

Semiparametric modeling: correcting low-dimensional model error in parametric models

Tyrus Berry^{a,*}, John Harlim^{a,b}

^a*Department of Mathematics, the Pennsylvania State University, 109 McAllister Building, University Park, PA 16802-6400, USA*

^b*Department of Meteorology, the Pennsylvania State University, 503 Walker Building, University Park, PA 16802-5013, USA*

Abstract

In this paper, a semiparametric modeling approach is introduced as a paradigm for addressing model error arising from unresolved physical phenomena. Our approach compensates for model error by learning an ancillary dynamical model for the unknown parameters. Practically, the proposed approach consists of the following steps. Given a physics-based model and a noisy data set of historical observations, a Bayesian filtering algorithm is used to extract a time-series of the parameter values. Subsequently, the *diffusion forecast* algorithm is applied to the retrieved time-series in order to construct the ancillary model for the evolving parameters. The semiparametric forecasting algorithm consists of integrating the existing physics-based model with an ensemble of parameters sampled from the probability density function of the diffusion forecast. To specify initial conditions for the diffusion forecast, a Bayesian semiparametric filtering method is introduced that extends the Kalman-based filtering framework. In difficult test examples, which introduce chaotically and stochastically evolving hidden parameters into the Lorenz-96 model, we show that our approach can effectively compensate for model error, with forecasting skill comparable to that of the perfect model.

Keywords: diffusion maps, model error, diffusion forecast, semiparametric modeling, Kalman filter, nonparametric modeling

1. Introduction

A long-standing problem in modeling natural processes is model error. This issue is attributed to our incomplete understanding of the underlying physics and our lack of computational resources to resolve physical processes at various time and length scales. Serious efforts to combat this issue have been proposed with parametric modeling approaches (for examples see [1, 2, 3, 4, 5, 6, 4, 7, 8] just to name a few). In fact, it was rigorously shown in a simple context that it is possible to fully compensate for model error arising from unresolved scales with appropriate stochastic parametric models. In particular, it was shown that one can simultaneously obtain optimal filtering and climatological statistical estimates assuming that one has access to the *right* stochastic parametric model [9] which can be quite demanding in general.

While stochastic parameterization techniques have been successful in many particular problems, this approach also has another limitation beyond the selection of the parametric form. Namely, determining the parameters in these models from a limited amount of noisy data can be nontrivial, especially when the parameters are not directly observed. This was shown in [9], which found that even when the appropriate parametric form is known, the resulting forecasting skill is sensitive to the stochastic parameter estimation scheme. While a class of adaptive parameter estimation schemes [8, 10, 11] produce much more accurate results relative to the standard linear regression fitting method [12], they are numerically expensive when the dimensionality of the problems increase.

The main goal of the present paper is to introduce a modeling paradigm that avoids the two fundamental issues in parametric modeling: namely the choice of parametric form and the fitting of parameters. Our approach is philosophically motivated by the semiparametric modeling method for *time-independent* densities, which is well known in the

*Corresponding author

Email addresses: thb11@psu.edu (Tyrus Berry), jharlim@psu.edu (John Harlim)

statistics literature as a flexible work-around for overcoming the curse-of-dimensionality [13]. Classical nonparametric models such as histograms or kernel density estimates can reconstruct an unknown *time-independent* density from a training data set, however the amount of data needed grows exponentially in the dimensionality of the data [14]. Alternatively, estimating parameters of a parametric statistical model can often recover a high-dimensional density from a relatively small data set. However, imposing the rigid structure of the parametric form opens up the possibility of model error. Semiparametric models in statistics attempt to take an intermediate road by choosing a parametric form but allowing the parameters to vary and building nonparametric models for the parameters. Choosing the structure of the parametric form allows one to encode the prior knowledge about the data set, hopefully reducing the intrinsic dimensionality of the parameter space to the point where a nonparametric model can be used to capture the remaining variability. In this paper, we will extend this statistical idea of semiparametric modeling to estimate *time-dependent* densities in order to improving forecasting and filtering of dynamical systems.

In particular, the semiparametric approach introduced in the present paper will be based on a recently developed nonparametric modeling approach for learning a dynamical system from a training data set. This nonparametric approach, called the *diffusion forecast*, was introduced for stochastic gradient flow systems in [15] and has been extended to model diffusion processes on smooth manifolds [16]. Essentially, the diffusion forecast is a data-driven method for solving the Fokker-Planck equations of any autonomous diffusion processes on a smooth manifold without requiring knowledge of the corresponding drift coefficients, diffusion tensor, or even the underlying manifold itself. It was shown in [17] that diffusion forecast provides high forecasting skill, and is competitive with some physics-based parametric models for predicting energetic modes of turbulent dynamical systems using relatively small training data sets (on the order of 10^3 data points). The novel aspect of the present paper, beyond [16, 17], is to overcome the curse-of-dimensionality which restricts the diffusion forecast to modeling low dimensional dynamics. We overcome this limitation by assuming that we have an approximate or incomplete parametric model and then using the diffusion forecast to fill in the missing components, in other words, to correct the model error. By assuming that the parametric model captures most of the variability, so that the model error is low-dimensional, the nonparametric model becomes feasible. Simultaneously, from the perspective of the modeling community, the goal of semiparametric modeling is to overcome model error by using historical data to ‘correct’ the existing physical models. These two perspectives indicate that semiparametric modeling has the potential to seamlessly blend the strengths of the parametric and nonparametric modeling approaches, thereby overcoming their complementary weaknesses.

Combining the partial parametric model with the nonparametric, diffusion forecasting, model requires new semiparametric forecasting and filtering algorithms. Since the framework is introduced for correcting model error in existing parametric models, an important consideration is that we wish to maintain as much of the standard parametric ensemble forecasting and filtering framework as possible. To achieve these goals, in Section 2 we will define the form of the model error that we will be able to address here, and we briefly review some simple stochastic parametric models which have been successful for certain types of model error. We will assume that the model error can be described by dynamically varying certain parameters in the parametric model, and that the evolution of these parameters is independent of the state variables of the parametric model. In Section 3 we introduce a semiparametric forecasting algorithm which combines diffusion forecast of [16] with a standard ensemble forecast method for the parametric model. For simplicity and clarity, in Section 3 we assume that a training data set of the varying parameters is available and that we are given noisy initial conditions for forecasting. In Sections 4 and 5, we will discuss additional strategies for dealing with the more realistic scenario when we only have noisy observations of the state variables of the parametric model, which is the common situation in practice. In particular, in Section 4 we use an adaptive filtering method developed in [10, 9] to extract a time series of the varying parameters from the noisy observations. In Section 5 we introduce a semiparametric filter which combines an Ensemble Kalman Filter (EnKF) for the parametric model with the nonparametric model (learned from the time series recovered in Section 4) in order to find initial conditions from the noisy observations. As a proof-of-concept example, we will demonstrate the semiparametric filter and forecast on the Lorenz-96 model with model error arising from a parameter which evolves according to low-dimensional chaotic or stochastic models. We close this paper with a short summary in Section 6.

2. Problem Statement and Background

We consider the problem of filtering and forecasting in the presence of model error. We assume that we are given a noisy time series of observations from a known dynamical model,

$$\dot{x} = f(x, \theta), \quad (1)$$

with an observation function,

$$y = h(x, \theta), \quad (2)$$

that depends on parameters θ which evolve according to an unknown stochastic model,

$$\dot{\theta} = g(\theta, \dot{W}), \quad (3)$$

where \dot{W} denotes a white noise process. We assume that the evolution of θ is ergodic on a physical parameter domain (or a Riemannian manifold) \mathcal{M} which may be a subset of the parameter space, $\theta \in \mathcal{M} \subset \mathbb{R}^n$. Moreover, we assume that the distribution of θ evolves according to a Fokker-Planck equation with a unique equilibrium density $p_{\text{eq}}(\theta)$. A key assumption in this paper is that the evolution of the parameters, θ , does not depend on the state x . This says that the model error is homogeneous with respect to the state space, which is a restrictive assumption for practical applications. The key difficulty in solving the *conditional* or *heterogeneous* problem is that if g was allowed to vary with x there would typically not be enough data at each state x to fit a nonparametric model; we discuss this further in Section 6.

With the above problem statement in mind, we will first consider an idealized scenario where we have a training time series of the parameters θ . In this first scenario we will also bypass the filtering step and assume that we are given an initial state $(x(t), \theta(t))$ at some time t from which we wish to forecast the state into the future. Without knowing the model in (3), our goal is to first use the training time series to learn an approximate model $\tilde{g} \approx g$ for θ . Subsequently, we combine this approximate model \tilde{g} for θ with the parametric model (1) in order to forecast the initial condition $(x(t), \theta(t))$. In the remainder of this section we will review several standard methods for building approximate models, \tilde{g} .

Given a discrete time series of historical values of the parameters $\{\theta_i = \theta(t_i)\}_{i=1}^N$ there are several standard approaches for building a forecast model for (x, θ) . The simplest method is to simply ignore the training data, and hold the value of θ constant at the initial parameter value $\theta(t)$, so that $\tilde{g} = 0$, and we will refer to this method as the *persistence model*. The persistence model integrates the system (1) with $\theta(t + \tau) = \theta(t)$ held constant at the initial value $\theta(t)$, in order to obtain the forecast of x . When the evolution of θ is deterministic and very slow relative to x , the persistence model can give reasonable short term forecasts, assuming that the system is still stable with θ fixed. Of course, if the evolution of θ is very fast, the persistence model will quickly diverge from the true state. Moreover, the persistence model will typically not capture the correct long term mean of the true system, leading to a very biased long term forecast which can easily be worse than simply predicting the mean value of x .

When the evolution of θ is very fast, we can assume that, relative to x , the parameters θ quickly equilibrate to their climatological distribution. Assuming that the evolution of θ is ergodic and that we have a sufficiently long training data set $\{\theta_i\}$, we can emulate the equilibrium density by simply drawing random parameters from the historical data. To use this in the forecast, for each integration step of the forecast model (1) we draw a random value of θ from the historical data; this strategy is similar to the *heterogeneous multiscale method (HMM)* framework [2]. For the long term forecast, the HMM gives an unbiased forecast model which recovers the correct equilibrium density, however for the short term forecast, the HMM completely ignores the known initial parameter value $\theta(t)$.

So far, the persistence model assumes the evolution of the parameters is slow and ignores the training data, whereas the HMM assumes the evolution of the parameters is fast and ignores the initial state $\theta(t)$. It turns out that blending these two approaches is quite challenging. A method which attempts to blend these, in the sense that it moves the initial state $\theta(t)$ toward the long term mean $\hat{\theta} = \frac{1}{N} \sum_{i=1}^N \theta_i$ at an appropriate speed, was introduced as part of the MTV (named after Majda, Timofeyev, and Vanden-Eijnden) strategy for reduced climate modeling [18]. In [18] they approximate the interaction between the unresolved scales with a parametric model involving a linear damping term and white noise stochastic forcing. Subsequently, they construct a closure model for $x(t)$ by employing the classical homogenization method, assuming a time scale separation between x and θ [19].

Following just the approximation step in the MTV, since g is assumed to be independent of x , we simply have a linear stochastic model,

$$\dot{\theta} = \tilde{g}(\theta, \dot{W}) = \alpha(\hat{\theta} - \theta) + \sigma\dot{W}, \quad (4)$$

where $\alpha = 1/T_\theta$ is determined by the correlation time T_θ of the historical data and the stochastic forcing $\sigma = \sqrt{2\alpha\text{var}(\theta)}$ is determined by the variance of the historical data. This modeling approach is known as the *mean stochastic model (MSM)* (see [20]). Assuming that the initial density for the parameters is Gaussian with mean given by the initial state $\theta(t)$ and an initial variance $s(t)$ (possibly coming from a Kalman filter), the density of the MSM forecast at time $t + \tau$ is a Gaussian and its mean and variance can be determined explicitly. Using the analytic solution for the linear (and Gaussian) model in (4), we can generate random samples from the Gaussian forecast at each integration step of the ensemble forecast. While the MSM effectively uses the initial state, it only captures two moments (namely the mean and covariance) and the two-time correlation statistics of the climatological distribution.

In our numerical experiments below, we will compare the forecasting skill of the semiparametric model with the three approaches mentioned above, namely the persistence, HMM, and MSM models, in addition to the perfect model. There are of course other more sophisticated parametric stochastic models, and we should point out that a comparison with these methods was already made in the context of nonparametric forecasting of turbulent modes in [17]. In this paper we assume that all of the physical knowledge has already been incorporated into the parametric model, so that we have absolutely no knowledge of the system (3) which could be used to suggest a better parametric form.

3. Semiparametric Forecasting

We will assume that the model for x is nonlinear and potentially high-dimensional and chaotic so that the most practical method of forecasting x is simply to integrate an ensemble of initial states forward in time using the model (1). Semiparametric forecasting consists of using the training data to learn a nonparametric model for θ and combining this model with the dynamical model (1) to improve the forecast of an initial state $(x(t), \theta(t))$.

To simplify the discussion in this section, we assume that we have a training data set consisting of a historical time series $\{\theta_i = \theta(t_i)\}$ of the parameters and we are given an initial state $x(t)$ and the initial parameter values $\theta(t)$ at some time t , from which we would like to forecast the state x into the future. We will consider the more realistic situation, where the training data set and initial conditions for θ is not available, in Sections 4 and 5. In the remainder of this section, under the current idealized scenario, we introduce the semiparametric forecasting approach. In Section 3.1 we briefly review the nonparametric modeling approach, namely the diffusion forecast [16] for θ . In Section 3.2 we show how to combine this nonparametric model with the known parametric model, f , to perform a semiparametric forecast. Finally, in Section 3.3 we demonstrate the semiparametric forecast on an example based on a Lorenz-96 model, where we introduce a complex model error which is governed by the Lorenz-63 model.

3.1. Step 1: Forecasting parameters with the nonparametric model

We assume that the true dynamics of the parameters $\theta(t)$ are constrained to a physical parameter domain $\mathcal{M} \subset \mathbb{R}^n$, and evolve according to a (3). Notice that we allow the data to be constrained to a domain \mathcal{M} inside the parameter space \mathbb{R}^n , so even when the dimension, n , of the parameter space is high, the data requirements will only depend on the dimension of manifold \mathcal{M} . We emphasize that the manifold \mathcal{M} , as well as the governing equations, g , are unknown and will be learned implicitly from the training data set. Moreover, the parameters θ may include variables which do not appear explicitly in the model (1), but we assume that our training data set only consists of observable variables which appear explicitly in (1). In order to compensate for unobserved variables in the dynamics of the parameters θ which appear in (1), we will apply a time-delay embedding to the available variables [21, 22, 23, 24, 25, 26, 17].

Given a time series $\theta_i = \theta(t_i)$ sampled at discrete times $\{t_i\}_{i=1}^N$ we are interested in constructing a forecasting model so that given an initial density $p(\theta, t)$ at time t we can estimate the density $p(\theta, t + \tau)$ at time $t + \tau$, where $\tau > 0$. We assume that θ_i is already the delay coordinate embedding of the available parameters which appear explicitly in the parametric model (1). Probabilistically, the evolution of the density, p , in the forecasting problem is governed by the Fokker-Planck equation,

$$\frac{\partial p}{\partial t} = \mathcal{L}^* p, \quad p(\theta, t) = p_t(\theta), \quad (5)$$

where \mathcal{L}^* is the Fokker-Planck operator, which we assume exists for the system (3). We note that the equilibrium distribution, $p_{\text{eq}}(\theta)$, of the underlying dynamics (3) satisfies, $\mathcal{L}^* p_{\text{eq}} = 0$.

The nonparametric forecasting algorithm introduced in [16] is called the *diffusion forecast*. The central idea of the diffusion forecast is to project the forecasting problem (5) onto a basis of smooth real-valued functions $\varphi_j(\theta) \in L^2(\mathcal{M}, p_{\text{eq}})$. Here, $L^2(\mathcal{M}, p_{\text{eq}})$ is the space functions which are square-integrable with respect to p_{eq} , which is a Hilbert space with inner product,

$$\langle f, g \rangle_{p_{\text{eq}}} = \int_{\mathcal{M}} f(\theta)g(\theta)p_{\text{eq}}(\theta) dV(\theta)$$

where $dV(\theta)$ is the volume form which \mathcal{M} inherits from the ambient space $\mathbb{R}^n \ni \theta$. We will discuss below how the basis $\{\varphi_j\}$ will be estimated using the training data; for now we assume that the basis functions are known.

In order to find the forecast density $p(\theta, t + \tau)$ from the initial density $p(\theta, t)$, we first project this function onto the basis $\{\varphi_j\}$ by computing the projection coefficients $c_j(t) = \langle p(\cdot, t), \varphi_j \rangle$. We can then forecast the coefficients $\vec{c}(t)$ with a linear map $\vec{c}(t + \tau) = A\vec{c}(t)$, whose components are defined by $A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\text{eq}}}]$, where S is the discrete *shift map* defined by $S\varphi_l(\theta_i) = \varphi_l(\theta_{i+1})$. It was shown in [16] that the linear map A is exactly the semigroup solution of the backward Kolmogorov equation of (3) represented in the basis $\{\varphi_j\}$, whose components are,

$$A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\text{eq}}}] = \langle \varphi_j, e^{\tau\mathcal{L}}\varphi_l \rangle_{p_{\text{eq}}}.$$

The time step, τ , of the forecasting operator is determined by the sampling time $\tau \equiv t_{i+1} - t_i$ of the training data. Finally, we can reconstruct the forecast density by,

$$p(\theta, t + \tau) = \sum_j c_j(t + \tau)p_{\text{eq}}(\theta)\varphi_j(\theta).$$

These formula were all derived in [16], and we now summarize the nonparametric forecasting approach with the following diagram,

$$\begin{array}{ccc} p(\theta, t) & \xrightarrow{\text{Nonparametric Forecast}} & p(\theta, t + \tau) \\ \downarrow \langle p, \varphi_j \rangle & & \uparrow \sum_j c_j \varphi_j p_{\text{eq}} \\ \vec{c}(t) & \xrightarrow{A_{lj} = \langle \varphi_j, e^{\tau\mathcal{L}}\varphi_l \rangle_{p_{\text{eq}}}} & \vec{c}(t + \tau) = A\vec{c}(t). \end{array}$$

In the above diagram, the nonparametric forecasting algorithm follows the solid arrows: projecting onto the basis, applying the linear map A (which will be approximated by \hat{A} as defined below), and then reconstructing the forecast density. Given a basis $\{\varphi_j\}$ and an estimate of the equilibrium density p_{eq} , it was shown in [16] that the above inner products can be estimated as Monte-Carlo integrals on the training data set $\{\theta_i\}$ given by,

$$c_j(t) \equiv \langle p(\cdot, t), \varphi_j \rangle \approx \frac{1}{N} \sum_{i=1}^N p(\theta_i, t)\varphi_j(\theta_i)p_{\text{eq}}(\theta_i)^{-1}, \quad (6)$$

$$A_{lj} \approx \hat{A}_{lj} \equiv \langle \varphi_j, S\varphi_l \rangle_{p_{\text{eq}}} \approx \frac{1}{N-1} \sum_{i=1}^{N-1} \varphi_j(\theta_i)\varphi_l(\theta_{i+1}). \quad (7)$$

Notice that we only need estimates of the basis $\{\varphi_j\}$ and the density p_{eq} evaluated on the training data set. Of course, in practice we will truncate the projections onto a finite number M of the basis functions, φ_j , so that we implicitly project the forecast density into the span of the first M basis functions, $\{\varphi_j\}_{j=1}^M$. Notice that by projecting and reconstructing the density p in the basis $\{\varphi_j\}$ (see the diagram above), we have projected the density in the ambient space \mathbb{R}^n onto the manifold \mathcal{M} , as determined by the historical data set. In fact, $\{\varphi_j\}$ is a generalized Fourier basis and since we only use M basis elements, it is possible to have Gibbs phenomenon oscillations in the reconstructed density, which can lead to negative values. In order to avoid these negative values, whenever we reconstruct the density we will always take the max of the reconstructed values and zero, and then renormalize the density by dividing by,

$$Z = \int_{\mathcal{M}} p(\theta, t)dV(\theta) \approx \frac{1}{N} \sum_{i=1}^N p(\theta_i, t)p_{\text{eq}}(\theta_i)^{-1}, \quad (8)$$

which estimates the normalization factor. In (8), dV denotes the volume form inherited by the manifold \mathcal{M} from the ambient space.

The final piece of the nonparametric forecasting algorithm is to choose an appropriate basis $\{\varphi_j\}$. It was shown in [16] that the optimal basis for minimizing the error between the estimate \hat{A} of the matrix A are the eigenfunctions of an elliptic operator $\hat{\mathcal{L}}$, which turns out to be the generator of a stochastic gradient flow,

$$d\theta = \nabla \log p_{\text{eq}}(\theta) dt + \sqrt{2} dW, \quad (9)$$

whose equilibrium density is exactly $p_{\text{eq}}(\theta)$, the equilibrium density of (3). By choosing this basis of eigenfunctions, it was shown in [16] that the difference between the estimate \hat{A}_{lj} and the true coefficient A_{lj} is an error of order τ/N in probability. Most importantly, these eigenfunctions, $\{\varphi_j\}$, and the density p_{eq} , can be estimated with the diffusion maps algorithm [27, 28].

Intuitively, the diffusion maps algorithm uses the data set $\{\theta_i\}_{i=1}^N$, to approximate the operator $\hat{\mathcal{L}}$, the generator of (9). For any function $F : \mathcal{M} \rightarrow \mathbb{R}$ we can represent $F \in L^2(\mathcal{M}, p_{\text{eq}})$ on the data set $\{\theta_i\}$ by an $N \times 1$ vector $\vec{F}_i = F(\theta_i)$. The operator $\hat{\mathcal{L}}$ is then approximated by an $N \times N$ matrix L , in the sense that the matrix-vector product $L\vec{F}$ has entries $(L\vec{F})_i \approx (\hat{\mathcal{L}}F)(\theta_i)$ so that the vector $L\vec{F}$ approximates the function $\hat{\mathcal{L}}F$ on the data set. The algorithm for constructing the matrix L , and the theory proving its convergence to $\hat{\mathcal{L}}$ in the limit of large data, were established in [28] for sparsely sampled data; generalizing the seminal result of [27]. The eigenvectors of the matrix L approximate the desired eigenfunctions evaluated on the training data set, as required in (6) and (7). The diffusion maps algorithm simultaneously provides a kernel density estimate of p_{eq} evaluated on the training data set as required in (6) and (8). As a result, the reconstructed forecast density, $p(\theta, t + \tau)$, is represented by its values on the training data set, $p(\theta_i, t + \tau)$. In this article, we use the algorithm of [28] since the equilibrium density, p_{eq} , may take values arbitrarily close to zero, which would imply regions of sparse sampling on \mathcal{M} . For a brief overview of the theory of [28], see the Supplementary Material of [16], and for a simple guide to the algorithm see the appendix of [17].

3.2. Step 2: Combining the nonparametric and parametric forecast models

In this section, we introduce the semiparametric forecasting scheme which combines the parametric model (1) for x and the nonparametric model of Section 3.1 for θ . We assume that we are given an ensemble of K initial conditions $\{x^k(t)\}_{k=1}^K$ which represent the distribution of the variable x at the initial time t , and an initial density $p(\theta, t)$ for the parameters, and our goal is to forecast these two quantities to the future time $t + \tau$.

We first describe how to forecast the ensemble $\{x^k\}$. Since the model for x is parametric of the form $f(x, \theta)$, the evolution of x depends on the parameters θ . In order to forecast each ensemble member, x^k , with the parametric model, we need to produce samples $\{\theta^k\}_{k=0}^K$ of the density $p(\theta, t)$. While various sampling methods can be used, in this paper we will sample the density $p(\theta, t)$ using the standard rejection method, where the reference density is the equilibrium density p_{eq} , which is also the sampling measure by our ergodicity assumption on θ . To perform the rejection sampling, we set $P = \max_i \{p(\theta_i, t)/p_{\text{eq}}(\theta_i)\}$ to be the upper bound of the ratio between the two distributions. Notice that since both densities are known on the historical training data set $\{\theta_i\}$ we can easily compute this optimal upper bound. Since the historical training data points $\{\theta_i\}$ are samples of p_{eq} , we can easily produce a random sample of the density p_{eq} by randomly choosing a training data point θ_m where $m \in \{1, \dots, N\}$ is a uniformly random index. We then choose a uniform random number $\xi \in [0, 1]$ and we accept θ_m as a sample of $p(\theta, t)$ if $\xi < p(\theta_m, t)/(P p_{\text{eq}}(\theta_m))$ and otherwise we reject θ_m and redraw until we accept. We repeat this rejection sampling until we have K samples $\{\theta^k(t)\}_{k=1}^K$ from the density $p(\theta, t)$. Each ensemble member $\{x^k(t)\}$ is then paired with the corresponding sample from $p(\theta, t)$ to form the combined ensemble $\{(x^k(t), \theta^k(t))\}$. We can now integrate the parametric model $\dot{x} = f(x, \theta)$, using each ensemble state $(x^k(t), \theta^k(t))$, for time τ in order to find the forecast ensemble $\{x^k(t + \tau)\}$ at time $t + \tau$.

The second part of the semiparametric forecasting algorithm is exactly the diffusion forecasting scheme described in Section 3.1. To repeat, we project $p(\theta, t)$ onto the basis $\{\varphi_j\}$ to find the coefficients $c_j(t)$ as in (6). We then forecast the density on the parameters using the matrix \hat{A} from (7) so that $c(t + \tau) = \hat{A}c(t)$ and we can then reconstruct the density at time $t + \tau$. Subsequently, we sample $p(\theta, t + \tau)$ using the same rejection sampling procedure to obtain an ensemble $\{\theta^k(t + \tau)\}$ which we then pair with $\{x^k(t + \tau)\}$ to form $\{(x^k(t + \tau), \theta^k(t + \tau))\}$ and again integrate to find $\{x^k(t + 2\tau)\}$. We can then repeat the procedure by finding $c(t + 2\tau) = \hat{A}c(t + \tau) = \hat{A}^2 c(t)$, reconstructing the density $p(\theta, t + 2\tau)$ and so on for as long as we would like to forecast. Therefore, in addition to the nonparametric forecasting

described before, the *semiparametric forecasting algorithm* consists of sampling the nonparametric forecast density at each forecast step and using these samples for integrating an ensemble forecast of the parametric model.

We summarize the semiparametric forecasting algorithm with the following diagram,

$$\begin{array}{ccc}
 (x^k(t), \theta^k(t)) & \xrightarrow{\hat{x}=f(x,\theta)} & (x^k(t+\tau), \theta^k(t+\tau)) \\
 \uparrow \theta^k(t) & & \uparrow \theta^k(t+\tau) \\
 p(\theta, t) & \xrightarrow{\text{Nonparametric Forecast}} & p(\theta, t+\tau) \\
 \downarrow \langle p, \varphi_j \rangle & & \uparrow \sum_j c_j \varphi_j p_{\text{eq}} \\
 \vec{c}(t) & \xrightarrow{A_{ij}=\langle \varphi_j, e^{\tau L} \varphi_i \rangle_{p_{\text{eq}}}} & \vec{c}(t+\tau) = A \vec{c}(t).
 \end{array}$$

Notice that the nonparametric model iterates independently from the parametric model, which follows from the assumption that the model (3) for θ is independent of the state variable x . Furthermore, the dashed line is not taken directly but is a summary of the path of solid lines which uses the projection onto the basis $\{\varphi_j\}$. The parametric model is used to forecast the ensemble of states $x^k(t)$ and at each step the ensemble of parameters $\theta^k(t)$ are updated by sampling the density $p(\theta, t)$ from the nonparametric forecast. As in Section 3.1 the initial density $p(\theta, t)$, which can be arbitrary, will be evaluated on the data set, and all subsequent densities $p(\theta, t + m\tau)$ are represented by the values $p(\theta_i, t + m\tau)$.

We should note that the ensemble members $\theta^k(t + m\tau)$ are not actually time series solutions of (3), since at each forecast step we choose a completely new set of random samples independently from the samples of the previous step. Producing a consistent time series for each ensemble forecast $\theta^k(t + m\tau)$ would require sampling the conditional density at each time given the sample at the previous time. This would allow more complex integration schemes to be considered for combining the samples $\{\theta^k\}$ with the ensemble $\{x^k\}$, however, for a small ensemble size it would result in a more biased forecast.

3.3. Application of semiparametric modeling to a chaotic system with model error

In this example we will introduce model error into the 40-dimensional Lorenz-96 [29] system,

$$\dot{x}_i = x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8, \quad (10)$$

where the indices are taken modulo 40, which is known to have chaotic dynamics. Notice that the system is dissipative in the sense that the evolution of the energy-like quantity, $S \equiv \sum_i x_i^2/40$ is simply $\dot{S} = -2S + 2/5 \sum_i x_i$. We will now introduce a model error by assuming that the evolution of the true system is governed by,

$$\begin{aligned}
 \dot{x}_i &= f(x_i, \theta) = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8, \\
 \dot{\theta} &= (a_1/40 + 1), \\
 \dot{a}_1 &= 10(a_2 - a_1)/\epsilon, \\
 \dot{a}_2 &= (28a_1 - a_2 - a_1a_3)/\epsilon, \\
 \dot{a}_3 &= (a_1a_2 - 8a_3/3)/\epsilon,
 \end{aligned} \quad (11)$$

which we call the L96-L63 system, where we replace the constant coefficient in the first quadratic term of (10) with a parameter θ . The parameter θ is an appropriate rescaling of the first component, a_1 , of the chaotic three-dimensional Lorenz-63 [30] system. The system (11) does not have the dissipative property of (10) but is empirically stable in long numerical simulations. Empirically, we found that the system (11) became very unstable if θ was allowed outside the interval $[0.5, 1.5]$. The constant ϵ is included to allow differences in the time-scale between the parametric model for x and the parameters θ . In this section we will use $\epsilon = 1$ however we will consider faster ($\epsilon < 1$) and slower

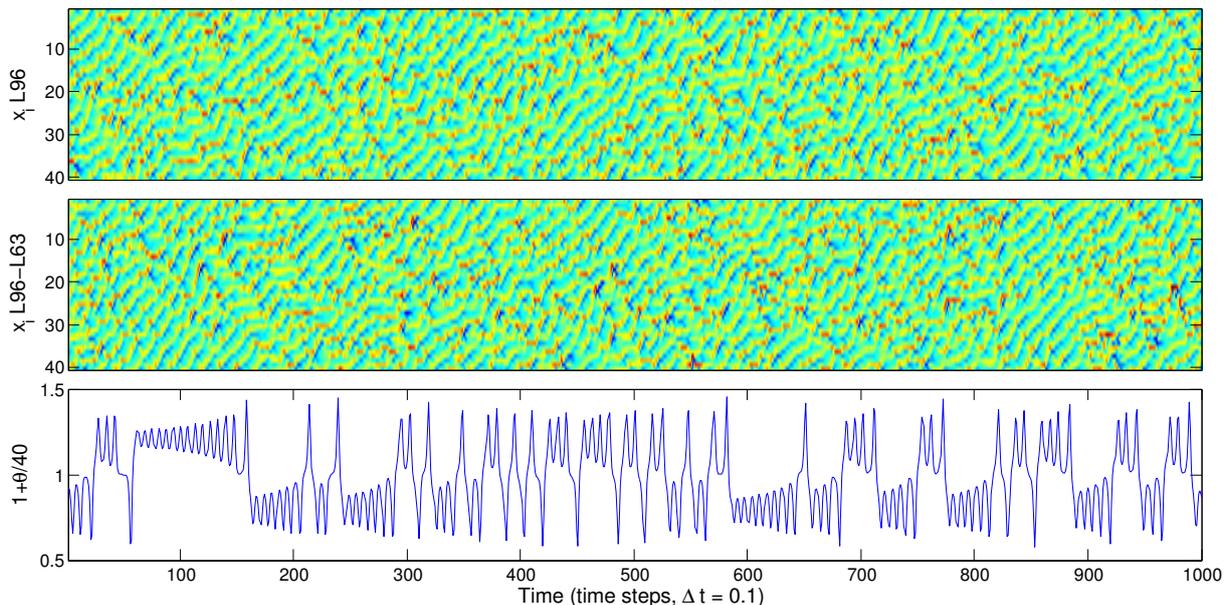


Figure 1: Comparison of the Lorenz-96 (top) model dynamics with the modified system (11) (middle) and the coefficient $(\theta/40 + 1)$ (bottom).

($\epsilon > 1$) time-scales in subsequent sections. In Figure 1 we compare the dynamics of original L96 model in (10) and the coupled L96-L63 system in (11). Notice that the spatio-temporal patterns of these two very different dynamical systems appear deceptively similar.

While this example is a proof of concept for semiparametric forecasting where we assume we are given a training time series, in real applications we can at most extract the time series of θ which appears explicitly in the parametric model. In Section 4 we describe a method of extracting a time series of θ from noisy observations, and we will see that we cannot hope to recover the three hidden variables $\{a_1, a_2, a_3\}$. With this realistic issue in mind and since the hidden variables do not appear explicitly in the parametric model f in (11), we will train the nonparametric model using only a time series of 5000 values of $\theta_i = \theta(t_i)$ (instead of a_1, a_2, a_3) with spacing $\Delta t = t_{i+1} - t_i = 0.1$. In order to account for the hidden variables in the nonparametric model, we perform a time delay embedding using 4 lags, meaning that we replace the one-dimensional time series θ_i with the time-delay embedding coordinates, $(\theta_i, \theta_{i-1}, \dots, \theta_{i-4})^\top$, prior to building the nonparametric model.

In order to compare the forecasting skill of various methods with imperfect initial conditions, we introduce a Gaussian perturbation to the true state,

$$(\tilde{x}(t), \tilde{\theta}(t)) = (x(t), \theta(t)) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, C(t)),$$

where the perturbation η_t has mean zero and variance equal to 0.1% of the long term variance of each variable. In other words, we define the covariance matrix,

$$C(t) = \begin{pmatrix} C_{xx}(t) & C_{x\theta}(t) \\ C_{\theta x}(t) & C_{\theta\theta}(t) \end{pmatrix}, \quad (12)$$

where $C_{xx}(t)$ is diagonal matrix of size 40×40 and $C_{\theta\theta}(t)$ is 1×1 for this example; and the diagonal entries of these sub-matrices are equal to 0.1% of the variance of the respective variables in (11). In this numerical experiment, we assume that the cross covariances $C_{x\theta}(t) = C_{\theta x}(t) = \vec{0}$, and moreover, notice that the covariance matrix $C(t)$ is actually the same for every initial time, t , in this experiment. In Section 5, the semiparametric filter solutions will produce time varying covariances and cross-covariance which are nonzero.

Each forecasting method will be given the perturbed initial state $(\tilde{x}(t), \tilde{\theta}(t))$ and the covariance matrix $C(t)$, and each method will attempt to forecast the true state $x(t+m\tau)$ for $m = 0, 1, \dots, 50$ time steps. Typically we are interested in

problems where only a small number K of ensemble members, $\{x^k(t)\}_{k=1}^K$, can be integrated; and a good choice are the sigma points of the covariance ellipse [31]. The sigma points are given by adding and subtracting the columns of the symmetric positive definite square root of the covariance matrix $C_{xx}(t)$, to the initial state estimate $\tilde{x}(t)$. This implies that the number of sigma points is twice the dimension of the model. Of course, for higher dimensional models this is often an unreasonably large ensemble, in which case the sigma points could be formed using the principal vectors of the covariance matrix. Adding and subtracting each column guarantees that the ensemble mean is exactly the initial state estimate.

For the persistence model, we simply fix $\theta = \tilde{\theta}(t)$ and apply the ensemble forecast to the parametric model with this fixed value of θ . For the MSM parameterization, we used the analytic solutions to the Ornstein-Uhlenbeck model (fit to the training data $\{\theta_i\}$) to forecast the initial mean $\tilde{\theta}(t)$ and covariance $C_{\theta\theta}(t)$ and we used the forecast mean at each discrete time step as the fixed parameter θ for the ensemble forecast of the parametric model. We also tried using the MSM forecast of the covariance to sample an ensemble of parameters, $\{\theta^k\}$, from the Gaussian forecast density, however we found that this was unstable for this problem. For the HMM parameterization, we immediately ignore $\tilde{\theta}(t)$ by drawing samples of initial conditions $\{\theta^k(t)\}_{k=1}^K$ from the equilibrium density, which means that each ensemble member θ^k is chosen by selecting a random index $m \in \{1, \dots, n\}$ and setting $\theta^k = \theta_m$ from the training data set. The ensemble of $\{\theta^k\}$ is combined with the ensemble $\{x^k\}$ and the parametric model is used to forecast one step and then the parameter ensemble $\{\theta^k\}$ is resampled and the process is repeated. For the perfect model, we generate an ensemble of $\{(x^k(t), a_1^k(t), a_2^k(t), a_3^k(t))\}_{k=1}^K$ using the sigma points of a 43×43 diagonal covariance matrix with diagonal entries equal to 0.1% of the long term variance of each variable and then perform the ensemble forecast using the full model (11). Note that the initial state for the full model is given by $\tilde{a}_1(t) = 40(\tilde{\theta}(t) - 1)$ and $(\tilde{a}_2(t), \tilde{a}_3(t)) = (a_2(t), a_3(t)) + (\eta_2(t), \eta_3(t))$ where η_2, η_3 are Gaussian with mean zero and variance equal to 0.1% of the variance of the respective variables. To initiate the nonparametric forecast in the semiparametric forecasting algorithm, we form an initial density for the parameters using a Gaussian $p(\theta_i, t) = \exp(-\|\tilde{\theta}(t) - \theta_i\|_{C_{\theta\theta}}^2)$ which we normalize by dividing $p(\theta_i, t)$ by the normalization factor (8). We then use the rejection method to sample $\{\theta^k(t)\}_{k=1}^K$ and perform the semiparametric forecast iteratively as described in Section 3.2.

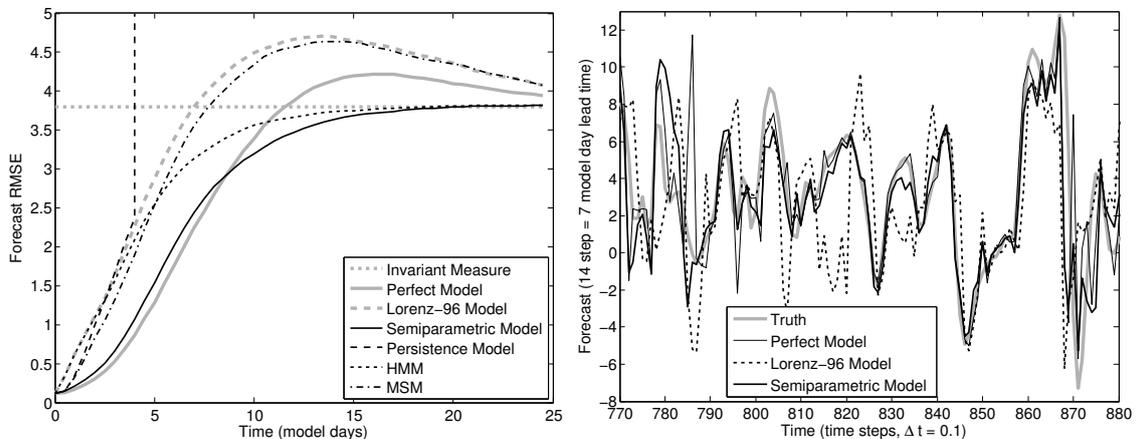


Figure 2: Comparison of forecasting methods for the model (11) with $\epsilon = 1$, given perfect training data and perturbed initial conditions for the parameter, θ . The left panel shows the forecast RMS errors as functions of forecast lead time and the right panel shows the 7 day lead-time forecast during the verification period of 770-880 time steps.

The above methods are applied to 1000 initial conditions, forecasting for a total of 50 time steps of length $\tau = 0.1$, which corresponds to 5 model time units, or equivalently 25 model days (using the convention of 1 model day being 0.2 model time units of [30]). The results are shown in Figure 2 with the root mean squared error (RMSE) averaged over the 1000 initial conditions. With only 86 ensemble members, the ensemble forecast using the perfect model (11) produces the best short term prediction, but it also seems to produce a biased forecast. This bias is shown by the RMS error rising above that of the climatological error in the intermediate term beyond 12 model days (which corresponds to 24 forecast steps, or 2.4 model time units). In contrast, the RMS error of the semiparametric forecast grows slightly

more quickly than the perfect model initially, but the forecast is unbiased in the intermediate term, approaching the climatology without exceeding the climatological error. In fact, after 8 model days the semiparametric forecast is producing a better forecast than the perfect model. This is due to the samples of the nonparametric forecast being independent samples of the forecast density, rather than being constrained to follow the paths of the initial ensemble members. Notice that the HMM, which immediately samples the equilibrium density p_{eq} by drawing random samples from the training data, initially matches the forecasting skill of the unmodified Lorenz-96 model of (10), and in the long term the HMM is unbiased whereas the unmodified skill model is very biased. Finally, the persistence model and the MSM use the initial condition $\theta(t)$ to improve the very short term forecast, but in the long term these methods do not capture the true climatology of θ , leading to a biased intermediate to long term forecast. In fact, the persistence model is so biased that the forecast model diverges catastrophically; this is because the model in (11) does not conserve energy when θ is held fixed at a value other than one. By varying θ in time, with a mean value of 1, all the models except for the persistence model are able to give a stable forecast. In the short term, the semiparametric forecast is able to provide an additional 1-2 model days of equivalent forecast skill compared to the standard methods considered here, and almost matches the perfect model forecast (see the 7 day lead-time forecast in the bottom panel of Figure 2 during the verification period 770-880 time steps). In the intermediate term forecast, the lack of bias in the semiparametric forecast provides a significant improvement in forecast skill, even compared to the perfect model for this small ensemble size.

4. Extracting a time series of hidden parameters

In this section, we will consider the more practical situation in which the historical time series of the parameters $\{\theta_i\}$ is not available. We will show that it is possible to extract such a training data set from noisy observations $y^o = y + \eta = h(x) + \eta$ where $\eta \sim \mathcal{N}(0, R)$ is a Gaussian observation noise. Of course, this recovered data set will not be perfect and it will not exactly obey the dynamics of θ in (11). In Section 5, we will show that the recovered data set still allows for training a useful nonparametric model which improves the forecast of x . If θ were fixed, or we had a parametric form for the dynamics of θ , then extracting the time series of θ from the noisy observations y^o would be a standard inverse problem. However, we would like to recover a time series of a dynamically evolving θ without knowing anything about the evolution of θ . This type of inverse problem should require at a minimum the same observability conditions as that of finding a fixed parameter θ . Our goal here is not to give a formal exposition of this inverse problem, but rather to present a proof-of-concept approach which shows that extracting a time series of θ is possible in a reasonably difficult context, and that the imperfectly recovered time series of θ can still be used to improve forecasting via a semi-parametric model.

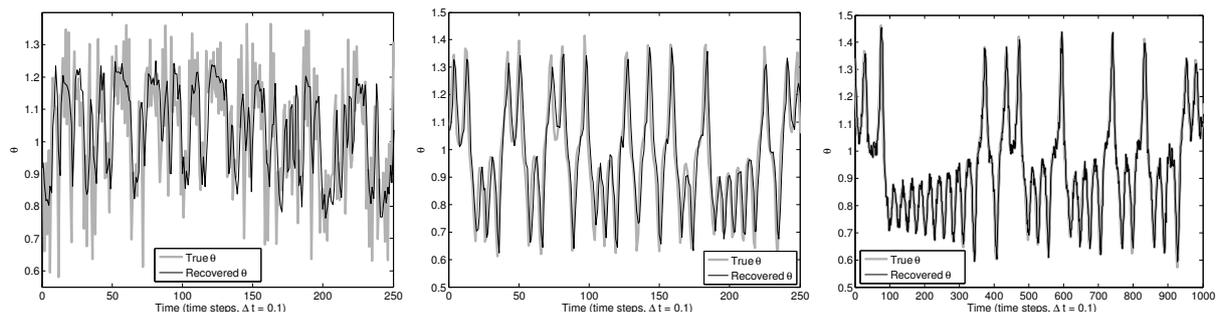


Figure 3: Comparison the true time series of θ and the recovered time series for $\epsilon = 0.25$ (left), $\epsilon = 1$ (middle), and $\epsilon = 4$ (right).

The approach we take here follows closely the standard state augmentation approach for parameter estimation. We will recover the time series of θ as part of an ensemble Kalman filter applied to recover the state variable x from the noisy observations y . This approach requires augmenting (1) with a model for θ . If θ were simply a constant parameter we could use the persistence model, $\dot{\theta} = 0$, and this corresponds to the standard *state augmentation* approach to parameter estimation [32]. Instead, because θ is changing in time, we will assume a white noise model, $\dot{\theta} = \sqrt{Q_{\theta\theta}}\dot{W}$. A typical approach is to empirically tune the value of $Q_{\theta\theta}$ [33] possibly using some a priori knowledge of the variable

θ . Intuitively, the true signal θ is being treated as a realization of a white noise process, and so a good choice for $Q_{\theta\theta}$ would be the variance of the time series $\{\theta_i\}$. Of course, since recovering the time series θ_i is our goal, we do not assume that the variance is known, and instead we use a method of adaptive filtering which can automatically recover $Q_{\theta\theta}$. The method used here to find $Q_{\theta\theta}$ is detailed in [Appendix A](#) and is a version of the approach introduced in [10] and applied in [9], which is a nonlinear generalization of an earlier approach in [34]. We note that a closely related technique was simultaneously developed in [8] and is refined in [11]. Intuitively, these methods use the error between the predicted and actual observations (which is also known as the innovation), to determine the amount of noise which must have been added to the state in order to explain this error. By analyzing correlations between the innovations at different lags it is possible to find the observation covariance R as well as observable parameters of the dynamical noise, such as $Q_{\theta\theta}$.

Once the value of $Q_{\theta\theta}$ is found from the adaptive EnKF, we use a standard EnKF to solve the following filtering problem,

$$\begin{aligned}\dot{x} &= f(x, \theta) \\ \dot{\theta} &= \sqrt{Q_{\theta\theta}} \dot{W} \\ y^o &= y + \eta = h(x) + \eta,\end{aligned}\tag{13}$$

where the goal is to recover the optimal estimates of (x, θ) at each time t_i , given all the observations $y^o(t_j)$ for $j \leq i$. Note that in the examples in this paper there is no stochastic forcing on the x variables, however this approach would allow an additive white noise forcing term of the form $\dot{x} = f(x, \theta) + \sqrt{Q_{xx}} \dot{W}_x$. Furthermore, we will only consider examples where the observation function h has no explicit dependence on θ since having direct observations of θ would typically make recovering θ less challenging. In [Figure 3](#) we show the results of applying this algorithm for the model (11) using an unscented ensemble Kalman filter [31] with 82 ensemble members (the filter equations used here are the same as those in [Section 5](#) for the parametric model).

In this example, we use a time series of 5000 noisy observations y^o with $\Delta t = t_{i+1} - t_i = 0.1$, the observation function is $h(x, \theta) = x$, and the observation noise variance is $R = \frac{1}{8} I_{40}$. In [Figure 3](#) we show the true time series of θ and the recovered time series for $\epsilon = 0.25, 1, 4$. In each case, the true variance of θ is 0.039, and the estimated variances are 0.0192, 0.0193, and 0.0066 for $\epsilon = 0.25, 1, 4$ respectively. When the Lorenz-63 system evolves more slowly, as in the cases $\epsilon = 1, 4$, a very accurate recovery is possible. In contrast, when $\epsilon = 0.25$, the Lorenz-63 system evolves quickly relative to the observations y of the state x , which makes clean recovery challenging, however the recovered time series is still highly correlated to the true dynamics. Obviously, one would get worse recovery when observation function h is a more complicated operator (for example, if the observations of the state x are sparse) or when the observation noise covariance R is larger. While these issues can be important in practice, we omit exploring them here since it is more related to the accuracy of the primary filter (EnKF) and our assumption is that θ can be recovered with an appropriate ensemble filtering method from the existing ensemble forecasting infrastructure.

In [Section 5](#) we will compare the filtering and forecasting skill of the semiparametric approach to the various parametric models introduced in [Section 2](#) for these three time scales ($\epsilon = 0.25, 1, 4$). As we will see, even when the parameters evolve quickly the recovered data set is still useful in overcoming the model error.

5. Semiparametric filtering

In this section we assume that a training data set has already been recovered using the method of [Section 4](#) and our goal is to estimate the current state, $x(t)$, and the posterior density $p(\theta, t)$ at some time t given a sequence of noisy observations $y^o(s)$ for $s \leq t$, which will serve as the initial conditions for the forecasting problem of [Section 3](#). Recall that the nonparametric forecasting procedure involves writing a density in the basis $\{\varphi_j\}$ and propagating the coefficients c_j forward in time with a linear map \hat{A} , where \hat{A} and φ_j are built using the time series extracted by the method in [Section 4](#). Therefore, we are actually interested in the coefficients $c_j(t)$ which describe the conditional density $p(\theta(t) | y^o(s), s \leq t)$ written in the basis φ_j . In other words, given all of the observations up to and including the current observation, we would like to find the optimal description of the current state. To solve this problem, we need to combine our nonparametric forecast model with the parametric filter to form a semiparametric filter.

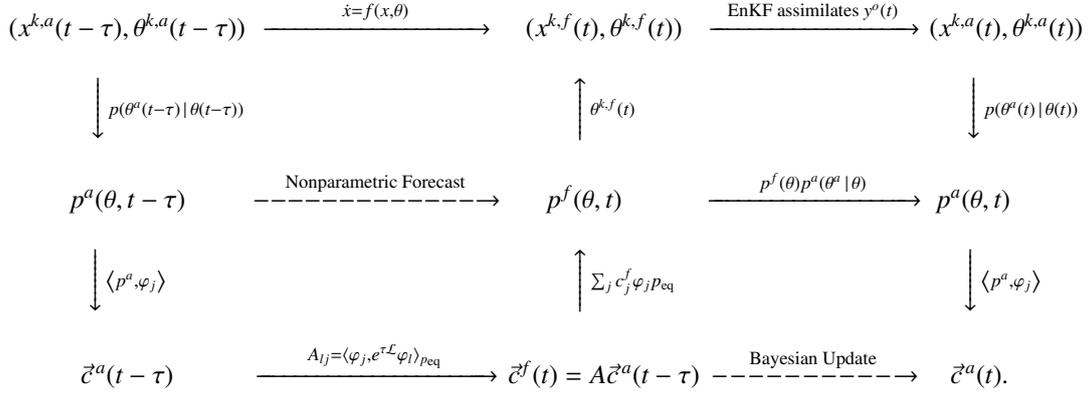


Figure 4: Diagram summarizing the semiparametric filtering algorithm.

The goal of semiparametric filtering is to combine the parametric model, f , for the evolution of x with a nonparametric model for the evolution of θ to determine the filtered state $p(x(t), \theta(t) | y^o(s), s \leq t)$ from the previous filtered state $p(x(t-\tau), \theta(t-\tau) | y^o(s), s \leq t-\tau)$. The algorithm introduced here will follow the standard *forecast-assimilate* paradigm for filtering, in which the *forecasting* step will be to perform a one step forecast to determine $p(x(t), \theta(t) | y^o(s), s \leq t-\tau)$ and the *assimilation* step uses the observation $y^o(t)$ to find the final density $p(x(t), \theta(t) | y^o(s), s \leq t)$.

We introduce the notation $x^{k,a}$ and $\theta^{k,a}$ to denote an ensemble of states x and parameters θ . The superscript k denotes the k -th ensemble member, and the superscript ‘a’ stands for analysis, indicating that this ensemble has assimilated all the information up to the current time, including the current observation. Similarly, $x^{k,f}$ and $\theta^{k,f}$ denote the forecast (also known as ‘prior’) ensemble, as indicated by the superscript ‘f’, which account for the observations up to the previous time step. To be consistent, we also use the same superscripts ‘a’ and ‘f’ to denote the nonparametric analysis density $p^a(\theta, t)$ and the nonparametric forecast density $p^f(\theta, t)$ as well as the associated coefficients c^a and c^f respectively.

We overview the semiparametric filtering algorithm with the diagram shown in Figure 4. As shown in Figure 4, information will flow up from the density to the ensemble in the forecast step and back down from the ensemble to the density in the assimilation step. The basic idea of the semiparametric filter is to apply the EnKF to estimate a Gaussian approximation to the posterior $p(x(t), \theta(t) | y^o(s), s \leq t)$ and then to treat the parametric filter estimate $\theta^a(t)$ as a noisy observation which we feed into a secondary nonparametric filter. The forecast step for θ can then be performed using the nonparametric model, and the first two statistics of this forecast are substituted into the Kalman filter prior statistics for θ .

To make this procedure precise, assume that we are given an estimate of the state $(x^a(t-\tau), \theta^a(t-\tau))$ along with a covariance matrix $C^a(t-\tau)$ of the form (12) and coefficients $c^a(t-\tau)$ which represent the current posterior density $p^a(\theta, t-\tau)$ given all the observations $y^o(s)$ for $s \leq t-\tau$. The filtering procedure is iterative, so given this information, it will assimilate the next observation $y^o(t)$ and produce $(x^a(t), \theta^a(t))$, $C^a(t)$ and $c^a(t)$.

To perform the forecast step of the semiparametric filter, we first form an ensemble $\{(x^{k,a}(t-\tau), \theta^{k,a}(t-\tau))\}_{k=1}^K$ of states which match the statistics $(x^a(t-\tau), \theta^a(t-\tau))$ and $C^a(t-\tau)$. In all the examples in this paper we use the unscented square root ensemble [31] as described in Section 3.3. Next, we use the model (1) to integrate this ensemble forward in time to find the forecast ensemble $\{x^{k,f}(t)\}$ (also known as the background ensemble). In order to update the statistics of θ , we run a single diffusion forecast step $c^f(t) = \hat{A} c^a(t-\tau)$. Using the ensemble $\{x^{k,f}(t)\}$ and the coefficients $c^f(t)$, we compute the forecast statistics (also known as the prior statistics or background statistics) by

mixing the parametric and nonparametric information as follows,

$$\begin{aligned}
x^f(t) &= \frac{1}{K} \sum_{k=1}^K x^{k,f}(t) \\
C_{xx}^f(t) &= \frac{1}{K-1} \sum_{k=1}^K (x^k(t) - x^f(t))(x^k(t) - x^f(t))^\top \\
\theta^a(t - \tau) &= \frac{1}{K} \sum_{k=1}^K \theta^{k,a}(t - \tau) \\
C_{x\theta}^f(t) &= C_{\theta x}^f(t)^\top = \sum_{k=1}^K \frac{(x^{k,f}(t) - x^f(t))(\theta^{k,a}(t - \tau) - \theta^a(t - \tau))^\top}{K-1} \\
p^f(\theta_i, t) &= \sum_{j=1}^M c_j(t) \varphi_j(\theta_i) p_{\text{eq}}(\theta_i) \\
\theta^f(t) &= \int_{\mathcal{M}} \theta p^f(\theta, t) dV(\theta) \approx \frac{1}{N} \sum_{i=1}^N \theta_i p^f(\theta_i, t) / p_{\text{eq}}(\theta_i) \\
C_{\theta\theta}^f(t) &= \int_{\mathcal{M}} (\theta - \theta^f(t))(\theta - \theta^f(t))^\top p^f(\theta, t) dV(\theta) \\
&\approx \frac{1}{N} \sum_{i=1}^N (\theta_i - \theta^f(t))(\theta_i - \theta^f(t))^\top p^f(\theta_i, t) / p_{\text{eq}}(\theta_i) \\
C^f(t) &= \begin{pmatrix} C_{xx}^f(t) & C_{x\theta}^f(t) \\ C_{\theta x}^f(t) & C_{\theta\theta}^f(t) \end{pmatrix}.
\end{aligned} \tag{14}$$

Notice that we use the ensemble to compute the statistics of x and the correlation between x and θ , while we use the nonparametric model to compute the statistics of θ . Moreover, the cross covariance terms $C_{x\theta}^f(t) = C_{\theta x}^f(t)^\top$ simply use the analysis ensemble $\theta^{k,a}(t - \tau)$ from the previous time step. This is required since we do not integrate the individual initial conditions $\theta^{k,a}$, but rather the full density. Using the semiparametric forecast mean $z^f(t) \equiv (x^f(t), \theta^f(t))$ and covariance $C^f(t)$, we now resample the forecast ensemble $z^{k,f}(t) = (x^{k,f}(t), \theta^{k,f}(t))$. In order to respect the covariance structure $C^f(t)$ we use a square root ensemble (adding and subtracting the columns of $\sqrt{C^f(t)}$ to the forecast mean). Another possibility would be to use the ensemble Kalman filter (EnKF) update strategy. For convenience we will use the same symbols for the resampled forecast ensemble. Naively, one might hope to simply apply the rejection sampling strategy to generate $\theta^{k,f}$, and keep using the ensemble $x^{k,f}$ from the ensemble forecast without resampling; however, this strategy does not preserve the cross-covariance structure between x and θ from the previous filter step. We attempted this alternative strategy and found that it significantly degraded the filter performance compared to resampling the ensemble $z^{k,f}(t)$ using (14). With the newly resampled ensemble, we form the observation ensemble $y^{k,f}(t) = h(x^{k,f}(t), \theta^{k,f}(t))$ by applying the observation function to the reformed ensemble. Notice that in general the observation function may depend on both the state x and the parameters θ , but in the examples below we will only consider the more difficult case where the observation function only depends on x (so that θ is not directly observed).

With the forecast step completed, we now perform the assimilation step, which will consist of an EnKF assimilation step and a secondary nonparametric assimilation step. We first perform the EnKF update in order to assimilate the information given by observations, $y^o(t)$, at time t , with the standard equations,

$$\begin{aligned}
y^f(t) &= \frac{1}{K} \sum_{k=1}^K y^{k,f}(t) \\
z^f(t) &= \frac{1}{K} \sum_{k=1}^K z^{k,f}(t) \\
C_{zy}(t) &= \frac{1}{K} \sum_{k=1}^K (z^{k,f}(t) - z^f(t))(y^{k,f}(t) - y^f(t))^\top \\
C_{yy}(t) &= \frac{1}{K} \sum_{k=1}^K (y^{k,f}(t) - y^f(t))(y^{k,f}(t) - y^f(t))^\top \\
K(t) &= C_{zy}(t) C_{yy}(t)^{-1} \\
z^a(t) &= (x^a(t), \theta^a(t)) = z^f(t) + K(t)(y^o(t) - y^f(t)) \\
C^a(t) &= C^f(t) - K(t) C_{yy}(t) K(t)^\top.
\end{aligned} \tag{15}$$

The ensemble Kalman update (15) yields the analysis mean $(x^a(t), \theta^a(t))$ and covariance $C^a(t)$ from the parametric model, and it remains to update the nonparametric model coefficients $\tilde{z}^f(t)$ in order to find the analysis coefficients $\tilde{z}^a(t)$ which are required in order to iterate the filter. Notice that if we do not update these coefficients to incorporate information from $y^o(t)$, they will evolve to represent the equilibrium density p_{eq} , and the forecast θ^f will eventually be constant, so this secondary assimilation step is crucial.

In order to update $\tilde{z}^f(t)$, we will take the posterior estimate $\theta^a(t)$ from the EnKF, and treat it as a noisy observation, which is reasonable since it is only an estimate of the true state, and the filter also gives us an error

estimate, namely the analysis covariance, $C_{\theta\theta}^a(t)$. If the observations were continuous in time, we could filter this observation using the Zakai equation projected onto the basis $\{\varphi_j\}$, and this was the approach taken in [15]. However, since the observation time may be long, we will apply a Bayesian update which was introduced in [17]. Explicitly, we want to find the posterior density $p^a(\theta, t) = p(\theta, t | y^o(s), s \leq t)$ by combining the forecast density $p^f(\theta, t) = p(\theta, t | y^o(s), s < t) = \sum_{j=1}^M c_j^f(t) \varphi_j(\theta) p_{\text{eq}}(\theta)$ from the nonparametric model with the EnKF analysis estimate $\theta^a(t)$. Since the ensemble Kalman filter assumes the analysis solutions to be Gaussian, we have the following marginal analysis density for θ ,

$$p(\theta | y^o(t)) \approx \tilde{p}(\theta | y^o(t)) \propto \exp\left(-\frac{1}{2} \|\theta - \theta^a(t)\|_{C_{\theta\theta}^a}^2\right). \quad (16)$$

We will treat this Gaussian analysis density as a likelihood function, that is, we define the likelihood function $p(\theta^a(t) | \theta) = \tilde{p}(\theta | y^o(t))$. We now use this likelihood function in the following Bayesian assimilation scheme,

$$\begin{aligned} p^a(\theta, t) &\propto p^f(\theta, t) p(\theta^a(t) | \theta) \\ &\propto \sum_{j=1}^M c_j^f(t) \varphi_j(\theta) p_{\text{eq}}(\theta) \exp\left(-\frac{1}{2} \|\theta - \theta^a(t)\|_{C_{\theta\theta}^a}^2\right). \end{aligned} \quad (17)$$

By evaluating (17) at each data point θ_i from the training data set, we can find $p^a(\theta, t)$ up to a normalization factor which we compute with (8). We can then compute the updated coefficients $c_l^a(t)$ by projecting the analysis density $p^a(\theta, t)$ onto the basis φ_l by computing,

$$c_l^a(t) = \langle p^a(\cdot, t), \varphi_l \rangle \approx \frac{1}{N} \sum_{i=1}^N \frac{p^a(\theta_i, t)}{p_{\text{eq}}(\theta_i)} \varphi_l(\theta_i). \quad (18)$$

With the updated coefficients $\tilde{c}^a(t)$ from (18) we are now ready to perform the next filter step. Moreover, at this point we can apply the semiparametric forecasting algorithm of Section 3 to the current state, represented by $x^a(t)$, $C_{xx}^a(t)$ and $\tilde{c}^a(t)$ to estimate the future densities $p(\theta, t + m\tau | y^o(s), s \leq t)$ at future times $t + m\tau$, given only the information up to the current time, t .

Initializing the mean $(x^a(0), \theta^a(0))$ and covariance $C^a(0)$, where time zero simply represents the first observation one desires to assimilate, is a standard problem in filtering and in this paper we will use the true initial state $(x(0), \theta(0))$ along with an empirically chosen diagonal covariance matrix with $C_{xx}^a = I_{40}$ and $C_{\theta\theta}^a = Q_{\theta\theta}$. To initialize the coefficients $\tilde{c}^a(0)$ we will use a non-informative prior, meaning that the initial density $p^a(\theta, 0) = p_{\text{eq}}(\theta)$ so that $c_j^a(0) = \langle p_{\text{eq}}, \varphi_j \rangle \approx \frac{1}{N} \sum_{i=1}^N \varphi_j(\theta_i)$.

5.1. Semiparametric filtering and forecasting a system with chaotic model error

In this example we apply the semiparametric filtering technique to the system (11) for $\epsilon \in \{0.25, 1, 4\}$ given only the model $f(x, \theta)$ and a time series of 5000 noisy observations $y^o(t_i)$ with $\Delta t = t_{i+1} - t_i = 0.1$, and the observation noise variance is $R = \frac{1}{8} I_{40}$. Implicitly, this means that we are applying the complete procedure, including estimating the variance of the parameters θ as in Appendix A, recovering a time series of the parameters θ as in Section 4, building the nonparametric model for θ , including applying diffusion maps algorithm to obtain the basis $\varphi_j(\theta_i)$ and the forecasting matrix \hat{A} as discussed in Section 3.1, implementing semiparametric filtering as described in Section 5, and semiparametric forecasting as described in Section 3. Note that we apply the time delay embedding to the recovered time series θ_i and use this embedded data set to build the nonparametric model, including the equilibrium density $p_{\text{eq}}(\theta_i)$, the basis $\varphi_j(\theta_i)$ and the forecasting matrix \hat{A} . Finally, we apply the semiparametric filter to an out-of-sample collection of 1000 noisy observations $y^o(t_i)$ where $i = 5101, \dots, 6100$. At each step, using the filter analysis as the initial conditions for the semiparametric forecast, we compute the forecast for 50 steps of length $\tau = \Delta t = 0.1$ (which is 25 model days using the convention of 1 model day = 0.2 model time units) and we compare the mean of the forecast to the true state x at the corresponding lead times.

In Figure 5 we compare the semiparametric filter to alternative methods of correcting for the model error, all of which use the same training data set for θ_i . Notice that when the standard Lorenz-96 model in (10) is used without any modification, the ensemble Kalman filter cannot even estimate the initial state x , and the forecast is worse than the climatology of the true system. The perfect model produces the most accurate forecasts for larger $\epsilon = 1, 4$. Notice that

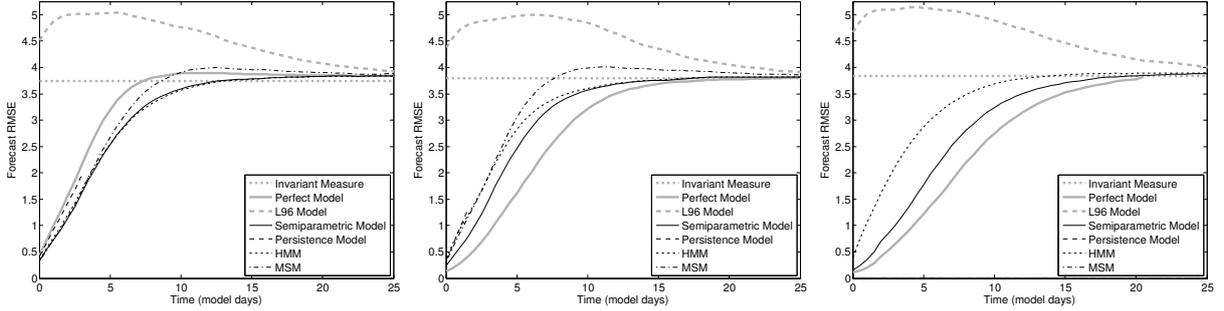


Figure 5: Comparison of forecasting methods for (11) with recovered training data and filtered initial condition for $\epsilon = 0.25$ (left), $\epsilon = 1$ (middle), and $\epsilon = 4$ (right).

for $\epsilon = 1$, the perfect model in Figure 5 does not exhibit the bias as seen in the long term forecast of Figure 2. This is due to the ensemble from the filtered state having non-diagonal covariance structure which places the ensemble closer to the attractor of the true system (11) and reduces the bias in the forecast. As shown in Figure 5, when $\epsilon = 0.25$ the system (11) is sufficiently stiff that the perfect model does not produce good filtered estimates and consequently the perfect model is outperformed by the alternative approaches.

The semiparametric filter and forecast give the next best forecasting skill and this approach is robust across the different values of ϵ . When the Lorenz-63 system evolves more slowly ($\epsilon = 1, 4$), the semiparametric model has a significant advantage over all except for the perfect model. The next best approach is HMM, which samples randomly from the training data set (see [35] for detailed implementation of the HMM in the data assimilation procedure). This approach matches the semiparametric method when the Lorenz-63 system is very fast ($\epsilon = 0.25$), which agrees with the theory of HMM in [2], because when ϵ is small the Lorenz-63 system goes to the equilibrium density very quickly relative to the Lorenz-96 system which means that forecast model \hat{A} cannot provide any additional information to the forecast. Finally, MSM performs quite well for small ϵ , but the predictive skill degrades severely as ϵ increases (catastrophic divergence in the case of $\epsilon = 4$). The persistence model forecast exhibits the catastrophic divergence for all values of ϵ since the model does not conserve energy.

5.2. Semiparametric filtering and forecasting a system with stochastic model error

In this example we apply the semiparametric filtering technique to the following system with four parameters $\theta = (\theta_1, \dots, \theta_4)$, each parameter θ_j is paired with the 10 state variables x_i with $\text{ceil}(i/10) = j$,

$$\begin{aligned} f(x_i, \theta) &= \theta_{\text{ceil}(i/10)} x_{i-1} x_{i+1} - x_{i-1} x_{i-2} - x_i + 6 \\ \theta_j &= 1 + (3/10) \sin(\gamma + (\pi/4)j) \\ \dot{\gamma} &= -(2 - \sin(2\gamma)/2)/\epsilon + \sqrt{1/\epsilon} \dot{W}, \end{aligned} \tag{19}$$

for $\epsilon \in \{0.25, 1, 4\}$. Although the parameter space θ is four dimensional, these parameters actually lie on a one-dimensional manifold \mathcal{M} embedded in the ambient 4-dimensional space. In this example, we set this one dimensional manifold to be a circle with latent coordinate γ which has a state dependent drift and a small amount of diffusion on the circle. We chose this example to demonstrate the semiparametric model on a higher-dimensional parameter space $\theta \in \mathbb{R}^4$ with stochastic evolution. Since the parameters θ are actually constrained to a one-dimensional physical domain, we expect to learn the model for the parameters with a very small training data set of noisy observations. Finally, this example will demonstrate the strength semiparametric approach since it is nontrivial to represent the hidden latent variable γ with a parametric ensemble forecast. Given only the model $f(x, \theta)$ and a time series of 5000 noisy observations y^o with $\Delta t = 0.1$ and observation noise variance $R = \frac{1}{32} I_{40}$, we apply the method of Appendix A to estimate the 4×4 covariance matrix of the parameters θ , and then we apply the method of Section 4 to recover the time series of θ . We found that accurate recovery of the covariance matrix and the time series of θ required a lower observation noise in this example than in the previous example, however a full exploration of the effect of observation noise level is beyond the scope of this paper. As an example, we show the recovered estimate of the time series of θ_1 is compared to the true time series of θ_1 in Figure 6 for $\epsilon = 4$.

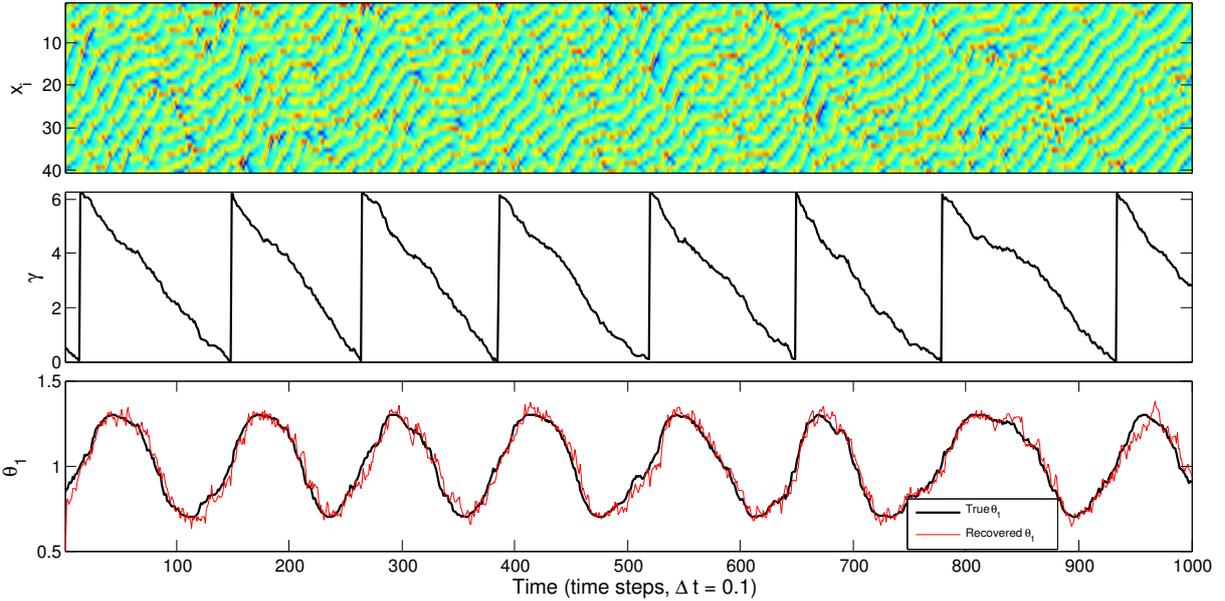


Figure 6: Demonstrating the evolution of the system (19) with $\epsilon = 4$ (top) with the corresponding time series of the latent variable γ (middle, plotted modulo 2π) and the first parameter θ_1 (bottom, black curve) along with the recovered time series of θ_1 estimated with the filter (bottom, red curve).

Even though the state γ is fully observable from the four dimensional time series θ , we still apply a time delay embedding to the recovered training time series with a single lag to smooth out the remaining noisy recovered time series. Notice in Figure 6 that the reconstruction of the true time series of parameters is somewhat noisy. It was shown in [17], applying the theory of [24], that the delay embedding can be used to reduce the influence of the noise on the nonparametric model. We note that while there is not a strong dependence on the number of lags used in the time delay embedding for this example, the theory introduced in [24] shows that as the number of delays increases, the geometry of the attractor becomes biased towards the stable components of the dynamics. Intuitively, projecting onto the stable components leads to a smoother reconstruction.

We learn the nonparametric model from 5000 recovered training data points of θ , and apply the semiparametric filter to an out-of-sample collection of 1000 noisy observations $y^o(t_i)$ where $i = 5101, \dots, 6100$, computing the forecast up to the 50 step (5 model time unit, 25 model day) lead time as in the previous example. In Figure 7 we see that the nonparametric filter significantly outperforms the perfect model filter across all three time scales. The poor performance of the perfect model is due to the small ensemble size, with only 82 ensemble members unable to capture the evolution of the stochastic system (19). Notice that any small perturbation of the latent variable γ leads to significant variability in the high-dimensional chaotic evolution of x , so sufficiently sampling the uncertainty of θ is difficult with a small ensemble of only 82 paths. In contrast, as previously noted, the semiparametric model does not produce paths of the evolution of γ , but rather samples independently from the forecast density at each forecast step. Only when the system is very slow, $\epsilon = 4$, is there a comparable filter, and in this case simply using the persistence model gives reasonable estimates. However, while the persistence model gives a reasonably good forecast in the very short term for the case $\epsilon = 4$, the long term forecast quickly degrades and in this case the forecast is unstable due to the dynamics (19) being unstable when θ is fixed at a value far from $(1, 1, 1, 1)$. Attempting to use the standard Lorenz-96 system (10) fails to filter the observations due to the model error, and while both the perfect model and the HMM give stable filters and forecasts they are significantly outperformed by the semiparametric model.

6. Conclusion

The goal of semiparametric modeling is to blend the strengths of parametric and nonparametric models. In particular, parametric models are able to use prior knowledge to describe intrinsically high-dimensional phenomena,

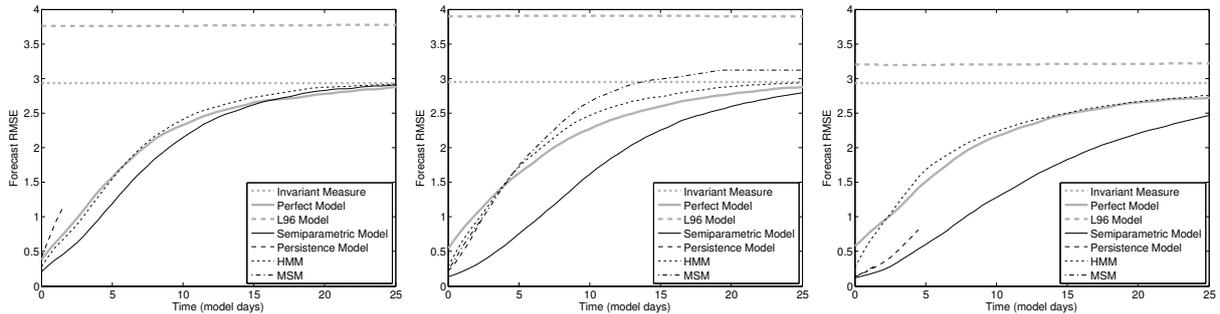


Figure 7: Comparison of forecasting methods for the L96-stochastic model with recovered training data and filtered initial condition for $\epsilon = 0.25$ (left), $\epsilon = 1$ (middle), and $\epsilon = 4$ (right).

however they are vulnerable to model error arising from unresolved phenomena or truncated scales. On the other hand, nonparametric models learn directly from the data and place very few prior assumptions on the data, however the data requirements grow exponentially in the intrinsic dimensionality of the problem.

We test our approach on a simple example, in which model error is introduced into the Lorenz-96 model with a parameter governed by either the Lorenz-63 model or a one-dimensional stochastic model. These examples were chosen as challenging test cases for semiparametric modeling in the following sense: The advection term in (11) or (19) does not actually conserve energy due to the parameter θ . Of course, the average value of θ in (11) or (19) is 1, so averaged over a large enough time window the L96-L63 advection does approximately conserve energy. As an analogy to a real physical problem, treating the L96-L63 model as the truth, one can imagine that a coarse physical model may have suggested using the Lorenz-96 model, which exactly conserves energy, while the true system does not. The difference between these two systems is very subtle as shown by the spatio-temporal patterns in Figure 1. However, this small model error has a dramatic effect on forecasting skill as shown in Figure 2 and even more so on the filtering as shown in Figure 5. The stochastic example in (19) is an even a more difficult test bed in the sense that model errors are introduced by four parameters whose intrinsic dynamics is only one-dimensional. Attempting to fit a model to the four dimensional ambient space would not respect the intrinsically one-dimensional structure of the attractor. Alternatively, one could fit four separate one-dimensional models, however this would not respect the correlations between the variables.

The examples discussed above suggest that coarse physical models can be significantly improved by the semiparametric modeling paradigm. Rather than choosing arbitrary models for unknown phenomena, the modeler can build a parametric model which captures all the known phenomena, and fit the remaining unknown phenomena with nonparametric modeling. Of course, if some knowledge of the parameters is available, one should first extend the parametric model as far as possible using real physical knowledge. Only when the physical knowledge is exhausted should the semiparametric modeling be applied, thereby reducing the dimensionality of the model error as much as possible before learning the nonparametric model. Choosing an appropriate parametric form to encode all the existing physical knowledge is a significant challenge, and the semiparametric modeling approach takes advantage of all the existing knowledge (represented by the physical model) to reduce the load on nonparametric model.

The algorithms introduced in this paper strive to be practical by seamlessly blending the nonparametric model into a standard existing framework for parametric filtering and forecasting. However, there are two significant practical limitations of this framework. First, in order to extract a time series of the parameters (to be used for building the nonparametric model), one requires a minimum observability condition for the parameters, as in any standard inverse problem. Assuming that this condition is met, we foresee that improving the methodology for extracting the time series of the hidden parameters from noisy data is the key remaining challenge to lifting this framework to real applications. Indeed, the approach introduced in Section 4 and Appendix A is intended only as a proof-of-concept which shows that extracting an unobserved time series of parameters is possible. Different parameter estimation schemes for extracting the training time series are worth considering, such as the MCMC [36]. Second, the evolution of the parameters $\dot{\theta} = g(\theta, W)$ is assumed to be independent of the state variable x of the parametric model. This assumption is currently required because the nonparametric model of [16] assumes that the evolution represented by

the times series is ergodic, and if g were allowed to depend on x the time series θ would not be ergodic by itself. Of course, combining the state variables (x, θ) would typically yield an ergodic time series, however, this system would once again be high-dimensional and would not be accessible to a nonparametric model. Similarly, if g depended on x , the theory of Takens [37, 38] suggests that a time-delay embedding of θ would attempt to reconstruct the attractor of the full system (x, θ) which again would again be limited by the curse-of-dimensionality. One possible avenue toward overcoming this restriction would be extending the state space θ to include sufficient information $\tilde{\theta}$ so that $(\theta, \tilde{\theta})$ is a closed model. Of course, the challenge is identifying the correct variables $\tilde{\theta}$, and in general this model closure problem is extremely difficult.

Acknowledgment

The research of J.H. is partially supported by the Office of Naval Research Grants N00014-11-1-0310, N00014-13-1-0797, MURI N00014-12-1-0912 and the National Science Foundation DMS-1317919. T. B. is supported under the ONR MURI grant N00014-12-1-0912.

References

- [1] W.W. Grabowski. Coupling cloud processes with the large-scale dynamics using the Cloud-Resolving Convection Parameterization (CRCP). *J. Atmos. Sci.*, 58:978–996, 2001.
- [2] W. E, B. Engquist, X. Li, and W. Ren. Heterogeneous multiscale methods: A review. *Commun. Comput. Phys.*, 2(3):367–450, 2007.
- [3] Y. Frenkel, A.J. Majda, and B. Khouider. Using the stochastic multicloud model to improve tropical convective parameterization: A paradigm example. *Journal of the Atmospheric Sciences*, 2012.
- [4] A.J. Majda and I. Grooms. New perspectives on superparameterization for geophysical turbulence. *Journal of Computational Physics (in press)*, 2013.
- [5] M.A. Katsoulakis, A.J. Majda, and A. Sopsakis. Multiscale couplings in prototype hybrid deterministic/stochastic systems: Part I, Deterministic closures. *Comm. Math. Sci.*, 2(2):255–294, 2004.
- [6] M.A. Katsoulakis, A.J. Majda, and A. Sopsakis. Multiscale couplings in prototype hybrid deterministic/stochastic systems: Part II, Stochastic closures. *Comm. Math. Sci.*, 3(3):453–478, 2004.
- [7] A.J. Majda and J. Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26:201–217, 2013.
- [8] J. Harlim, A. Mahdi, and A.J. Majda. An ensemble kalman filter for statistical estimation of physics constrained nonlinear regression models. *Journal of Computational Physics*, 257, Part A:782 – 812, 2014.
- [9] T. Berry and J. Harlim. Linear theory for filtering nonlinear multi scale systems with model error. *Proc. R. Soc. A*, 470(2167), 2014.
- [10] T. Berry and T. Sauer. Adaptive ensemble Kalman filtering of nonlinear systems. *Tellus A*, 65:20331, 2013.
- [11] Y. Zhen and J. Harlim. Adaptive error covariance estimation methods for ensemble kalman filtering. *Journal of Computational Physics*, 273:619–638, 2015.
- [12] H. M. Arnold, I. M. Moroz, and T. N. Palmer. Stochastic parametrizations and model uncertainty in the lorenz 96 system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1991), 2013.
- [13] W. Härdle. *Nonparametric and semiparametric models*. Springer, 2004.
- [14] D. G. Terrell and D. W. Scott. Variable kernel density estimation. *Annals of Statistics*, 20:1236–1265, 1992.
- [15] T. Berry and J. Harlim. Nonparametric uncertainty quantification for stochastic gradient flows. *SIAM/ASA J. Uncer. Quant. (in press)*, 2015.
- [16] T. Berry, D. Giannakis, and J. Harlim. Nonparametric forecasting of low-dimensional dynamical systems. *Physical Review E*, 91:032915, 2015.
- [17] T. Berry and J. Harlim. Forecasting turbulent modes with nonparametric diffusion models. *submitted to Physica D*, 2015.
- [18] A.J. Majda, I. Timofeyev, and E. Vanden-Eijnden. Models for stochastic climate prediction. *Proc. Nat. Acad. Sci.*, 96:15687–15691, 1999.
- [19] R. Z. Khasminskii. Principle of averaging for parabolic and elliptic differential equations and for markov processes with small diffusion. *Theory of Probability & Its Applications*, 8(1):1–21, 1963.
- [20] A.J. Majda, B. Gershgorin, and Y. Yuan. Low frequency response and fluctuation-dissipation theorems: Theory and practice. *J. Atmos. Sci.*, 67:1181–1201, 2010.
- [21] T. Sauer, J.A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3-4):579–616, 1991.
- [22] J. Stark. Delay embeddings for forced systems. I. Deterministic forcing. *Journal of Nonlinear Science*, 9:255–332, 1999.
- [23] J. Stark, D.S. Broomhead, M.E. Davies, and J. Huke. Delay embeddings for forced systems. II. Stochastic forcing. *Journal of Nonlinear Science*, 13:519–577, 2003.
- [24] T. Berry, J. R. Cressman, Z. Gregurić Ferenček, and T. Sauer. Time-scale separation from diffusion-mapped delay coordinates. *SIAM J. Appl. Dyn. Syst.*, 12:618–649, 2013.
- [25] D. Giannakis and A. J. Majda. Time series reconstruction via machine learning: Revealing decadal variability and intermittency in the north pacific sector of a coupled climate model. In *Conference on Intelligent Data Understanding (CIDU)*, pages 107–117, Mountain View, California, 2011.
- [26] D. Giannakis and A. J. Majda. Nonlinear laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proc. Nat. Acad. Sci.*, 109(7):2222–2227, 2012.
- [27] R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21:5–30, 2006.

- [28] T. Berry and J. Harlim. Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 2015.
- [29] E.N. Lorenz. Predictability - a problem partly solved. In *Proceedings on predictability, held at ECMWF on 4-8 September 1995*, pages 1–18, 1996.
- [30] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.
- [31] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [32] B. Friedland. Treatment of bias in recursive filtering. *IEEE Trans. Automat. Contr.*, AC-14:359–367, 1969.
- [33] B. Friedland. Estimating sudden changes of biases in linear dynamical systems. *IEEE Trans. Automat. Contr.*, AC-27:237–240, 1982.
- [34] R.K. Mehra. Approaches to adaptive filtering. *Automatic Control, IEEE Transactions on*, 17(5):693–698, 1972.
- [35] E.L. Kang and J. Harlim. Filtering partially observed multiscale systems with heterogeneous multiscale methods-based reduced climate models. *Monthly Weather Review*, 140(3):860–873, 2012.
- [36] A. Golightly and D.J. Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674 – 1693, 2008.
- [37] F. Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young, editors, *In: Dynamical Systems and Turbulence, Warwick, Eds. Rand, D. and Young, L.-S.*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin / Heidelberg, 1981.
- [38] T. Sauer, J.A. Yorke, and M. Casdagli. Embedology. *J. Stat. Phys.*, 65(3):579–616, 1991.

Appendix A. Estimating the variance of unobserved and unmodeled parameters

In this appendix we consider the filtering problem,

$$\begin{aligned}
 \dot{x} &= f(x, \theta) \\
 \dot{\theta} &= g(\theta, \dot{W}) \approx \sqrt{Q_{\theta\theta}} \dot{W} \\
 y &= h(x) + \eta
 \end{aligned} \tag{A.1}$$

where we have access to the noisy observations y at discrete times t_i and the noise η is normal with variance R . We assume that the model f and the observation function h are known, but that the model g is unknown and we will instead use a white noise model $\tilde{g}(\theta, \dot{W}) = \sqrt{Q_{\theta\theta}} \dot{W}$. We will apply the method of [10] to find the variance $Q_{\theta\theta}$ of the unmodeled parameters θ in (A.1). The method of [10] finds the covariance matrix $Q = \begin{pmatrix} Q_{xx} & Q_{x\theta} \\ Q_{\theta x} & Q_{\theta\theta} \end{pmatrix}$ of the full system (x, θ) . Thus the technique developed here will allow stochastic forcing with covariance Q_{xx} on the x variable, although none of the examples in this paper have this feature.

The method of [10] requires us to parameterize the matrix $Q = \sum_{r=1}^R q_r Q_r$ where each Q_r is a fixed matrix of the same size as Q and our goal is to find the parameters q_r . The key to finding the parameter $Q_{\theta\theta}$ for the problem (A.1) is to parameterize the cross covariance $Q_{\theta x}$ between the observed variables x and the unmodeled unobserved variables θ . Letting x be n dimensional and letting θ be m dimensional, for each entry $(Q_{\theta x})_{ij}$ we introduce a parameter q_{in+j} which corresponds to a fixed matrix Q_{in+j} which has a 1 in the (i, j) -th position of $Q_{\theta x}$ and in the (j, i) -th position of $Q_{x\theta}$ and is zero otherwise. Notice, that whenever the matrix $Q = \sum_{r=1}^R q_r Q_r$ is formed, the sub-matrix $Q_{\theta\theta}$ will be identically zero since there are no parameter matrices with nonzero entries in $Q_{\theta\theta}$. Thus, whenever we form the matrix $Q = \sum_{r=1}^R q_r Q_r$ we must project this matrix onto the nearest symmetric positive definite covariance matrix. We do this by finding the singular value decomposition $Q = U\Lambda V^T$ and then replacing Q with $\hat{Q} = U\Lambda U^T$. With this choice of parameterization of Q , along with the method of projecting on \hat{Q} , we can apply the method of [10] to find the parameters q_r from an adaptive EnKF.