GP model of [3] to very long time series, thanks to the SS representation. Also in this case we obtain positive results w.r.t the competitors.

Moreover, once the covariance functions of the Gaussian are represented in the SS framework, they can be combined with the existing SS models. This opens up the possibility of developing novel time series models. As a proof of concept, we consider a traditional state-space model (additive exponential smoothing) and we replace its seasonal component with the SS representation of the periodic kernel of the GP. We obtain a less parameterized model, which has higher accuracy on the time series of the M3 competition. The resulting model is also more flexible; for instance, it could be easily extended to manage time series containing multiple seasonal patterns, unlike the traditional exponential smoothing.

2 Background

In the following section, we provide a background on (i) Gaussian Processes; (ii) State Space models; (iii) the State Space representation of Gaussian Processes.

2.1 Gaussian Process

We consider the regression model

$$y = f(\mathbf{x}) + v, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^p$, $f : \mathbb{R}^p \to \mathbb{R}$ and $v \sim N(0, s_v^2)$ is the noise. Our goal is to estimate f given the training data $\mathcal{D} = \{(\mathbf{x}_i, y_i),\ i = 1, \ldots, n\}$. In GP regression, we place a GP prior on the unknown f, $f \sim GP(0, k_{\boldsymbol{\theta}})$,[3] and calculate the posterior distribution of f given the data $\mathcal{D}$. We then employ this posterior to make inferences about f.

In particular, we are interested in predictive inferences. Based on the training data $X^T = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $\mathbf{y} = [y_1, \ldots, y_n]^T$, and given m test inputs $(X^*)^T = [\mathbf{x}_1^*, \ldots, \mathbf{x}_m^*]$, we aim to find the posterior distribution of $\mathbf{f}^* = [f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*)]^T$. From (1) and the properties of the Gaussian distribution, [4]

[3] A GP prior with zero mean function and covariance function $k_{\boldsymbol{\theta}} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+$, which depends on a vector of hyperparameters $\boldsymbol{\theta}$

[4] In this work, we include the additive noise v into the kernel by adding a White noise kernel term.

the posterior distribution of $\mathbf{f}^*$ is Gaussian [15, Sec. 2.2]:

$$p(\mathbf{f}^*|X^*, X, \mathbf{y}, \boldsymbol{\theta}) = N(\mathbf{f}^*; \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(X^*|X, \mathbf{y}), \hat{K}_{\boldsymbol{\theta}}(X^*, X^*|X)), \tag{2}$$

with mean and covariance given by:

$$\begin{aligned}
&\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\mathbf{f}^*|X, \mathbf{y}) = K_{\boldsymbol{\theta}}(X^*, X)(K_{\boldsymbol{\theta}}(X, X))^{-1}\mathbf{y}, \\
&\hat{K}_{\boldsymbol{\theta}}(X^*, X^*|X) = K_{\boldsymbol{\theta}}(X^*, X^*) - K_{\boldsymbol{\theta}}(X^*, X)(K_{\boldsymbol{\theta}}(X, X))^{-1}K_{\boldsymbol{\theta}}(X, X^*).
\end{aligned} \tag{3}$$

In GPs, the kernel defines the Covariance Function (CF) between any two function values: $Cov(f(\mathbf{x}), f(\mathbf{x}^*)) = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}^*)$. Common kernels are the White Noise (WN), the Linear (LIN), the Matern 3/2 (MAT32), the Matern 5/2 (MAT52), the Squared Exponential (RBF), the Cosine (COS) and the Periodic (PER). Hereafter, we provide the expressions of these kernels for $p = 1$, which is the case of time series; see instead [15] for generalizations:

WN: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_v^2 \delta_{x_1, x_2}$
LIN: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_b^2 + s_l^2 x_1 x_2$
MAT32: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_e^2 \left(1 + \frac{\sqrt{3}|x_1 - x_2|}{\ell_e}\right) \exp\left(-\frac{\sqrt{3}|x_1 - x_2|}{\ell_e}\right)$
MAT52: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_e^2 \left(1 + \frac{\sqrt{5}|x_1 - x_2|}{\ell_e} + \frac{5(x_1 - x_2)^2}{3\ell_e^2}\right) \exp\left(-\frac{\sqrt{5}|x_1 - x_2|}{\ell_e}\right)$
RBF: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_r^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\ell_r^2}\right)$
COS: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_c^2 \cos\left(\frac{x_1 - x_2}{\tau}\right)$
PER: $k_{\boldsymbol{\theta}}(x_1, x_2) = s_p^2 \exp\left(-\frac{(2\sin^2(\pi|x_1 - x_2|/p_e)}{\ell_p^2}\right)$

where δ_{x_1, x_2} is the Kronecker delta, which equals one when $x_1 = x_2$ and zero otherwise. The hyperparameters are the variances $s_v^2, s_l^2, s_e^2, s_r^2, s_c^2, s_p^2 > 0$, the lengthscales $\ell_r, \ell_e, \ell_p, \tau > 0$ and the period p_e.

Selecting a kernel, or a combination of kernels, to determine the structure of the covariance is a crucial factor governing the performance of a GP model. Spectral mixture kernels (SM) [25] have been devised to overcome this issue thanks to their property of being able to approximate any stationary kernel.[5] SM define a covariance kernel by taking the inverse Fourier transform of a weighted sum of different shifts of a probability density. In the original formulation [25], the authors considered a Gaussian PDF, resulting into a covariance kernel which is the sum of the RBF×COS kernels, so each term in the sum is equal to:

$$\text{SM}_i\text{: } k_{\boldsymbol{\theta}}(x_1, x_2) = s_{m_i}^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\ell_{m_i}^2}\right) \cos\left(\frac{x_1 - x_2}{\tau_{m_i}}\right),$$

with hyperparameters s_{m_i}, ℓ_{m_i} and τ_{m_i}.

[5] A stationary kernel is one which is translation invariant: $k_{\boldsymbol{\theta}}(x_1, x_2)$ depends only on $x_1 - x_2$, like for instance the Matern and RBF kernels.

Learning the hyperparameters We denote by $\boldsymbol{\theta}$ the vector containing all the kernels' hyperparameters. In practical application of GPs, $\boldsymbol{\theta}$ have to be selected. We use Bayesian model selection to consistently set such parameters. Variances and lengthscales are non-negative hyperparameters, to which we assign log-normal priors (later we show how we define the priors). We then compute the maximum a-posteriori (MAP) estimate of $\boldsymbol{\theta}$, that is we maximize w.r.t. $\boldsymbol{\theta}$ the joint marginal probability $p(\mathbf{y}, \boldsymbol{\theta})$, which is the product of the prior $p(\boldsymbol{\theta})$ and the marginal likelihood [15, Ch.2]:

$$p(\mathbf{y}|X, \boldsymbol{\theta}) = N(\mathbf{y}; 0, K_{\boldsymbol{\theta}}(X, X)). \tag{4}$$

Usually $\boldsymbol{\theta}$ is selected by maximizing the marginal likelihood of Eq. (4). Yet, better estimates can be obtained by assigning prior to the hyperparameters and then performing MAP estimation. The MAP approach yields reliably estimates also on short time series, as pointed out by [3], in which it is also proposed a methodology to define such priors.

2.2 State Space models

Consider the following stochastic continuous time-variant (LTV) State Space (SS) model [10]

$$\begin{cases} d\mathbf{f}(t) = \mathbf{F}(t)\,\mathbf{f}(t)dt + \mathbf{L}(t)\,dw(t), \\ y(t_k) = \mathbf{C}(t_k)\,\mathbf{f}(t_k), \end{cases} \tag{5}$$

where $\mathbf{f}(t) = [f_1(t), \ldots, f_m(t)]^T$ is the state vector,[6] $y(t_k)$ is the measurement at time t_k, $\mathbf{F}(t), \mathbf{C}(t), \mathbf{L}(t)$ are known matrices of appropriate dimensions and $w(t)$ is a one-dimensional Wiener noise process with intensity $q(t)$. We further assume that the initial state $\mathbf{f}(t_0)$ and $w(t)$ are independent for each $t \geq t_0$. The solution of the stochastic differential equation in (5) is [10]:

$$\mathbf{f}(t_k) = \boldsymbol{\psi}(t_k, t_0)\,\mathbf{f}(t_0) + \int_{t_0}^{t_k} \boldsymbol{\psi}(t_k, \tau)\mathbf{L}(\tau)\,dw(\tau), \tag{6}$$

with $\boldsymbol{\psi}(t_k, t_0) = \exp(\int_{t_0}^{t_k} \mathbf{F}(t)dt)$ is the state transition matrix, which is computed as a matrix exponential.[7] Assuming that $E[\mathbf{f}(t_0)] = \mathbf{0}$, then it can be easily proven that the vector of observations $[y(t_1), y(t_2), \ldots, y(t_n)]^T$ is Gaussian distributed with zero mean and covariance matrix whose elements are given by:

$$\begin{aligned} E[y(t_i)y(t_j)] = {} & \mathbf{C}(t_i)\boldsymbol{\psi}(t_i, t_0)E[\mathbf{f}(t_0)\mathbf{f}^T(t_0)](\mathbf{C}(t_j)\boldsymbol{\psi}(t_j, t_0))^T \\ & + \int_{t_0}^{\min(t_i,t_j)} h(t_i, u)h(t_j, u)q(u)du \end{aligned} \tag{7}$$

[6] m is a latent dimension which defines the dimension of the state space. The state is a function of tim.

[7] The matrix exponential is $e^A = I + A + A^2/2! + A^3/3! + \ldots$ and, for many matrices A, it can be computed analytically.

where we have exploited the fact that $E[dw(u)dw(v)] = q(u)\delta(u-v)dudv$ [10] and defined $h(t_1, t_2) = \mathbf{C}(t_1)\boldsymbol{\psi}(t_1, t_2)\mathbf{L}(\tau)$.

In SS models, one aims to estimate the states $\mathbf{f}(t_1), \ldots, \mathbf{f}(t_n)$ given the observations $y(t_1), \ldots, y(t_n)$ and the initial condition. There are in particular two problems of interest: (i) *filtering* whose aim is to compute $p(\mathbf{f}(t_k)|y(t_1), \ldots, y(t_k))$ for every t_k; (ii) *smoothing* whose aim is to compute $p(\mathbf{f}(t_k)|y(t_1), \ldots, y(t_n))$ for every t_k. For stochastic LTV systems, filtering and smoothing can be solved exactly using the Kalman Filter (KF) and the Rauch-Tung-Striebel smoother [10] with complexity $\mathcal{O}(n)$.

2.3 SS models representation of GPs

When the GP has one-dimensional input, it is possible to represent (or approximate) the GP with a SS model. The advantage of the SS representation is that estimates and inferences can be computed with complexity $\mathcal{O}(n)$. In practice, one has to find a SS whose covariance matrix (7) coincides (or approximates) that of the GP. This provides the SS representation of the GP, which then allows us to estimate $\mathbf{f}(t_k)$ given data $\{y(t_1), \ldots, y(t_n)\}$ using the KF and the Rauch-Tung-Striebel smoother (with complexity $\mathcal{O}(n)$). This can be obtained as follows:

1. Discretize the continuous-time SS to obtain a discrete-time SS (this step basically consists on applying (6)):

$$\begin{cases} \mathbf{f}(t_k) = \boldsymbol{\psi}(t_k, t_{k-1})\mathbf{f}(t_{k-1}) + \boldsymbol{\nu}(t_{k-1}), \\ y(t_k) = \mathbf{C}(t_k)\,\mathbf{f}(t_k), \end{cases} \tag{8}$$

   where $\boldsymbol{\nu}(t_{k-1}) = \int_{t_{k-1}}^{t_k} \boldsymbol{\psi}(t_k, \tau)\mathbf{L}\, dw(\tau)$.
2. Compute the probability density function (PDF) $p(\mathbf{f}(t_k)|y(t_1), \ldots, y(t_k))$, which is Gaussian. The mean and covariance matrix of this Gaussian PDF can be computed efficiently by using the KF.
3. Compute the Gaussian posterior PDF $p(\mathbf{x}(t_k)|y(t_1), \ldots, y(t_n))$ – the mean and covariance matrix of this PDF can be computed very efficiently by using the Rauch-Tung-Striebel smoother. This step returns the estimates of the state given all observations.
4. To estimate the hyperparameters of the CF, we can perform MAP (as for GPs). Note that, the marginal likelihood of the SS model can be computed efficiently by the KF.

State Space representation of covariance functions The time continuous SS representation of the covariance functions of Section 2.1 is given in Table 1. Such representations do not include the variance scaling parameter that multiplies the CF; it can be however included in the SS model by rescaling either the stochastic forcing term or the initial condition (for SS without forcing term).

WN	$\begin{cases} \frac{df}{dt}(t) = \frac{dw}{dt}(t) \\ y(t_k) = f(t_k) \end{cases}$
LIN	$\begin{cases} \frac{df_1}{dt}(t) = f_2(t) \\ \frac{df_2}{dt}(t) = 0 \\ y(t_k) = f_1(t_k) \end{cases} \quad \begin{bmatrix} f_1(t_0) \\ f_2(t_0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} s_b^2 & 0 \\ 0 & s_l^2 \end{bmatrix} \right)$
MAT32	$\begin{cases} \frac{df_1}{dt}(t) = f_2(t) \\ \frac{df_2}{dt}(t) = -\frac{3}{\ell^2} f_1(t) - \frac{2\sqrt{3}}{\ell} f_2(t) + \frac{12\sqrt{3}}{\ell^3} \frac{dw}{dt}(t) \\ y(t_k) = f_1(t_k) \end{cases}$
MAT52	$\begin{cases} \frac{df_1}{dt}(t) = f_2(t) \\ \frac{df_2}{dt}(t) = f_3(t) \\ \frac{df_3}{dt}(t) = -\frac{3\sqrt{5}}{\ell} f_1(t) - \frac{15}{\ell^2} f_2(t) - \frac{3\sqrt{5}}{\ell} f_3(t) + \frac{400\sqrt{5}}{3\ell^5} \frac{dw}{dt}(t) \\ y(t_k) = f_1(t_k) \end{cases}$
COS	$\begin{cases} \frac{df_1}{dt}(t) = \frac{1}{\tau} f_2(t) \\ \frac{df_2}{dt}(t) = -\frac{1}{\tau} f_1(t) \\ y(t_k) = f_1(t_k) \end{cases} \quad \begin{bmatrix} f_1(t_0) \\ f_2(t_0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$

Table 1. SS representation of the CFs. When the distribution of the initial state is not provided, it is assumed to be equal to zero. The intensity of the Wiener process w is assumed to be $q = 1$.

Representing compositions of covariance functions Additive combination of covariance functions can be represented by stacking SS models; this is called *cascade composition* [17]. For instance, the SS model corresponding to WN+LIN is:

$$\begin{cases} \frac{df_1}{dt}(t) = \frac{dw}{dt}(t) \\ \frac{df_2}{dt}(t) = f_3(t) \\ \frac{df_3}{dt}(t) = 0 \\ y(t_k) = f_1(t_k) + f_2(t_k) \end{cases} \quad \begin{bmatrix} f_2(t_0) \\ f_3(t_0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} s_b^2 & 0 \\ 0 & s_l^2 \end{bmatrix} \right).$$

Multiplicative composition of covariance functions can be obtained via *parallel composition* [17] of SS models. For instance, the COS×MAT32 kernel is represented as:

$$\begin{cases} \frac{df_1}{dt}(t) = \omega f_2(t) + f_3(t) \\ \frac{df_2}{dt}(t) = -\omega f_1(t) + f_4(t) \\ \frac{df_3}{dt}(t) = -\frac{3}{\ell^2} f_1(t) - \frac{2\sqrt{3}}{\ell} f_3(t) + \omega f_4(t) + \frac{12\sqrt{3}}{\ell^3} \frac{dw_1}{dt}(t) \\ \frac{df_4}{dt}(t) = -\frac{3}{\ell^2} f_2(t) - \omega f_3(t) - \frac{2\sqrt{3}}{\ell} f_4(t) + \frac{12\sqrt{3}}{\ell^3} \frac{dw_2}{dt}(t) \\ y(t_k) = f_1(t_k) \end{cases}$$

The RBF and PER kernel do not admit an exact SS representation; for this reason, they are not shown in Table 1. However, an approximated SS representation can be given. The PER kernel can be approximated as the sum of different Cosine covariance functions (COS+COS+...+COS), with a suitable choice of their lengthscales (defined using a Fourier series expansion of the PER kernel) [21]. In this paper, we use 7 COS terms to approximate the PER kernel. The RBF kernel can be approximated by a SS model based on the Matern d/2 kernel, where $d = 1, 3, 5, 7, 9, \ldots$ and the approximation improves as d increases. In this paper, we will use $d = 3$.

2.4 Time series forecasting and priors

In [3], GP regression was proposed for time series forecasting using the following composite kernel:

$$\mathrm{K} = \mathrm{PER} + \mathrm{LIN} + \mathrm{RBF} + \mathrm{SM}_1 + \mathrm{SM}_2 + \mathrm{WN}. \tag{9}$$

The periodic kernel (PER) captures the seasonality of the time series. LIN captures the linear trend. Long-term trends are generally smooth, and can be properly modelled by the RBF kernel. The two SM kernels are used to pick up the remaining signal. Finally, the WN kernel represents the observation (Gaussian) noise.

parameter	ν	λ
variance	-1.5	1.0
	lengthscales	
std_periodic	0.2	1.0
rbf	1.1	1.0
SM_1	-0.7	1.0
SM_2	1.1	1.0

Table 2. Parameters of the lognormal priors. The same prior is adopted for the variances of all components in Eq. (9)

This results in a kernel capturing a wide range of patterns but comprising 16 hyperparameters, which must be estimated from data. This might be challenging on short time series, such as monthly or quarterly ones. In [3] the problem is addressed by setting priors on the hyperparameters. In particular, lognormal priors are adopted and they are defined through a hierarchical Bayes approach, i.e., by analyzing a subset of monthly time series from the M3 competition. The priors, which we also adopt, are given in Tab.2.

2.5 SS approximation

To achieve $O(n)$ complexity, we replace the kernel in (9) with this approximation

$$\tilde{K} = (+_8\text{COS}) + \text{LIN} + \text{MAT32} + \text{COS} \times \text{MAT32} + \text{COS} \times \text{MAT32} + \text{WN}. \tag{10}$$

Note we have approximated PER with the sum of 7 COS kernel and RBF with MAT32.[8] A GP with the above kernel can equivalently be represented by a SS model who state has dimension $7 \times 2 + 2 + 2 + 4 + 4 + 1 = 24$.

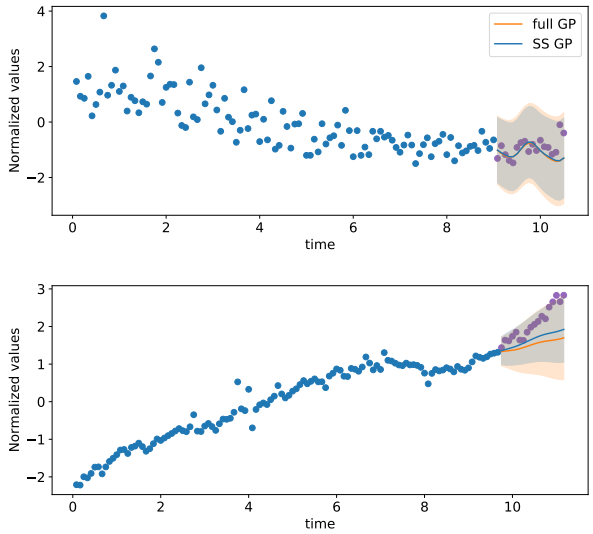

Fig. 1. Comparison of GP and SS forecasts. The blue dots are the training data and the purple dots the test data. The small differences between full GP and SS are due to the slightly different estimation of the hyperparameters. The time series are monthly and the forecasts are computed up to 1.5 years ahead; time is expressed in years.

Figure 1 compares the GP estimate and forecast based on the kernel (9) and the SS approximation based on the kernel (10) on some time series from the M3 competition.[9] The SS approximation provides close forecasts to the full GP. We provide a more in-depth analysis when discussing the experiments.

2.6 Combining GP kernel with exponential smoothing

Our framework is so flexible, that it allows combining the state-space representations of covariance functions and existing state-space models, thus obtaining some novel time series models.

[8] We also tried a more accurate approximation of the periodic kernel, 11 COS kernels, but it did not provide a significant better performance in the M3 competition.

[9] In both cases, we have estimated the kernels hyperparameters using MAP.

As a proof of concept, we consider state-space additive exponential smoothing (*additive ets*), and we replace its seasonal component with the PER kernel.

The discrete-time SS representation of exponential smoothing with linear trend is [8]:

$$\text{Holt:} \begin{cases} f_1((k+1)\Delta_t) = f_1(k\Delta_t) + f_2(k\Delta_t) + \alpha w((k+1)\Delta_t) \\ f_2((k+1)\Delta_t) = f_2(k\Delta_t) + \alpha\beta w((k+1)\Delta_t) \\ y((k+1)\Delta_t) = f_1(k\Delta_t) + f_2(k\Delta_t) + w((k+1)\Delta_t) \end{cases} \quad \begin{bmatrix} f_1(t_0) \\ f_2(t_0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} s_l^2 & 0 \\ 0 & s_b^2 \end{bmatrix} \right)$$

where Δ_t is the sampling frequency and w are independent Gaussian noises with zero mean and variance s_v^2. Such model model has five parameters: $\alpha, \beta \in [0, 1]$ and s_l^2, s_b^2, s_v^2.

We then complete the SS model by adding the (approximated) SS representation of the PER kernel, constituted by the sum of seven COS covariance functions. When estimating the hyperparameters, automatic relevance determination (ARD) automatically makes irrelevant the unnecessary component, without the need for a separate model selection step.[10]

3 Experiments

We consider the following GP models:

- full-GP: the model of Eq. (9), trained with priors [3];
- full-GP$_0$: the same model, trained by maximizing the marginal likelihood (no priors);
- SS-GP and SS-GP$_0$, i.e., the corresponding SS models (Eq. 10) trained with and without priors.

We use a single restart when training all the models.

As benchmarks, we consider *auto.arima* and *ets*, both available from the forecast package [9]. The *auto.arima* algorithm first makes the time series stationary via differentiation; then it fits an ARMA model selecting the orders via AICc. The *ets* algorithm fits several state-space exponential smoothing models [8], characterized by different types of trend, seasonality and noise; the best model is eventually chosen via AICc. All the considered models represent the forecast uncertainty via a Gaussian distribution.

Metrics As performance metric, we consider the mean absolute error (MAE) on the test set:

$$\text{MAE} = \sum_{t=1}^{T} |y_t - \hat{y}_t|$$

[10] For the variances of the Holt's model we use the same priors as in Table 2. For α, β, we use the prior Beta$(1, 1.4)$ and, respectively, Beta$(1, 11.4)$. We learned the parameters of these priors using a hierarchical model similar to the one described in [3].

where we denote by y_t and $\hat{y}_t$ the actual value and the expected value of the time series at time t; σ_t^2 denotes the variance of the forecast at time t and by T the length of the test set.

Furthermore, we compute the continuous-ranked probability score (CRPS) [5], which generalizes the MAE to the case of probabilistic forecasts. It is a proper scoring rule for probabilistic forecasts, which corresponds to the integral of the Brier scores over the continuous predictive distribution. MAE and CRPS are loss functions, hence the lower the better.

3.1 Monthly M3

	median		mean	
Algorithm	MAE	CRPS	MAE	CRPS
SS-GP	**0.489**	**0.342**	**0.567**	**0.421**
full-GP	**0.482**	**0.347**	**0.565**	**0.414**
SS-GP$_0$	0.550	0.408	0.627	0.499
full-GP$_0$	0.546	0.390	0.628	0.460
ETS	0.516	0.369	0.595	0.436
Auto.arima	0.515	0.373	0.588	0.430

Table 3. Performance on the M3 monthly time series.

The M3 competition includes 1489 monthly time series. We exclude 350 of them, which were used in [3] to define the priors of Tab.2, which we also adopt. We thus run experiments on the remaining 1079 monthly time series. The length of training set varies between 49 and 126 months, while the test set is always 18 months long. We standardize each time series using the mean and the standard deviation of the training set. We fix the period of the periodic kernel to one year, which is standard practice for M3.

The median and mean results for time series are given in Tab. 3. The SS-GP and full-GP obtain the best median and mean performance on all indicators. The performance of full-GP and of its state-space representation is practically identical, showing that the SS approximation is very accurate. We tried also Prophet [23] but its accuracy was not competitive. We thus dropped it.

The large improvement of full-GP and SS-GP over full-GP$_0$ and SS-GP$_0$ confirms that the priors are necessary to exploit the potential of the GP.

3.2 Combining GP kernel and exponential smoothing

The SS representation of GPs allows us to combine GPs with state-of-the-art models for time series forecasting such the ETS model [8].

In this section, we compare the SS model discussed previously, which uses the following kernel:

$$\tilde{K}_1 = (+_7\text{COS}) + \text{Holt}, \tag{11}$$

where the Holt kernel has been defined in Sec.2.6.

We compare this model with *additive ETS* model, defined as follows. The additive ets model fits four different models via maximum likelihood and performs model selection via AICc. The four models are simple exponential smoothing (*ses*, no trend and no seasonality), *ses* with linear trend, *ses* with linear trend and additive seasonality, *ses* with additive seasonality but no trend. We implement all such models using the forecast package for R [9]. The *ets* framework makes available also multiplicative models, that however we do not consider in this section.

The seasonal component of exponential smoothing has some shortcomings: it requires to estimate $(m+1)$ parameters, where m denotes then number of samples within a period (e.g., m=12 for monthly time series); moreover, it does not manage complex seasonalities such non-integer periods or multiple seasonal pattern. In our model we thus substitute it with the PER kernel (equivalently $(+_7\text{COS})$ kernel), which has only two (hyper)-parameters and which can model complex seasonalities (e.g, multiple seasonalities can be modelled by adding multiple PER kernels).

Therefore, the main differences between additive ets and our novel model are thus:

- PER kernel vs seasonal component of exponential smoothing;
- automatic relevance determination vs model selection.

The simulation results are shown in Table 4. SS-GP is again the best model. Comparing SS-GP performance in Table 3 and 4 is evident that the more complex kernel (10) provides a better the performance. However, this shows how the SS representation of GPs opens up the possibility of developing novel time series models combining traditional time series models with "machine-learning-like" models.

	median		mean	
Algorithm	MAE	CRPS	MAE	CRPS
SS-GP	**0.511**	**0.368**	**0.581**	**0.436**
SS-GP$_0$	0.538	0.387	0.608	0.461
Additive ETS	0.533	0.381	0.601	0.439

Table 4. Performance on M3 monthly. SS-GP with kernel $\tilde{K}_1$ compared to additive ETS.

3.3 Large datasets and multiple seasonality

By contrast to full GP, SS models can scale to large datasets. We provide a proof-of-concept of that by applying the SS model to two time series in the UCI's Electricity Dataset. Each time series is relative to the electricity consumption

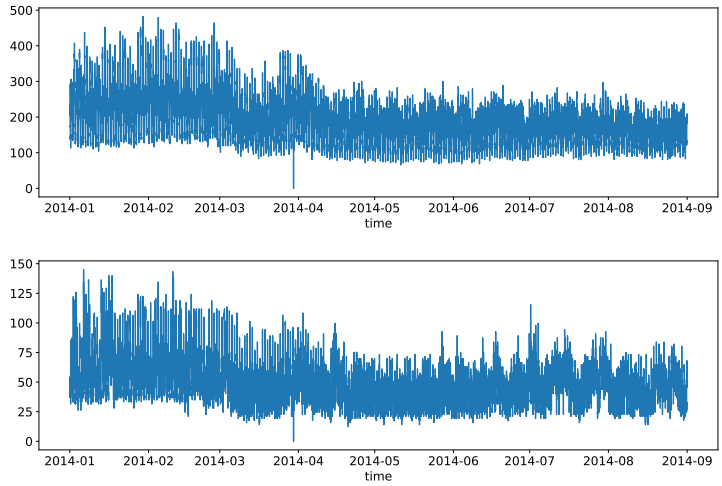

Fig. 2. Two time series taken from the Electricity Dataset

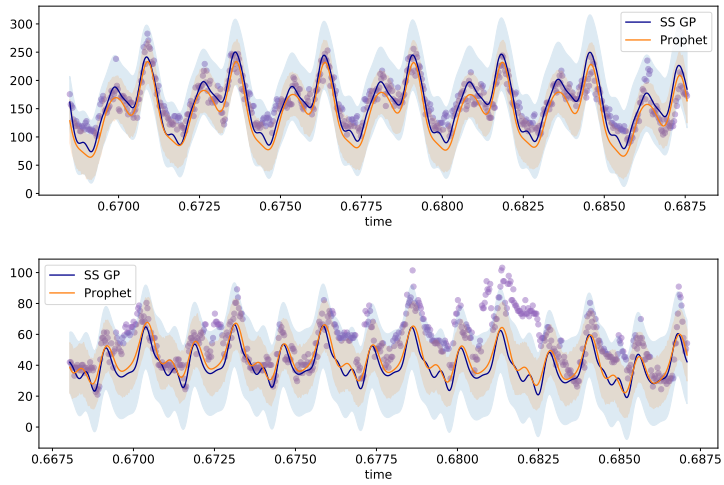

Fig. 3. One week ahead forecast computed by (i) the proposed SS model; (ii) Facebook's Prophet; for the two time series in Figure 2. The time has been normalized: 1 is one year.

of client from a period of 2011 to 2014 at an interval of 15 minutes. The goal is to forecast the electricity consumption one week ahead. The length of each time series is 23997 and, therefore, we cannot run full GP (on a standard PC). Moreover, the time series have both daily and weekly periodicity, which means the kernel in (10) is not appropriate.

However, we can easily deal with multiple seasonality by adding another periodic component to the kernel:

$$\tilde{K} = (+_7\text{COS})+(+_7\text{COS}) + \text{LIN} + \text{MAT32} + \text{COS} \times \text{MAT32} + \text{COS} \times \text{MAT32} + \text{WN} \tag{12}$$

where the first periodic kernel (the term $(+_7\text{COS})$) has period 1/365.25 and the second 7/365.25.[11]

Figure 2 shows two time series taken from the Electricity Dataset. Figure 3 reports the relative one week ahead forecast computed by (i) the proposed SS model; (ii) Facebook's Prophet. The training times are of few seconds for Prophet, and about 300 seconds for the SS model.

While our implementation is currently slower than Prophet, it already handles flawlessly this time series. The training time of our implementation can be largely reduced by using Stochastic Gradient (SGD) optimization, thus working with minibatch of data. The forecasts show that the SS model is competitive also on long time series; however, the analysis of a large number of time series is needed in order to achieve conclusions which are significant. We defer this analysis to future work, after the completion of a faster implementation of SS-GP based on SGD.

4 Conclusions

Focusing on time series forecasting, we have shown that a Gaussian Process with a complex composite kernel can be accurately approximated by a state space model. The resulting state space model has a comparable performance, but with a complexity which scales linearly in the input size. Moreover, given state-of-the-art models for time series forecasting are implemented in state space form, the state space representation of Gaussian Processes allowed us to combine traditional models (like exponential smoothing) with kernel-based models (like periodic kernel) in a sound Bayesian manner.

Several future research directions are possible. One is the extension to time series characterized by non-Gaussian likelihoods, such as count time series or intermittent time series. Other possibilities include the combination of exponential smoothing with the spectral mixture or the Neural Network kernel. We also plan to compare our approach with other Generalised Additive (Mixture) Models used for time-series forecasting.

Acknowledgements The authors acknowledge support from the Swiss National Research Programme 75 "Big Data" Grant No. 407540_167199/1.

[11] By contrast to arima and ETS, GP and SS models can easily model non-integer seasonality like the ones in the Electricity dataset, see [3] for more details.

References

1. Bauer, M., van der Wilk, M., Rasmussen, C.E.: Understanding probabilistic sparse Gaussian process approximations. In: Advances in neural information processing systems. pp. 1533–1541 (2016)
2. Benavoli, A., Zaffalon, M.: State Space representation of non-stationary Gaussian processes. arXiv preprint arXiv:1601.01544 (2016)
3. Corani, G., Benavoli, A., Zaffalon, M.: Time series forecasting with Gaussian Processes needs priors. Proc. ECML PKDD (accepted) (2021), `https://arxiv.org/abs/2009.08102`
4. Foreman-Mackey, D., Agol, E., Ambikasaran, S., Angus, R.: Fast and scalable Gaussian process modeling with applications to astronomical time series. The Astronomical Journal **154**(6), 220 (2017)
5. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association **102**(477), 359–378 (2007)
6. Hensman, J., Fusi, N., Lawrence, N.D.: Gaussian processes for big data. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence. p. 282–290. UAI'13, AUAI Press, Arlington, Virginia, USA (2013)
7. Hernández-Lobato, D., Hernández-Lobato, J.M.: Scalable gaussian process classification via expectation propagation. In: Artificial Intelligence and Statistics. pp. 168–176 (2016)
8. Hyndman, R.J. & Athanasopoulos, G.: Forecasting: principles and practice, 2nd edition,. OTexts: Melbourne, Australia (2018), `OTexts.com/fpp2`
9. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. Journal of Statistical Software **26**(3), 1–22 (2008), `http://www.jstatsoft.org/article/view/v027i03`
10. Jazwinski, A.H.: Stochastic processes and filtering theory. Courier Corporation (2007)
11. Karvonen, T., Sarkkä, S.: Approximate state-space gaussian processes via spectral transformation. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2016)
12. Lloyd, J.R.: GEFCom2012 hierarchical load forecasting: Gradient boosting machines and Gaussian processes. International Journal of Forecasting **30**(2), 369–374 (2014)
13. Loper, J., Blei, D., Cunningham, J.P., Paninski, L.: General linear-time inference for gaussian processes on one dimension. arXiv preprint arXiv:2003.05554 (2020)
14. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. The Journal of Machine Learning Research **6**, 1939–1959 (2005)
15. Rasmussen, C., Williams, C.: Gaussian processes for machine learning. Gaussian Processes for Machine Learning (2006)
16. Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S.: Gaussian processes for time-series modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **371**(1984), 20110550 (2013)
17. Särkkä, S., Hartikainen, J.: Infinite-dimensional kalman filtering approach to spatio-temporal gaussian process regression. In: International Conference on Artificial Intelligence and Statistics. pp. 993–1001 (2012)
18. Sarkka, S., Solin, A., Hartikainen, J.: Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. Signal Processing Magazine, IEEE **30**(4), 51–61 (2013)

19. Schuerch, M., Azzimonti, D., Benavoli, A., Zaffalon, M.: Recursive estimation for sparse Gaussian process regression. Automatica **120**, 109–127 (2020)
20. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: Advances in neural information processing systems. pp. 1257–1264 (2006)
21. Solin, A., Särkkä, S.: Explicit link between periodic covariance functions and state space models. In: Artificial Intelligence and Statistics. pp. 904–912. PMLR (2014)
22. Solin, A., Sarkka, S.: Gaussian quadratures for state space approximation of scale mixtures of squared exponential covariance functions. In: Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on. pp. 1–6. IEEE (2014)
23. Taylor, S.J., Letham, B.: Forecasting at scale. The American Statistician **72**(1), 37–45 (2018)
24. Titsias, M.: Variational learning of inducing variables in sparse Gaussian processes. In: van Dyk, D., Welling, M. (eds.) Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 5, pp. 567–574. PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA (16–18 Apr 2009)
25. Wilson, A., Adams, R.: Gaussian process kernels for pattern discovery and extrapolation. In: International conference on machine learning. pp. 1067–1075. PMLR (2013)
26. Wood, S.N.: Generalized additive models: an introduction with R. CRC press (2017)