Computationally Efficient Bayesian Learning of Gaussian Process State Space Models

Andreas Svensson
Uppsala University, Sweden

Arno Solin
Aalto University, Finland

Simo Särkkä
Aalto University, Finland

Thomas B. Schön
Uppsala University, Sweden

Abstract

Gaussian processes allow for flexible specification of prior assumptions of unknown dynamics in state space models. We present a procedure for efficient Bayesian learning in Gaussian process state space models, where the representation is formed by projecting the problem onto a set of approximate eigenfunctions derived from the prior covariance structure. Learning under this family of models can be conducted using a carefully crafted particle MCMC algorithm. This scheme is computationally efficient and yet allows for a fully Bayesian treatment of the problem. Compared to conventional system identification tools or existing learning methods, we show competitive performance and reliable quantification of uncertainties in the model.

1 INTRODUCTION

Gaussian processes (GPs, Rasmussen and Williams 2006) have been proven to be powerful probabilistic non-parametric modeling tools for *static* nonlinear functions. However, many real-world applications, such as control, target tracking, and time-series analysis are tackling problems with nonlinear *dynamical* behavior. The use of GPs in modeling nonlinear dynamical systems is an emerging topic, with many strong contributions during the recent years, for example the work by Turner et al. (2010), Frigola et al. (2013, 2014a,b) and Mattos et al. (2016). The aim of this paper is to advance the state-of-the-art in Bayesian inference on Gaussian process state space models (GP-SSMs). As we will detail, a GP-SSM is a state space model, using a GP as its state transition function. Thus, the GP-SSM is not a GP itself, but a state space model (*i.e.*, a dynamical system). Overviews of GP-SSMs are given by, *e.g.*, McHutchon (2014) and Frigola-Alcade (2015).

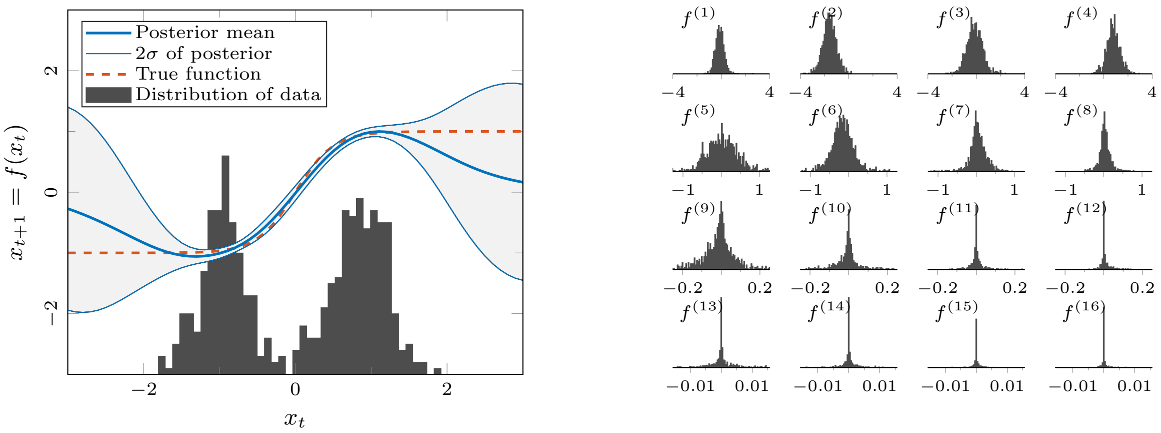
We provide a novel reduced-rank model formulation of the GP-SSM with good convergence properties both in theory and practice. The advantage with our approach over the variational approach by Frigola et al. (2014b), as well as other inducing-point-based approaches, is that our approach attempts to approximate the optimal Karhunen–Loeve eigenbasis for the reduced-rank approximation instead of using the sub-optimal Nyström approximation which implicitly is the underlying approximation in all inducing point methods. Because of this we do not need to resort to variational approximations, but we can instead perform the Bayesian computations in full. By utilizing the structure of the reduced-rank model, we construct a computationally efficient linear-time-complexity MCMC-based algorithm for learning in the proposed GP-SSM model, which we demonstrate and evaluate on several challenging examples. We also provide a proof of convergence of the reduced-rank GP-SSM to a full GP-SSM (in the supplementary material).

GP-SSMs are a general class of models defining a dynamical system for $t = 1, 2, \dots, T$ given by

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{f}(\mathbf{x}_t) + \mathbf{w}_t, \\ &\qquad \text{with } \mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\kappa}_{\theta,f}(\mathbf{x}, \mathbf{x}')), \qquad (1a)\\ \mathbf{y}_t &= \mathbf{g}(\mathbf{x}_t) + \mathbf{e}_t, \\ &\qquad \text{with } \mathbf{g}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\kappa}_{\theta,g}(\mathbf{x}, \mathbf{x}')), \qquad (1b)\end{aligned}$$

where the noise terms $\mathbf{w}_t$ and $\mathbf{e}_t$ are i.i.d. Gaussian, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. The latent state $\mathbf{x}_t \in \mathbb{R}^{n_x}$ is observed via the measurements $\mathbf{y}_t \in \mathbb{R}^{n_y}$. The key feature of this model is the nonlinear transformations $\mathbf{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $\mathbf{g} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ which are not known explicitly and do not adhere to any specific parametrization. The model functions $\mathbf{f}$ and $\mathbf{g}$ are assumed to be realizations from a Gaussian process prior over $\mathbb{R}^{n_x}$ with a given covariance function

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.



(a) The learned model

(b) The posterior weights $f^{(i)}$

Figure 1: An example illustrating how the GP-SSMs handle uncertainty. (a) The learned model from data $y_{1:T}$. The bars show where the data is located in the state space, *i.e.*, what part of the model is excited in the data set, affecting the posterior uncertainty in the learned model. (b) Our approach relies on a basis function expansion of f, and learning f amounts to finding the posterior distribution of the weights $f^{(i)}$ depicted by the histograms.

$\boldsymbol{\kappa}_\theta(\mathbf{x}, \mathbf{x}')$ subject to some hyperparameters $\boldsymbol{\theta}$. Learning of this model, which we will tackle, amounts to inferring the posterior distribution of $\mathbf{f}$, $\mathbf{g}$, $\mathbf{Q}$, $\mathbf{R}$, and $\boldsymbol{\theta}$ given a set of (noisy) observations $\mathbf{y}_{1:T} \triangleq \{\mathbf{y}_i\}_{i=1}^T$.

The strength of including the GP in (1) is its ability to systematically model *uncertainty*—not only uncertainty originating from stochastic noise within the system, but also uncertainty inherited from data, such as few measurements or poor excitation of the dynamics in certain regions of the state space. An example of this is given by Figure 1, where we learn the posterior *distribution* of the unknown function $\mathbf{f}(\cdot)$ in a GP-SSM (see Sec. 5 for details). An inspiring real-world example on how such probabilistic information can be utilized for simultaneous learning and control is given by Deisenroth et al. (2015).

Non-probabilistic methods for modeling nonlinear dynamical systems include learning of state space models using a basis function expansion (Ghahramani and Roweis, 1998), but also nonlinear extensions of AR(MA) and GARCH models from the time-series analysis literature (Tsay, 2010), as well as nonlinear extensions of ARX and state space models from the system identification literature (Sjöberg et al., 1995; Ljung, 1999). In particular, nonlinear ARX models are now a standard tool for the system identification engineer (The MathWorks, Inc., 2015). For probabilistic modeling, the latent force model (Alvarez et al., 2009) presents one approach for modeling dynamical phenomena using GPs by encoding *a priori* known dynamics within the construction of the GP. Another approach is the Gaussian process dynamical model (Wang et al., 2008), where a GP is used to model the nonlinear function within an SSM, that is, a GP-SSM. However, the work by Wang et al. (2008) is, as opposed to this paper, mostly focused around the problem setting when $n_y \gg n_x$. That is also the focus for the further development by Damianou et al. (2011), where the EM algorithm for learning is replaced by a variational approach.

State space filtering and smoothing in GP-SSMs has been tackled before (*e.g.*, Deisenroth et al. 2012; Deisenroth and Mohamed 2012), and recent interest has been in learning GP-SSMs (Turner et al., 2010; Frigola et al., 2013, 2014a,b). An inherent problem in learning the GP-SSM is the entangled relationship between the states $\mathbf{x}_t$ and the nonlinear function $\mathbf{f}(\cdot)$. Two different approaches have been proposed in the literature: In the first approach the GP is represented by a parametrized form (Turner et al. use a pseudo-training data set, akin to the inducing inputs by Frigola et al. 2014b, whereas we will employ a basis function expansion). The second approach (used by Frigola et al. 2013, 2014a) is handling the nonlinear function implicitly by marginalizing it out. Concerning learning, Turner et al. (2010) and Frigola et al. (2014a) use an EM-based procedure, whereas we and Frigola et al. (2013) use an MCMC algorithm.

The main bottleneck prohibiting the use in practice of some of the previously proposed GP-SSMs methods is the computational load. For example, the training of a one-dimensional system using $T = 500$ data points (*i.e.*, a fairly small example) is in the magnitude of several hours for the solution by Frigola et al. (2013).

Akin to Frigola et al. (2014b), our proposed method will typically handle such an example within minutes, or even less. To reduce the computational load, Frigola et al. (2014b) suggests variational sparse GP techniques to approximate the solution. Our approach, however, is using the reduced-rank GP approximation by Solin and Särkkä (2014), which is a disparate solution with different properties. The reduced-rank GP approximation enjoys favorable theoretical properties, and we can prove convergence to a non-approximated GP-SSM.

The outline of the paper is as follows: In Section 2 we will introduce reduced-rank Gaussian process state space models by making use of the representation of GPs via basis functions corresponding to the prior covariance structure (Solin and Särkkä, 2014), a theoretically well-supported approximation significantly reducing the computational load. In Section 3 we will develop an algorithm for learning reduced-rank Gaussian process state space models by using recent MCMC methods (Lindsten et al., 2014; Wills et al., 2012). We will also demonstrate it on synthetic as well as real data examples in Section 5, and finally discuss the contribution and further extensions in Section 6.

2 REDUCED-RANK GP-SSMs

We use GPs as flexible priors in Bayesian learning of the state space model. The covariance function $\kappa(\mathbf{x}, \mathbf{x}')$ encodes the prior assumptions of the model functions, thus representing the best belief of the behavior of the nonlinear transformations. In the following we present an approach for parametrizing this model in terms of an m-rank truncation of a basis function expansion as presented by Solin and Särkkä (2014). Related ideas have also been proposed by, for example, Lázaro-Gredilla et al. (2010).

Provided that the covariance function is stationary (homogeneous, *i.e.* $\kappa(\mathbf{x} - \mathbf{x}') \triangleq \kappa(\mathbf{x}, \mathbf{x}')$), the covariance function can be equivalently represented in terms of the spectral density $S(\boldsymbol{\omega})$. This Fourier duality is known as the *Wiener–Khintchin theorem*, which we parametrize as: $S(\boldsymbol{\omega}) = \int \kappa(\mathbf{r}) \exp(-\mathrm{i}\,\boldsymbol{\omega}^\mathsf{T}\mathbf{r})\,\mathrm{d}\mathbf{r}$. We employ the relation presented by Solin and Särkkä (2014) to approximate the covariance operator corresponding to $\kappa(\cdot)$. This operator is a pseudo-differential operator, which we approximate by a series of differential operators, namely Laplace operators ∇^2. In the isotropic case, the approximation of the covariance function is given most concisely in the following form:

$$\kappa_\theta(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{m} S_\theta(\lambda_j)\, \phi^{(j)}(\mathbf{x})\, \phi^{(j)}(\mathbf{x}'), \tag{2}$$

where $S_\theta(\cdot)$ is the spectral density function of $\kappa_\theta(\cdot)$, and λ_j and $\phi^{(j)}$ are the Laplace operator eigenvalues and eigenfunctions solved for the domain $\Omega \ni \mathbf{x}$. See Solin and Särkkä (2014) for a detailed derivation and convergence proofs.

The key feature in the Hilbert space approximation (2) is that λ_j and $\phi^{(j)}$ are independent of the hyperparameters $\boldsymbol{\theta}$, and it is only the spectral density that depends on $\boldsymbol{\theta}$. Equation (2) is a direct approximation of the eigendecomposition of the Gram matrix (*e.g.*, Rasmussen and Williams 2006), and it can be interpreted as an optimal parametric expansion with respect to the given covariance function in the GP prior.

In terms of a basis function expansion, this can be expressed as

$$f(\mathbf{x}) \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}')) \quad \Leftrightarrow \quad f(\mathbf{x}) \approx \sum_{j=1}^{m} f^{(j)} \phi^{(j)}(\mathbf{x}), \tag{3}$$

where $f^{(j)} \sim \mathcal{N}(0, S(\lambda_j))$. In the case $n_x > 1$, this formulation does allow for non-zero covariance between different components of the state space. We can now formulate a reduced-rank GP-SSM, corresponding to (1a), as

$$\mathbf{x}_{t+1} = \underbrace{\begin{bmatrix} f_1^{(1)} & \dots & f_1^{(m)} \\ \vdots & & \vdots \\ f_{n_x}^{(1)} & \dots & f_{n_x}^{(m)} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \phi^{(1)}(\mathbf{x}_t) \\ \vdots \\ \phi^{(m)}(\mathbf{x}_t) \end{bmatrix}}_{\mathbf{\Phi}(\mathbf{x}_t)} + \mathbf{w}_t, \tag{4}$$

and similarly for (1b). Henceforth we will consider a reduced-rank GP-SSM,

$$\mathbf{x}_{t+1} = \mathbf{A}\Phi(\mathbf{x}_t) + \mathbf{w}_t, \tag{5a}$$

$$\mathbf{y}_t = \mathbf{C}\Phi(\mathbf{x}_t) + \mathbf{e}_t, \tag{5b}$$

where $\mathbf{A}$ and $\mathbf{C}$ are matrices of weights with priors for each element as described by (3).

3 LEARNING GP-SSMs

Learning in reduced-rank Gaussian process state space models (5) from $\mathbf{y}_{1:T}$ amounts to inferring the posterior distribution of $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, $\mathbf{R}$, and the hyperparameters $\boldsymbol{\theta}$. For clarity in the presentation, we will focus on inferring the dynamics, and assume the observation model ($\mathbf{g}(\cdot)$ and $\mathbf{R}$) to be known *a priori*. However, the extension to an unknown observation model—as well as exogenous input signals—follows in the same fashion, and will be demonstrated in the numerical examples.

To infer the sought distributions, we will use a blocked Gibbs sampler outlined in Algorithm 1. Although involving sequential Monte Carlo (SMC) for inference in

Algorithm 1 Learning of reduced-rank GP-SSMs.

```
Input: Data $y_{1:T}$, priors on $\mathbf{A}$, $\mathbf{Q}$ and $\boldsymbol{\theta}$.
Output: $K$ MCMC-samples with $p(\mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A}, \boldsymbol{\theta} \mid \mathbf{y}_{1:T})$ as invariant distribution.
1: Sample initial $\mathbf{x}_{1:T}[0], \mathbf{Q}[0], \mathbf{A}[0], \boldsymbol{\theta}[0]$.
2: for $k = 0$ to $K$ do
3:     Sample $\mathbf{x}_{1:T}[k+1] \mid \mathbf{Q}[k], \mathbf{A}[k], \boldsymbol{\theta}[k]$                          by Algorithm 2.
4:     Sample $\mathbf{Q}[k+1] \mid \mathbf{A}[k], \boldsymbol{\theta}[k], \mathbf{x}_{1:T}[k+1]$                  according to (10).
5:     Sample $\mathbf{A}[k+1] \mid \boldsymbol{\theta}[k], \mathbf{x}_{1:T}[k+1], \mathbf{Q}[k+1]$              according to (11).
6:     Sample $\boldsymbol{\theta}[k+1] \mid \mathbf{x}_{1:T}[k+1], \mathbf{Q}[k+1], \mathbf{A}[k+1]$   by using MH (Section 3.3).
7: end for
```

state space, the validity of this approach is *not* relying on asymptotics ($N \to \infty$) in the SMC algorithm, thanks to recent particle MCMC methods (Lindsten et al., 2014; Andrieu et al., 2010).

It is possible to learn (5) under different assumptions on what is known. We will focus on the general (and in many cases realistic) setting where the distributions of $\mathbf{A}$, $\mathbf{Q}$ and $\boldsymbol{\theta}$ are all unknown. In cases when $\mathbf{Q}$ or $\boldsymbol{\theta}$ are known *a priori*, the presented scheme is straightforward to adapt. To be able to infer the posterior distribution of $\mathbf{Q}$ and $\boldsymbol{\theta}$, we make the additional prior assumptions:

$$\mathbf{Q} \sim \mathcal{IW}(\ell_Q, \boldsymbol{\Lambda}_Q), \qquad\qquad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \qquad (6)$$

where $\mathcal{IW}$ denotes the Inverse Wishart distribution. For brevity, we will omit the problem of finding the unknown initial distribution $p(\mathbf{x}_1)$. It is possible to treat this rigorously akin to $\boldsymbol{\theta}$, but it is of minor importance in most practical situations. We will now in Section 3.1–3.3 explain the four main steps 3–6 in Algorithm 1.

3.1 Sampling in State Space with SMC

SMC methods (Doucet and Johansen, 2011) are a family of techniques developed around the problem of inferring the posterior state distribution in SSMs. SMC can be seen as a sequential application of importance sampling along the sequence of distributions $\dots, p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}), p(\mathbf{x}_t \mid \mathbf{y}_{1:t}), \dots$ with a resampling procedure to avoid sample depletion.

To sample the state space trajectory $\mathbf{x}_{1:T}$, conditional on a model $\mathbf{A}$, $\mathbf{Q}$ and data $\mathbf{y}_{1:T}$, we employ a conditional particle filter with ancestor sampling, forming a particle Gibbs Markov kernel Algorithm 2 (PGAS, Lindsten et al. 2014). PGAS can be thought of as an SMC algorithm for finding the so-called smoothing distribution $p(\mathbf{x}_{1:T} \mid \mathbf{A}, \mathbf{Q}, \mathbf{y}_{1:T})$ to be used within an MCMC procedure.

3.2 Sampling of Covariances and Weights

The sampling of the weights $\mathbf{A}$ and the noise covariance $\mathbf{Q}$, conditioned on $\mathbf{x}_{1:T}$ and $\boldsymbol{\theta}$, can be done exactly, by following the procedure of Wills et al. (2012). With the priors (3) and (6), the joint prior of $\mathbf{A}$ and $\mathbf{Q}$ can be written using the Matrix Normal Inverse Wishart (MNIW) distribution as

$$p(\mathbf{A}, \mathbf{Q}) = \mathcal{MNIW}(\mathbf{A}, \mathbf{Q} \mid \mathbf{0}, \mathbf{V}, \ell_Q, \boldsymbol{\Lambda}_Q). \qquad (7)$$

Details on the parametrization of the MNIW distribution we use is available in the supplementary material, and it is given by the hierarchical model $p(\mathbf{Q}) = \mathcal{IW}(\mathbf{Q} \mid \ell_Q, \boldsymbol{\Lambda}_Q)$ and $p(\mathbf{A} \mid \mathbf{Q}) = \mathcal{MN}(\mathbf{A} \mid \mathbf{0}, \mathbf{Q}, \mathbf{V})$. For our problem, the most important is the second argument, the inverse row covariance $\mathbf{V}$, a square matrix with the inverse spectral density of the covariance function as its diagonal entries:

$$\mathbf{V} = \operatorname{diag}\left(\left[S^{-1}(\lambda_1) \ \cdots \ S^{-1}(\lambda_m)\right]\right). \qquad (8)$$

This is how the prior from (3) enters the formulation. (Note that the marginal variance of each element in $\mathbf{A}$ is also scaled $\mathbf{Q}$, and thereby $\ell_Q, \boldsymbol{\Lambda}_Q$. For notational convenience, we refrain from introducing a scaling factor, but let it be absorbed into the covariance function.) With this (conjugate) prior, the posterior follows analytically by introducing the following statis-

Algorithm 2 Particle Gibbs Markov kernel.

```
Input: Trajectory $\mathbf{x}_{1:T}[k]$, number of particles $N$
Output: Trajectory $\mathbf{x}_{1:T}[k+1]$
Sample $\mathbf{x}_1^{(i)} \sim p(\mathbf{x}_1)$, for $i = 1, \dots, N-1$.
Set $\mathbf{x}_1^N = \mathbf{x}_1[k]$.
For $t = 1$ to $T$
  Set $w_t^{(i)} = p(\mathbf{y}_t \mid \mathbf{x}_t^{(i)}) = \mathcal{N}(\mathbf{g}(\mathbf{x}_t^{(i)}) \mid \mathbf{y}_t, \mathbf{R})$, for $i = 1, \dots, N$.
  Sample $a_t^{(i)}$ with $\mathbb{P}(a_t^{(i)} = j) \propto w_t^{(j)}$, for $i = 1, \dots, N-1$.
  Sample $\mathbf{x}_{t+1}^{(i)} \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_t^{a_t^{(i)}}), \mathbf{Q})$, for $i = 1, \dots, N-1$.
  Set $\mathbf{x}_{t+1}^N = \mathbf{x}_{t+1}[k]$.
  Sample $a_t^N$ with $\mathbb{P}(a_t^N = j) \propto$
  $w_t^{(j)} p(\mathbf{x}_{t+1}^N \mid \mathbf{x}_t^{(j)}) = w_t^{(j)} \mathcal{N}(\mathbf{x}_{t+1}^N \mid \mathbf{f}(\mathbf{x}_t^{(j)}), \mathbf{Q})$.
  Set $\mathbf{x}_{1:t+1}^{(i)} = \{\mathbf{x}_{1:t}^{a_t^{(i)}}, \mathbf{x}_{t+1}^{(i)}\}$, for $i = 1, \dots, N$.
End for
Sample $J$ with $\mathbb{P}(J = i) \propto w_T^{(i)}$ and set $\mathbf{x}_{1:T}[k+1] = \mathbf{x}_{1:T}^J$.
```

tics of the sampled trajectory $\mathbf{x}_{1:T}$:

$$\boldsymbol{\Phi} = \sum_{t=1}^{T} \boldsymbol{\zeta}_t \boldsymbol{\zeta}_t^{\mathsf{T}}, \quad \boldsymbol{\Psi} = \sum_{t=1}^{T} \boldsymbol{\zeta}_t \mathbf{z}_t^{\mathsf{T}}, \quad \boldsymbol{\Sigma} = \sum_{t=1}^{T} \mathbf{z}_t \mathbf{z}_t^{\mathsf{T}}, \tag{9}$$

where $\boldsymbol{\zeta}_t = \mathbf{x}_{t+1}$ and $\mathbf{z}_t = \left[\phi^{(1)}(\mathbf{x}_t) \dots \phi^{(m)}(\mathbf{x}_t)\right]^{\mathsf{T}}$. Using the Markov property of the SSM, it is possible to write the conditional distribution for $\mathbf{Q}$ as (Wills et al., 2012, Eq. (42)):

$$p(\mathbf{Q} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = \\ \mathcal{IW}(\mathbf{Q} \mid T + \ell_Q, \boldsymbol{\Lambda}_Q + \left(\boldsymbol{\Phi} - \boldsymbol{\Psi}(\boldsymbol{\Sigma} + \mathbf{V})^{-1}\boldsymbol{\Psi}^{\mathsf{T}}\right)). \tag{10}$$

Given the prior (7), $\mathbf{A}$ can now to be sampled from (Wills et al., 2012, Eq. (43)):

$$p(\mathbf{A} \mid \mathbf{Q}, \mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = \\ \mathcal{MN}(\mathbf{A} \mid \boldsymbol{\Psi}(\boldsymbol{\Sigma} + \mathbf{V})^{-1}, \mathbf{Q}, (\boldsymbol{\Sigma} + \mathbf{V})^{-1}). \tag{11}$$

3.3 Marginalizing the Hyperparameters

Concerning the sampling of the hyperparameters $\boldsymbol{\theta}$, we note that we can easily evaluate the conditional distribution $p(\boldsymbol{\theta} \mid \mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A})$ up to proportionality as

$$p(\boldsymbol{\theta} \mid \mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A}) \propto \\ p(\boldsymbol{\theta})\, p(\mathbf{Q} \mid \mathbf{x}_{1:T}, \mathbf{Q}, \boldsymbol{\theta})\, p(\mathbf{A} \mid \mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A}, \boldsymbol{\theta}). \tag{12}$$

To utilize this, we suggest to sample the hyperparameters by using a Metropolis–Hastings (MH) step, resulting in a so-called Metropolis-within-Gibbs procedure.

4 THEORETICAL RESULTS

Our model (5) and learning Algorithm 1 inherits certain well-defined properties from the reduced-rank approximation and the presented sampling scheme. In the first theorem, we consider the convergence of a series expansion approximation to the GP-SSM with an increasing number m of basis functions. As in Solin and Särkkä (2014), we only provide the convergence results for a rectangular domain with Dirichlet boundary conditions, but the result could easily be extended to a more general case. Proofs for all theorems are included in the supplementary material.

Theorem 4.1. *The probabilistic model implied by the dynamic and measurement models of the approximate GP-SSM convergences in distribution to the exact GP-SSM, when the size of the domain Ω and the number of basis functions m tends to infinity.*

The above theorem means that in the limit any probabilistic inference in the approximate model will be equivalent to inference on the exact model, because the prior and likelihood models become equivalent. The benefit of considering the m-rank model instead of a standard GP, is the following:

Theorem 4.2. *Provided the rank-reduced approximation, the computational load scales as $\mathcal{O}(m^2 T)$ as opposed to $\mathcal{O}(T^3)$.*

Furthermore, the proposed learning procedure enjoys sound theoretical properties:

Theorem 4.3. *Assume that the support of the proposal in the MH algorithm covers the support of the posterior $p(\boldsymbol{\theta} \mid \mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A}, \mathbf{y}_{1:T})$, and $N \geq 2$ in Algorithm 2. Then the invariant distribution of Algorithm 1 is $p(\mathbf{x}_{1:T}, \mathbf{Q}, \mathbf{A}, \boldsymbol{\theta} \mid \mathbf{y}_{1:T})$.*

Hence, Theorem 4.3 guarantees that our learning procedure indeed is sampling from the distribution we expect it to, even when a *finite* number of particles $N \geq 2$ is used in the Monte Carlo based Algorithm 2. It is also possible to prove uniform ergodicity for Algorithm 1, as such a result exists for Algorithm 2 (Lindsten et al., 2014).

5 NUMERICAL EXAMPLES

In this section, we will demonstrate and evaluate our contribution, the model (5) and the associated learning Algorithm 1, using four numerical examples. We will demonstrate and evaluate the proposed method (including the convergence of the learning algorithm) on two synthetic examples and two real-world datasets, as well as making a comparison with other methods.

In all examples, we separate the data set into training data $\mathbf{y}^{\mathrm{t}}$ and evaluation data $\mathbf{y}^{\mathrm{e}}$. To evaluate the performance quantitatively, we compare the estimated data $\widehat{\mathbf{y}}$ to the true data $\mathbf{y}^{\mathrm{e}}$ using the root mean square error (RMSE) and the mean log likelihood (LL):

$$\text{RMSE} = \sqrt{\frac{1}{T_{\mathrm{e}}} \sum_{t=1}^{T_e} |\widehat{\mathbf{y}}_t - \mathbf{y}_t^{\mathrm{e}}|^2} \tag{13}$$

and

$$\text{LL} = \frac{1}{T_{\mathrm{e}}} \sum_{t=1}^{T_e} \log \mathcal{N}(\mathbf{y}_t^{\mathrm{e}} \mid \mathbb{E}[\widehat{\mathbf{y}}_t], \mathbb{V}[\widehat{\mathbf{y}}_t]). \tag{14}$$

The source code for all examples is available via the first authors homepage.

Table 1: Results for synthetic and real-data numerical examples.

Data / Method	RMSE	LL	Train time [min]	Test time [s]	Comments
Synthetic data:					
PMCMC GP-SSM Frigola et al. (2013)	**1.12**	−1.57	547	420	As reported by Frigola et al. (2014b).
Variational GP-SSM Frigola et al. (2014b)	1.15	−1.61	2.1	**0.14**	As reported by Frigola et al. (2014b).
Reduced-rank GP-SSM	**1.10**	**−1.52**	**0.7**	**0.18**	Average over 10 runs.
Damper modeling:					
Linear OE model (4th order)	27.1	N/A			
Hammerstein–Wiener (4th order)	27.0	N/A			
NARX (3rd order, wavelet network)	24.5	N/A			
NARX (3rd order, Tree partition)	19.3	N/A			
NARX (3rd order, sigmoid network)	**8.24**	N/A			
Reduced-rank GP-SSM	**8.17**	**−3.71**			
Energy forecasting:					
Static GP	27.7	−2.54			
Reduced-rank GP-SSM	**21.8**	**−2.41**			

5.1 Synthetic Data

As a proof-of-concept already presented in Figure 1, we have $T = 500$ data points from the model

$$x_{t+1} = \tanh(2x_t) + w_t, \qquad y_t = x_t + e_t, \qquad (15)$$

where $e_t \sim \mathcal{N}(0, 0.1)$ and $w_t \sim \mathcal{N}(0, 0.1)$. We inferred f and Q, using a GP with the exponentiated quadratic (squared exponential, parametrized as in Rasmussen and Williams 2006) covariance function with unknown hyperparameters, and $Q \sim \mathcal{IW}(10, 1)$ as priors. In this one-dimensional case ($x \in [-L, L], L = 4$), the eigenvalues and eigenfunctions are $\lambda_j = (\pi j/(2L))^2$ and $\phi^{(j)}(x) = 1/\sqrt{L} \sin(\pi j(x + L)/(2L))$. The spectral density corresponding to the covariance function is $S_\theta(\omega) = \sigma^2 \sqrt{2\pi\ell^2} \exp(-\omega^2\ell^2/2)$.

The posterior estimate of the learned model is shown in Figure 1, together with the samples of the basis function weights $f^{(j)}$. The variance of the posterior distribution of f increases in the regimes where the data is not exciting the model.

As a second example, we repeat the numerical benchmark example on synthetic data from Frigola et al. (2014b): A one-dimensional state space model $x_{t+1} = x_t + 1 + w_t$, if $x_t < 4$, and $x_{t+1} = -4x_t + 21$, if $x_t \geq 4$ with known measurement equation $y_t = x_t + e_t$, and noise distributed as $w_t \sim \mathcal{N}(0, 1)$ and $e_t \sim \mathcal{N}(0, 1)$. The model is learned from $T = 500$ data points, and evaluated with $T_\mathrm{e} = 10^5$ data points. As in Frigola et al. (2014b), a Matérn covariance function is used (see, *e.g.*, Section 4.2.1 of Rasmussen and Williams 2006 for details, including its spectral density). The results for our model with $K = 200$ MCMC iterations and $m = 20$ basis functions are provided in Table 1.

We also re-state two results from Frigola et al. (2014b): The GP-SSM method by Frigola et al. (2013) (which also uses particle MCMC for learning) and the variational GP-SSM by Frigola et al. (2014b). Due to the compact writing in Frigola et al. (2013, 2014b), we have not been able to reproduce the results, but to make the comparison as fair as possible, we average our results over 10 runs (with different realizations of the training data). Our method was evaluated using the provided Matlab implementation on a standard desktop computer[1].

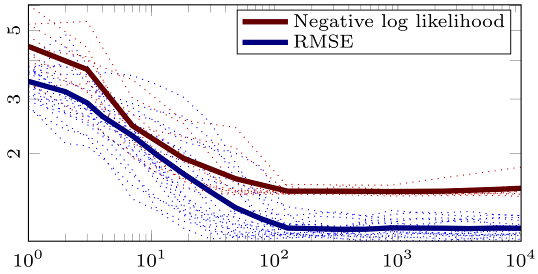


Figure 2: The (negative) log likelihood and RMSE for the second synthetic example, as a function of number of MCMC samples K, averaged (solid lines) over 10 runs (dotted lines).

The choice to use only $K = 200$ iterations of the learning algorithm is motivated by Figure 2, illustrating the 'model quality' (in terms of log likelihood and RMSE) as a function of K: It is clear from Figure 2 that the model quality is of the same magnitude after a few hundred samples and after 10 000 samples. As we know the sampler converges to the right distribution in the limit $K \to \infty$, this indicates that the sampler converges already after a few hundred samples for this example. This is most likely thanks to the linear-in-parameter structure of the reduced-rank GP, allowing for the efficient Gibbs updates (10–11).

There is an advantage for our proposed reduced-rank GP-SSM in terms of LL, but considering the stochastic elements involved in the experiment, the different RMSE performance results are hardly outside the error bounds. Regarding the computational load, however, there is a substantial advantage for our proposed method, enjoying a training time less only a third of the one by the variational GP-SSM, which in turn outperforms the method by Frigola et al. (2013).

[1] Intel i7-4600 2.1 GHz CPU.

5.2 Nonlinear Modeling of a Magneto-Rheological Fluid Damper

We also compare our proposed method to state-of-the-art conventional system identification methods (Ljung, 1999). The problem is the modeling of input–output behavior of a magneto-rheological fluid damper, introduced by Wang et al. (2009) and used as a case study in the System Identification Toolbox for Mathworks Matlab (The MathWorks, Inc., 2015). The data consists of 3 499 data points, of which 2 000 are used for training and the remaining for evaluation, shown in Figure 3a. The data exhibits some non-trivial dynamics, and as the $T = 2\,000$ data points probably not contain enough information to determine the system uniquely, a certain amount of uncertainty is present in the posterior. This is thus an interesting and realistic problem for a Bayesian method, as it possibly can provide useful information about the posterior uncertainty, not captured in classical maximum likelihood methods for system identification.

We learn a three-dimensional model:

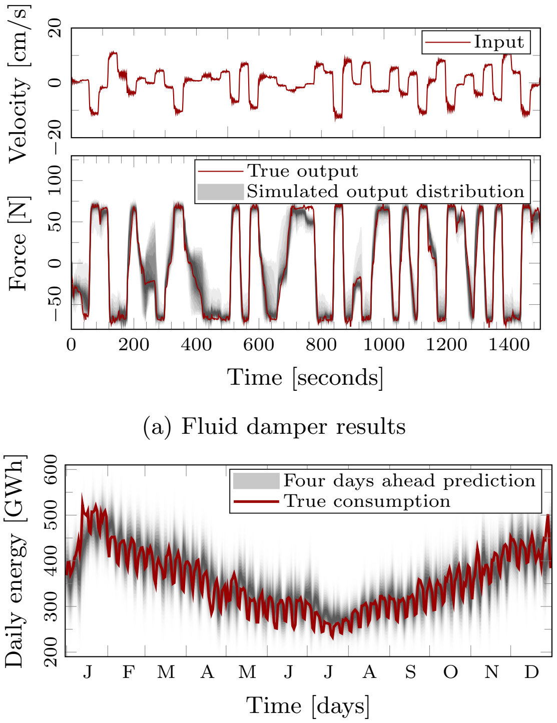
$$\mathbf{x}_{t+1} = \mathbf{f}_x(\mathbf{x}_t) + \mathbf{f}_u(u_t) + \mathbf{w}_t, \tag{16a}$$

$$y_t = [0\ 0\ 1]\mathbf{x}_t + e_t \tag{16b}$$

where $\mathbf{x}_t \in \mathbb{R}^3$, $e_t \sim \mathcal{N}(0, 5)$, and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ with $\mathbf{Q}$ unknown. We assume a GP prior with an exponentiated quadratic covariance function, with separate length-scales for each dimension. We use $m = 7^3 = 343$ basis functions[2] for $\mathbf{f}_x$ and 8 for $\mathbf{f}_u$, which in total gives 1 037 basis function weights $f^{(j)}$ and 5 hyperparameters $\boldsymbol{\theta}$ to sample.

The learned model was used to simulate a distribution of the output for the test data, plotted in Figure 3a. Note how the variance of the prediction changes in different regimes of the plot, quantifying the uncertainty in the posterior belief. The resulting output is also evaluated quantitatively in Table 1, together with five state-of-the-art maximum likelihood methods, and our proposed method performs on par with the best of these. The learning algorithm took about two hours to run on a standard desktop computer.

The assumed model with known linear g and additive form $\mathbf{f}_x + \mathbf{f}_u$ could be replaced by an even more general structure, but this choice seems to give a sensible trade-off between structure (reducing computational load) and flexibility (increasing computational load) for this particular problem. Our proposed Bayesian method does indeed appear as a realistic alternative to the maximum likelihood methods, without any more problem-specific tailoring than the rather natural model assumption (16a).



(a) Fluid damper results

(b) Electricity consumption example

Figure 3: Data (red) and predicted distributions (gray) for the real-data examples. It is interesting to note how the variance in the prediction changes between different regimes in the plots.

5.3 Energy Consumption Forecasting

As a fourth example, we consider the problem of forecasting the daily energy consumption in Sweden [3] four days in advance. The daily data from 2013 was used for training, and the data from 2014 for evaluation. The time-series was modeled as an autonomous dynamical system (driven only by noise), and a three dimensional reduced-rank GP-SSM was trained for this, with all functions and parameters unknown. To obtain the forecast, the model was used inside a particle filter to find the state distribution, and the four step ahead prediction density was computed. The data and the predictions are shown in Figure 3b.

As a sanity check, we compare to a standard GP, not explicitly accounting for any dynamics in the time-series. The standard GP was trained to the mapping from y_t to y_{t+4}, and the performance is evaluated in Table 1. From Table 1, the gain of encoding dynamical behavior in the model is clear.

[2] Explicit expression for the basis functions in the multidimensional case is found in the supplementary material.

[3] Data from Svenska Kraftnät, available: http://www.svk.se/aktorsportalen/elmarknad/statistik/.

6 DISCUSSION

6.1 Tuning

For a successful application of the proposed algorithm, there are a few algorithm-specific parameters for the user to choose: The number of basis functions m and the number of particles N in PGAS. A large number of basis functions m makes the model more flexible and the reduced-rank approximation 'closer' to a non-approximated GP, but it also increases the computational load. With a smooth covariance function κ, the prior is in practice $f^{(j)} \approx 0$ for moderate j, and m can be chosen fairly small (as a rule of thumb, say, 6–15 per dimension) without making a too crude approximation. In our experience, the number of particles N in PGAS can be chooses fairly small (say, 20), without affecting the mixing properties of the Markov chain heavily. This is in accordance to what has been reported in the literature by Lindsten et al. (2014).

6.2 Properties of the Proposed Model

We have proposed to use the reduced-rank approximation of GPs by Solin and Särkkä (2014) within a state space model, to obtain a GP-SSM which efficiently can be learned using a PMCMC algorithm. As discussed in Section 3 and studied using numerical examples in Section 5, the linear-in-the-parameter structure of the reduced-rank GP-SSM allows for a computationally efficient learning algorithm. However, the question if a similar performance could be obtained with another GP approximation method or another learning scheme arises naturally.

Other GP approximation methods, for example pseudo-inputs, would most likely not allow for such efficient learning as the reduced-rank approximation does; unless closed-form Gibbs updates are available (requiring a linear-in-the-parameter structure, or similar), the parameter learning would have to resort to Metropolis–Hastings, which most likely would give a significantly slower learning procedure. For many GP approximation methods it is also more natural to find a point estimate of the parameters (the inducing points, for example) using, for example, EM, rather than inferring the parameter posterior, as is the case in this paper.

The learning algorithm, on the other hand, could probably be replaced by some other method also inferring (at least approximately) the posterior distribution of the parameters, such as SMC^2 (Chopin et al., 2013) or a variational method. However, to maintain efficiency, the method has to utilize the linear-in-the-parameter structure of the model to reach a computational load competitive with our proposed scheme. Such an alternative (however only inferring MAP estimate of the sought quantities) could possibly be the method by Kokkala et al. (2014).

6.3 Conclusions

We have proposed the reduced-rank GP-SSM (5), and provided theoretical support for convergence towards the full GP-SSM. We have also proposed a theoretically sound MCMC-based learning algorithm (including the hyperparameters) utilizing the structure of the model efficiently.

By demonstration on several examples, the computational load and the modeling capabilities of our approach have been proven to be competitive. The computational load of the learning is even less than in the variational sparse GP solution provided by Frigola et al. (2014b), and the performance in challenging input–output modeling is on par with well-established state-of-the-art maximum likelihood methods.

6.4 Possible Extensions and Further Work

A natural extension for applications where some domain knowledge is present, is to let the model include some functions with an *a priori* known parametrization. The handling of such models in the learning algorithm should be feasible, as it is already known how to use PGAS for such models (Lindsten et al., 2014). Further, the assumptions of the $\mathcal{IW}$ prior of $\mathbf{Q}$ (6) are possible to circumvent by using, for example, MH, at the cost of an increased computational load. The same holds true for the Gaussian noise assumption in (5).

Another direction for further work is to adapt the process to be able to sequentially learn and improve the model when data is added in batches, by formulating the previously learned model as the prior to the next iteration of the learning. This could probably be useful in, for example, reinforcement learning, along the lines of Deisenroth et al. (2015).

In the engineering literature, dynamical systems are mostly defined in discrete time. An interesting approach to model the continuous-time counterpart using Gaussian processes is presented by Ruttor et al. (2013). A development of the reduced-rank GP-SSM to continuous time dynamical models using stochastic Runge–Kutta methods would be of great interest for further research.

References

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, 2006.

R. D. Turner, M. P. Deisenroth, and C. E. Rasmussen. State-space inference and learning with Gaussian processes. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 868–875, 2010.

R. Frigola, F. Lindsten, T. B. Schön, and C. Rasmussen. Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems*, volume 26, pages 3156–3164, 2013.

R. Frigola, F. Lindsten, T. B. Schön, and C. Rasmussen. Identification of Gaussian process state-space models with particle stochastic approximation EM. In *Proceedings of the 19th IFAC World Congress*, pages 4097–4102, 2014a.

R. Frigola, Y. Chen, and C. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems*, volume 27, pages 3680–3688, 2014b.

C. L. C. Mattos, Z. Dai, A. Damianou, J. Forth, G. A. Barreto, and N. D. Lawrence. Recurrent Gaussian processes. *arXiv preprint arXiv:1511.06644*, 2016. To be presented at the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, May 2016.

A. McHutchon. *Nonlinear Modelling and Control Using Gaussian Processes.* PhD thesis, University of Cambridge, 2014.

R. Frigola-Alcade. *Bayesian Time Series Learning with Gaussian Processes.* PhD thesis, University of Cambridge, 2015.

M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

Z. Ghahramani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 431–437, 1998.

R. S. Tsay. *Analysis of Financial Time Series.* Wiley, Hoboken, NJ, 3rd edition, 2010.

J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

L. Ljung. *System Identification: Theory for the User.* Prentice Hall, Upper Saddle River, NJ, 1999.

The MathWorks, Inc. Nonlinear modeling of a magneto-rheological fluid damper. Example file provided by Matlab® R2015b System Identification Toolbox™, 2015. Available at http://mathworks.com/help/ident/examples/nonlinear-modeling-of-a-magneto-rheological-fluid-damper.html.

M. A. Alvarez, D. Luengo, and N. D. Lawrence. Latent force models. In *Proceedings of 12th International Conference on Artificial Intelligence and Statistics*, pages 9–16, 2009.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, volume 24, pages 2510–2518, 2011.

M. P. Deisenroth, R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen. Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871, 2012.

M. Deisenroth and S. Mohamed. Expectation propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 2609–2617, 2012.

A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*, 2014.

F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

A. Wills, T. B. Schön, F. Lindsten, and B. Ninness. Estimation of linear systems using a Gibbs sampler. In *Proceedings of the 16th IFAC Symposium on System Identification*, pages 203–208, 2012.

M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11(1):1865–1881, 2010.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (3):269–342, 2010.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *Nonlinear Filtering Handbook*, pages 656–704. Oxford University Press, Oxford, 2011.

J. Wang, A. Sano, T. Chen, and B. Huang. Identification of hammerstein systems without explicit parameterisation of non-linearity. *International Journal of Control*, 82(5): 937–952, 2009.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. SMC^2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

J. Kokkala, A. Solin, and S. Särkkä. Expectation maximization based parameter estimation by sigma-point and particle smoothing. In *Proceedings of 17th International Conference on Information Fusion*, pages 1–8, 2014.

A. Ruttor, P. Batz, and M. Opper. Approximate Gaussian process inference for the drift of stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 26, pages 2040–2048, 2013.

S. Särkkä and R. Piché. On convergence and accuracy of state-space approximations of squared exponential covariance functions. In *Proceedings of the International Workshop on Machine Learning for Signal Processing*, 2014.

L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, pages 1701–1728, 1994.