Kernel Observers: Systems-Theoretic Modeling and Inference of Spatiotemporally Evolving Processes

Authored by:

Hassan A. Kingravi
Harshal R. Maske
Girish Chowdhary

Abstract

We consider the problem of estimating the latent state of a spatiotemporally evolving continuous function using very few sensor measurements. We show that layering a dynamical systems prior over temporal evolution of weights of a kernel model is a valid approach to spatiotemporal modeling that does not necessarily require the design of complex nonstationary kernels. Furthermore, we show that such a predictive model can be utilized to determine sensing locations that guarantee that the hidden state of the phenomena can be recovered with very few measurements. We provide sufficient conditions on the number and spatial location of samples required to guarantee state recovery, and provide a lower bound on the minimum number of samples required to robustly infer the hidden states. Our approach outperforms existing methods in numerical experiments.

1 Paper Body

Modeling of large-scale stochastic phenomena with both spatial and temporal (spatiotemporal) evolution is a fundamental problem in the applied sciences and social networks. The spatial and temporal evolution in such domains is constrained by stochastic partial differential equations, whose structure and parameters may be time-varying and unknown. While modeling spatiotemporal phenomena has traditionally been the province of the field of geostatistics, it has in recent years gained more attention in the machine learning community [2]. The data-driven models developed through machine learning techniques provide a way to capture complex spatiotemporal phenomena that are not easily modeled by first-principles alone, such as stochastic partial differential equations. In the machine learning community, kernel methods represent a class of extremely well-studied and powerful methods for inference in spatial domains; in these techniques, correlations between the input variables are encoded through

a covariance kernel, and the model is formed through a linear weighted combination of the kernels [14]. In recent years, kernel methods have been applied to spatiotemporal modeling with varying degrees of success [2, 14]. Many recent techniques in spatiotemporal modeling have focused on nonstationary covariance kernel design and associated hyperparameter learning algorithms [4, 7, 12]. The main benefit of careful design of covariance kernels over approaches that simply include time as an additional input variable is that they can account for intricate spatiotemporal couplings. However, there are two key challenges with these approaches: the first is ensuring the scalability of the model to large scale phenomena, which manifests due to the fact that the hyperparameter optimization problem is not convex in general, leading to methods that are difficult to implement, susceptible to local minima, and that can become computationally intractable for large datasets. In addition to the challenge of modeling spatiotemporally varying processes, we are interested in addressing the second very important, and widely unaddressed challenge: given a predictive model of the spatiotemporal phenomena, how can the current latent state of the phenomena be estimated using as few sensor measurements as possible? This is called the monitoring problem. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Monitoring a spatiotemporal phenomenon is concerned with estimating its current state, predicting its future evolution, and inferring the initial conditions utilizing limited sensor measurements. The key challenges here manifest due to the fact that it is typically infeasible or expensive to deploy sensors at a large scale across vast spatial domains. To minimize the number of sensors deployed, a predictive data-driven model of the spatiotemporal evolution could be learned from historic datasets or through remote sensing (e.g. satellite, radar) datasets. Then, to monitor the phenomenon, the key problem would boil down to reliably and quickly estimating the evolving latent state of the phenomena utilizing measurements from very few sampling locations. In this paper, we present an alternative perspective on solving the spatiotemporal monitoring problem that brings together kernel-based modeling, systems theory, and Bayesian filtering. Our main contributions are two-fold: first, we demonstrate that spatiotemporal functional evolution can be modeled using stationary kernels with a linear dynamical systems layer on their mixing weights. In other words, the model proposed here posits differential constraints, embodied as a linear dynamical system, on the spatiotemporal evolution of a kernel based models, such as Gaussian Processes. This approach does not necessarily require the design of complex spatiotemporal kernels, and can accommodate positive-definite kernels on any domain on which it?s possible to define them, which includes non-Euclidean domains such as Riemannian manifolds, strings, graphs and images [6]. Second, we show that the model can be utilized to determine sensing locations that guarantee that the hidden states of functional evolution can be estimated using a Bayesian state-estimator with very few measurements. We provide sufficient conditions on the number and location of sensor measurements required and prove non-conservative lower bounds on the minimum number of sampling locations. The validity of the presented model and sensing techniques is corrob-

orated using synthetic and large real datasets. 1.1

Related Work

There is a large body of literature on spatiotemporal modeling in geostatistics where specific processdependent kernels can be used [17, 2]. From the machine learning perspective, a naive approach is to utilize both spatial and temporal variables as inputs to a Mercer kernel [10]. However, this technique leads to an ever-growing kernel dictionary. Furthermore, constraining the dictionary size or utilizing a moving window will occlude learning of long-term patterns. Periodic or nonstationary covariance functions and nonlinear transformations have been proposed to address this issue [7, 14]. Work focusing on nonseparable and nonstationary covariance kernels seeks to design kernels optimized for environment-specific dynamics, and to tune their hyperparameters in local regions of the input space. Seminal work in [5] proposes a process convolution approach for space-time modeling. This model captures nonstationary structure by allowing the convolution kernel to vary across the input space. This approach can be extended to a class of nonstationary covariance functions, thereby allowing the use of a Gaussian process (GP) framework, as shown in [9]. However, since this model?s hyperparameters are inferred using MCMC integration, its application has been limited to smaller datasets. To overcome this limitation, [12] proposes to use the mean estimates of a second isotropic GP (defined over latent length scales) to parameterize the nonstationary covariances. Finally, [4] considers nonistropic variation across different dimension of input space for the second GP as opposed to isotropic variation by [12]. Issues with this line of approach include the nonconvexity of the hyperparameter optimization problem and the fact that selection of an appropriate nonstationary covariance function for the task at hand is a nontrivial design decision (as noted in [16]). Apart from directly modeling the covariance function using additional latent GPs, there exist several other approaches for specifying nonstationary GP models. One approach maps the nonstationary spatial process into a latent space, in which the problem becomes approximately stationary [15]. Along similar lines, [11] extends the input space by adding latent variables, which allows the model to capture nonstationarity in original space. Both these approaches require MCMC sampling for inference, and as such are subject to the limitations mentioned in the preceding paragraph. A geostatistics approach that finds dynamical transition models on the linear combination of weights of a parameterized model [2, 8] is advantageous when the spatial and temporal dynamics are hierarchically separated, leading to a convex learning problem. As a result, complex nonstationary kernels are often not necessary (although they can be accommodated). The approach presented in this paper aligns closely with this vein of work. A system theoretic study of this viewpoint enables the fundamental contributions of the paper, which are 1) allowing for inference on more general domains with a larger class of basis functions than those typically considered in the geostatistics community, 2

0.9

0.9

0.8

0.8
0.7
0.7
0.6
0.6
0.5
0.5
0.4
0.4
0.3
0.3
0.2
0.2
0.1
0.1

(a) 1-shaded (Def. 1)(b) 2-shaded (Eq. (4))

Figure 1: Two types of Hilbert space evolutions.

Figure 2: Shaded observation matrices for dictio-

Left: discrete switches in RKHS H; Right: smooth evolution in H.

nary of atoms.

and 2) quantifying the minimum number of measurements required to estimate the state of functional evolution. It should be noted that the contribution of the paper concerning sensor placement is to provide sufficient conditions for monitoring rather than optimization of the placement locations, hence a comparison with these approaches is not considered in the experiments.

2

Kernel Observers

This section outlines our modeling framework and presents theoretical results associated with the number of sampling locations required for monitoring functional evolution. 2.1

Problem Formulation

We focus on predictive inference of a time-varying stochastic process, whose mean f evolves temporally as f? +1 ? F(f? , ??), where F is a distribution varying with time ? and exogenous inputs ?. Our approach builds on the fact that in several cases, temporal evolution can be hierarchically separated from spatial functional evolution. A classical and quite general example of this is the abstract evolution equation (AEO), which can be defined as the evolution of a function u embedded in a Banach space B: u(t) ? = Lu(t), subject to u(0) = u0 , and L : B ? B determines spatiotemporal transitions of u ? B [1]. This model of spatiotemporal evolution is very general (AEOs, for example, model many PDEs), but working in Banach spaces can be computationally taxing. A simple way to make the approach computationally realizable is to place restrictions on B: in particular, we restrict the sequence f? to lie in a reproducing kernel Hilbert space (RKHS), the theory of which provides powerful tools for generating flexible classes of functions with relative ease [14]. In a kernel-based model, k : ? ? ? ? R is a positive-definite Mercer kernel on a domain ? that

models the covariance between any two points in the input space, and implies the existence of a smooth map ? : ? ? H, where H is an RKHS with the property k(x, y) = h?(x), ?(y)iH . The key insight behind the proposed model is that spatiotemporal evolution in the input domain corresponds to temporal evolution of the mixing weights of a kernel model alone in the functional domain. Therefore, f? can be modeled by tracing the evolution of its mean embedded in a RKHS using switched ordinary differential equations (ODE) when the evolution is continuous, or switched difference equations when it is discrete (Figure 1). The advantage of this approach is that it allows us to utilize powerful ideas from systems theory for deriving necessary and sufficient conditions for spatiotemporal monitoring. In this paper, we restrict our attention to the class of functional evolutions F defined by linear Markovian transitions in an RKHS. While extension to the nonlinear case is possible (and non-trivial), it is not pursued in this paper to help ease the exposition of the key ideas. The class of linear transitions in RKHS is rich enough to model many real-world datasets, as suggested by our experiments. Let y? ? RN be the measurements of the function available from N sensors at time ? , A : H ? H be a linear transition operator in the RKHS H, and K : H ? RN be a linear measurement operator. The model for the functional evolution and measurement studied in this paper is: f? +1 = Af? + ?? ,

y? = Kf? + ?? ,

(1)

where ?? is a zero-mean stochastic process in H, and ?? is a Wiener process in RN . Classical treatments of kernel methods emphasize that for most kernels, the feature map ? is unknown, and possibly infinite-dimensional; this forces practioners to work in the dual space of H, whose dimensionality is the number of samples in the dataset being modeled. This conventional wisdom precludes the use of kernel methods for most tasks involving modern datasets, which may have 3

millions and sometimes billions of samples [13]. An alternative is to work with a feature map b b with the property that for every ?(x) := [?b1 (x) ??? ?bM (x)] to an approximate feature space H, b b b element f ? H, ?f ? H and an ¿ 0 s.t. kf ? f k ¡ for an appropriate function norm. A few such approximations are listed below. Dictionary of atoms Let ? be compact. Given points C = {c1 , . . . , cM }, ci ? ?, we have a dictionary of atoms F C = {?(c1), ? ? ? , ?(cM)}, ?(ci) ? H, the span of which is a strict subspace b of the RKHS H generated by the kernel. Here, H ?bi (x) := h?(x), ?(ci)iH = k(x, ci)

(2) M ?M

Low-rank approximations Let ? be compact, let C = {c1 , . . . , cM }, ci ? ?, and let K ? R , Kij := k(ci , cj) be the Gram matrix computed from C. This matrix can be diagonalized to compute bi , ?bi (x)) of the eigenvalues and eigenfunctions (?i , ?i (x)) of the kernel [18]. approximations (? p bi ?bi (x). These spectral quantities can then be used to compute ?bi (x) := ? 2

2

Random Fourier features Let ? ? Rn be compact, and let k(x, y) = e?kx?yk /2? be the Gaussian RBF kernel. Then random Fourier features approximate

the kernel feature map as ?b? : b where ? is a sample from the Fourier transform of k(x, y), with the property that k(x, y) = ? ? H, b E? [h?? (x), ?b? (y)iHb] [13]. In this case, if V ? RM/2?n is a random matrix representing the sample ?, then ?bi (x) := [?1M sin([V x]i), ?1M cos([V x]i)]. Similar approximations exist for other radially symmetric and dot product kernels. b ? H. b In the approximate space case, we replace the transition operator A : H ? H in (1) by Ab : H This approximate regime, which trades off the flexibility of a truly nonparametric approach for computational realizability, still allows for the representation of rich phenomena, as will be seen in the sequel. The finite-dimensional evolution equations approximating (1) in dual form are b ? + ?? , y? = Kw? + ?? , w? +1 = Aw (3) b ? RM ?M , K ? RN ?M , the vectors w? ? RM , and where we have where we have matrices A b counterparts. Here K is the matrix whose slightly abused notation to let ?? and ?? denote their H b b b rows are of the form K(i) = ?(xi) = [?1 (xi) ?2 (xi) ??? ?bM (xi)]. In systems-theoretic language, each row of K corresponds to a measurement at a particular location, and the matrix itself acts as

a measurement operator. We define the generalized observability matrix [20] as O? =

b?1 KA ??? b?L KA

where ? = {?1 , . . . , ?L } are the set of instances ?i when we apply the operator K. A linear system is said to be observable if O? has full column rank (i.e. RankO? = M) for ? = {0, 1, . . . , M ? 1} [20]. Observability guarantees two critical facts: firstly, it guarantees that the state w0 can be recovered exactly from a finite series of measurements {y?1 , y?2 , . . . , y?L }; in particular, defining

T y? = y?T1 , y?T2 , ? ? ? , y?LT , we have that y? = O? w0 . Secondly, it guarantees that a feedback based observer can be designed such that the estimate of w? , denoted by w b? , converges exponentially fast b is available: while we to w? in the limit of samples. Note that all our theoretical results assume A perform system identification in the experiments (Section 3.3), it is not the focus of the paper. We are now in a position to formally state the spatiotemporal modeling and inference problem considered: given a spatiotemporally evolving system modeled using (3), choose a set of N sensing locations such that even with N M , the functional evolution of the spatiotemporal model can be estimated (which corresponds to monitoring) and can be predicted robustly (which corresponds to Bayesian filtering). Our approach to solve this problem relies on the design of the measurement b is observable: any Bayesian state estimator (e.g. a Kalman filter) operator K so that the pair (K, A) utilizing this pair is denoted as a kernel observer 1 . We will leverage the spectral decomposition of b for this task (see ??? in supplementary for details on spectral decomposition). A 2.2 Main Results In this section, we prove results concerning the observability of spatiotemporally varying functions modeled by the functional evolution and measurement equations (3) formulated in Section 2.1. In 1

In the case where no measurements are taken, for the sake of consistency, we denote the state estimator as an autonomous kernel observer, despite this being something of an oxymoron.

4

particular, observability of the system states implies that we can recover the current state of the spatiotemporally varying function using a small number of sampling locations N , which allows us to 1) track the function, and 2) predict its evolution forward in time. We work with the approximation b ? H: given M basis functions, this implies that the dual space of H b is RM . Proposition 1 shows H b that if A has a full-rank Jordan decomposition, the observation matrix K meeting a condition called shadedness (Definition 1) is sufficient for the system to be observable. Proposition 2 provides a b lower bound on the number of sampling locations required for observability which holds for any A. e Proposition 3 constructively shows the existence of an abstract measurement map K achieving this lower bound. Finally, since the measurement map does not have the structure of a kernel matrix, b is in Theorem 1. Proofs of all a slightly weaker sufficient condition for the observability of any A claims are in the supplementary material. Definition 1. (Shaded Observation Matrix) Given k : ? ? ? ? R positive-definite on a domain ?, b and let {?b1 (x), . . . , ?bM (x)} be the set of bases generating an approximate feature map ?b : ? ? H, N ?M b let X = {x1 , . . . , xN }, xi ? ?. Let K ? R be the observation matrix, where Kij := ?j (xi). (i) (i) (i) b b For each row K(i) := [?1 (xi) ??? ?M (xi)], define the set I(i) := {?1 , ?2 , . . . , ?Mi } to be the indices S in the observation matrix row i which are nonzero. Then if i?{1,...,N } I (i) = {1, 2, . . . , M }, we denote K as a shaded observation matrix (see Figure 2a). This definition seems quite abstract, so the following remark considers a more concrete example. Remark 1. Let ?b be generated by the dictionary given by C = {c1 , . . . , cM }, ci ? ?. Note that since ?bj (xi) = h?(xi), ?(cj)iH = k(xi , cj), K is the kernel matrix between X and C. For the kernel matrix to be shaded thus implies that there does not exist an atom ?(cj) such that the projections h?(xi), ?(cj)iH vanish for all xi , 1 ? i ? N . Intuitively, the shadedness property requires that the sensor locations xi are privy to information propagating from every cj . As an example, note that, in principle, for the Gaussian kernel, a single row generates a shaded kernel matrix2 . Proposition 1. Given k : ? ? ? ? R positive-definite on a domain ?, let {?b1 (x), . . . , ?bM (x)} be b and let X = {x1 , . . . , xN }, the set of bases generating an approximate feature map ?b : ? ? H, b xi ? ?. Consider the discrete linear system on H given by the evolution and measurement equations b ? RM ?M of the form A b = P ?P ?1 (3). Suppose that a full-rank Jordan decomposition of A exists, where ? = [?1 ??? ?O], and there are no repeated eigenvalues. Then, given a set of time instances ? = {?1 , ?2 , . . . , ?L }, and a set of sampling locations X = {x1 , . . . , xN }, the system (3) is observable if the observation matrix Kij is shaded according to Definition 1, ? has distinct values, and —?— ? M . When the eigenvalues of the system matrix are repeated, it is not enough for K to be shaded. In b to the next proposition, we take a geometric approach and utilize the rational canonical form of A obtain a lower bound on the number of sampling locations required. Let r be the number of unique b and let ?? denote the geometric multiplicity of eigenvalue ?i . Then the cyclic eigenvalues of A, i b is defined as ‘ = max1?i?r ?? [19] (see supplementary section ?? for details).

index of A i Proposition 2. Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan blocks [?1 ??? ?O] may have repeated eigenvalues (i.e. ??i and ?j s.t. ?i = ?j). Then there exist kernels k(x, y) such that the lower bound ‘ on the number of sampling locations N is given by the b cyclic index of A. Section ?? in supplementary gives a concrete example to build intuition regarding this lower bound. e corresponding to the lower bound ‘. We now show how to construct a matrix K Proposition 3. Given the conditions stated in Proposition 2, it is possible to construct a measurement e ? R‘?M for the system given by (3), such that the pair (K, e A) b is observable. map K The construction provided in the proof of Proposition 3 is utilized in Algorithm 1, which uses b to generate a series of vectors vi ? RM , whose iterations the rational canonical structure of A 2

However, in this case, the matrix can have many entries that are extremely close to zero, and will probably be very ill-conditioned.

5

e Algorithm 1 Measurement Map K b ? RM ?M Input: A bT Q. Set C0 := C, and M0 := M . Compute rational canonical form, such that C = Q?1 A for i = 1 to ‘ do Obtain MP ?i (?) of Ci?1 . This returns associated indices J (i) ? {1, 2, . . . , Mi?1 }. Construct vector vi ? RM such that ?vi (?) = ?i (?) . Use indices {1, 2, . . . , Mi?1 } J (i) to select matrix Ci . Set Mi := —{1, 2, . . . , Mi?1 } J (i) — end for ? = [v1T , v2T , ..., v‘T]T Compute K e = KQ ? ?1 Output: K

bm1 ?1 v1 , . . . , v‘ , . . . , A bm‘ ?1 v‘ } generate a basis for RM . Unfortunately, the measurement {v1 , . . . , A e map K, being an abstract construction unrelated to the kernel, does not directly select X . We will show how to use the measurement map to guide a search for X in Remark ??. For now, we state a sufficient condition for observability of a general system. Theorem 1. Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan b Define ? ? ? ?O] may have repeated eigenvalues. Let ‘ be the cyclic index of A. blocks [?1 K = [K (1)T

??? K (‘)

T

T

]

(4)

as the ‘-shaded matrix which consists of ‘ shaded matrices with the property that any subset of ‘ columns in the matrix are linearly independent from each other. Then system (3) is observable if ? has distinct values, and —?— ? M . While Theorem 1 is a quite general result, the condition that any ‘ columns of K be linearly independent is a very stringent condition. One scenario where this condition can be met with b minimal measurements is in the case when the feature map ?(x) is generated by a dictionary of atoms with the Gaussian RBF kernel evaluated at sampling locations {x1 , . . . , xN } according to (2), where xi ? ? ? Rd , and xi are sampled from a non-degenerate probability distribution on ? such as the uniform distribution. For a semi-deterministic approach, when the dynamics matrix b is block-diagonal, a simple heuristic is given in Remark ??

in the supplementary. Note that in A b needs to be inferred from measurements of the process f? . If no assumptions practice the matrix A b at least M sensors are required for the system identification phase. Future work will are placed on A, study the precise conditions under which system identification is possible with less than M sensors. Finally, computing the Jordan and rational canonical forms can be computationally expensive: see the supplementary for more details. We note that the crucial step in our approach is computing the cyclic index, which gives us the minimum number of sensors that need to be deployed, the computational complexity of which is O(M 3). Computation of the canonical forms is required in the case we need to strictly realize the lower bound on the number of sensors.

3 3.1

Experimental Results Sampling Locations for Synthetic Data Sets

The goal of this experiment is to investigate the dependency of the observability of system (3) on the shaded observation matrix and the lower bound presented in Proposition 2. The domain is fixed on the interval ? = [0, 2?]. First, we pick sets of points C (?) = {c1 , . . . , cM? }, cj ? ?, M = 50, and construct a dynamics matrix A = ? ? RM ?M , with cyclic index 5. We pick the RBF kernel 2 2 k(x, y) = e?kx?yk /2? , ? = 0.02. Generating samples X = {x1 , . . . , xN }, xi ? ? randomly, we compute the ‘-shaded property and observability for this system. Figure 3a shows how shadedness is a necessary condition for observability, validating Proposition 1: the slight gap between shadedness and observability here can be explained due to numerical issues in computing the rank of O? . Next, we again pick M = 50, but for a system with a cyclic index ‘ = 18. We constructed the measurement e using Algorithm 1, and the heuristic in Remark ?? (Algorithm 2 in the supplementary) as map K well as random sampling to generate the sampling locations X . These results are presented in Figure 3b. The plot for random sampling has been averaged over 100 runs. It is evident from the plot that 6

observability cannot be achieved for a number of samples N ¡ ‘. Clearly, the heuristic presented outperforms random sampling; note however, that our intent is not to compare the heuristic against random sampling, but to show that the lower bound ‘ provides decisive guidelines for selecting the number of samples while using the computationally efficient random approach. 3.2

Comparison With Nonstationary Kernel Methods on Real-World Data

We use two real-world datasets to evaluate and compare the kernel observer with the two different lines of approach for non-stationary kernels discussed in Section 1.1. For the Process Convolution with Local Smoothing Kernel (PCLSK) and Latent Extension of Input Space (LEIS) approaches, we compare with NOSTILL-GP [4] and [11] respectively, on the Intel Berkeley and Irish Wind datasets. Model inference for the kernel observer involved three steps: 1) picking the Gaussian RBF kernel 2 2 k(x, y) = e?kx?yk /2? , a search for the ideal ? is performed for a sparse Gaussian Process model (with a fixed basis vector set C selected using the method in [3]. For the data set discussed in this section, the number of basis vectors were equal to the number of sensing locations in the training set, with the domain for input set defined over R2 ; 2)

having obtained ?, Gaussian process inference is used to generate weight vectors for each time-step in the training set, resulting in the sequence b (Algorithm 3 in w? , ? ? {1, . . . , T }; 3) matrix least-squares is applied to this sequence to infer A b is used to propagate the state w? the supplementary). For prediction in the autonomous setup, A forward to make predictions with no feedback, and in the observer setup, a Kalman filter (Algorithm 4 in the supplementary) with N determined using Proposition 2, and locations picked randomly, is used to propagate w? forward to make predictions. We also compare with a baseline GP (denoted by ?original GP?), which is the sparse GP model trained using all of the available data. Our first dataset, the Intel Berkeley research lab temperature data, consists of 50 wireless temperature sensors in indoor laboratory region spanning 40.5 meters in length and 31 meters in width3 . Training data consists of temperature data on March 6th 2004 at intervals of 20 minutes (beginning 00:20 hrs) which totals to 72 timesteps. Testing is performed over another 72 timesteps beginning 12:20 hrs of the same day. Out of 50 locations, we uniformly selected 25 locations each for training and testing purposes. Results of the prediction error are shown in box-plot form in Figure 4a and as a b time-series in Figure 4b, note that ?Auto? refers to autonomous set up. Here, the cyclic index of A was determined to be 2, so N was set to 2 for the kernel observer with feedback. Note that here, even the autonomous kernel observer outperforms PCLSK and LEIS overall, and the kernel observer with feedback N = 2 does so significantly, which is why we did not include results with N ¿ 2. The second dataset is the Irish wind dataset, consisting of daily average wind speed data collected from year 1961 to 1978 at 12 meteorological stations in the Republic of Ireland4 . The prediction b error is in box-plot form in Figure 5a and as a time-series in Figure 5b. Again, the cyclic index of A was determined to be 2. In this case, the autonomous kernel observer?s performance is comparable to PCLSK and LEIS, while the kernel observer with feedback with N = 2 again outperforms all other methods. Table ?? in the supplementary reports the total training and prediction times associated with PCLSK, LEIS, and the kernel observer. We observed that 1) the kernel observer is an order of magnitude faster, and 2) even for small sets, competing methods did not scale well. 3.3 Prediction of Global Ocean Surface Temperature We analyzed the feasibility of our approach on a large dataset from the National Oceanographic Data Center: the 4 km AVHRR Pathfinder project, which is a satellite monitoring global ocean surface temperature (Fig. 6a). This dataset is challenging, with measurements at over 37 million possible coordinates, but with only around 3-4 million measurements available per day, leading to a lot of missing data. The goal was to learn the day and night temperature models on data from the year 2011, and to monitor thereafter for 2012. Success in monitoring would demonstrate two things: 1) the modeling process can capture spatiotemporal trends that generalize across years, and 2) the observer framework allows us to infer the state using a number of measurements that are an order of magnitude fewer than available. Note that due to the size of the dataset and the high computational requirements of the nonstationary kernel methods, a comparison with them was not pursued. To build the autonomous kernel observer and general kernel observer models, we

followed the same procedure outlined in Section 3.2, but with C = {c1 , . . . ,
cM }, cj ? R2 , —C— = 300. Cyclic 3 4

http://db.csail.mit.edu/labdata/labdata.html http://lib.stat.cmu.edu/datasets/wind.desc

7

0.4 0.2 0 20

30

40

0.8 0.6 0.4 0.2

10

Samples

20

30

40

295 6

290 285

4

280 2 275 0

(a) AVHHR estimate

4 2

1 0 0

20

40

14 12 10 8 6 4

20

18

18

16 14 12 10 8 6 4

(b) Error (time-series)

Original

Auto

N=250

N=500

40 Timesteps

60

80

(b) Error (time-series)

20

Original Autonomous KO (N = 1000)

16

20

2 30

LEIS

2

Original Autonomous KO (N = 1000)

18 16 14 12 10 8 6 4 2

Jul 11 Sep 11 Nov 11 Jan 12 Mar 12 May 12 Timesteps

(b) Error-day (time-series) (c) Error-night (time-series) RMS Error in Temperature (K)
RMS Error in Temperature (K)
6
20 Timesteps
Observer PCLSK
3
Jul 11 Sep 11 Nov 11 Jan 12 Mar 12 May 12 Timesteps
Original Autonomous Kernel Observer N = 2 PCLSK LEIS
10
Auto
18
LEIS
8
0 0
1
4
2
270
(a) Error (boxplot) RMS Error in Wind Speed (knots)
RMS Error in Temperature (K)
RMS Error in Wind Speed (knots)
300
10
2 1.5
20
305
8
12
3 2.5
Figure 4: Comparison of kernel observer to PCLSK and LEIS methods on Intel dataset.
310 10
Observer PCLSK
3.5
(a) Error (boxplot)
Figure 3: Kernel observability results.
Auto
Original Autonomous Kernel Observer N = 2 PCLSK LEIS
5
50
Samples
(a) Shaded vs. observability (b) Heuristic vs. random
Original
4
Original

0
50
4.5
16 14 12 10 8 6
(d) Error-day (boxplot)
5 4 3 2 1
4 2
N=1000
6 Training Time (seconds)
10
1
5 RMS Error in Temperature (oC)
0.6
Random Heuristic
1.2
RMS Error in Temperature (K)
0.8
RMS Error in Temperature (oC)
1
Percentage Observable
Percentage Observable
6
obs. shaded
1.2
0 Original
Auto
N=250
N=500
N=1000
(e) Error-night (boxplot)
Original
Auto
N=250
N=500
N=1000
(f) Estimation time (day)

Figure 5: Irish Wind Figure 6: Performance of the kernel observer over AVVHR satellite 2011-12 data with different numbers of observation locations. b was determined to be 250 and hence the Kalman filter for kernel observer model using index of A N ? {250, 500, 1000} at random locations was utilized to track the system state given a random initial condition w0 . As a fair baseline, the observers are compared to training a sparse GP model (labeled ?original?) on approximately 400, 000 measurements per day. Figures 6b and 6c compare the autonomous and feedback approach with 1, 000 samples to the baseline GP; here, it can be seen that the autonomous does well in the beginning, but then incurs an unacceptable amount of error when the time series goes into

2012, i.e. where the model has not seen any training data, whereas KO does well throughout. Figures 6d and 6e show a comparison of the RMS error of estimated values from the real data. This figure shows the trend of the observer getting better state estimates as a function of the number of sensing locations N 5 . Finally, the prediction time of KO is much less than retraining the model every time step, as shown in Figure 6f.

4

Conclusions

This paper presented a new approach to the problem of monitoring complex spatiotemporally evolving phenomena with limited sensors. Unlike most Neural Network or Kernel based models, the presented approach inherently incorporates differential constraints on the spatiotemporal evolution of the mixing weights of a kernel model. In addition to providing an elegant and efficient model, the main benefit of the inclusion of the differential constraint in the model synthesis is that it allowed the derivation of fundamental results concerning the minimum number of sampling locations required, and the identification of correlations in the spatiotemporal evolution, by building upon the rich literature in systems theory. These results are non-conservative, and as such provide direct guidance in ensuring robust real-world predictive inference with distributed sensor networks. Acknowledgment This work was supported by AFOSR grant #FA9550-15-1-0146. 5

Note that we checked the performance of training a GP with only 1, 000 samples as a control, but the average error was about 10 Kelvins, i.e. much worse than KO.

8

2 References

[1] Haim Brezis. Functional analysis, Sobolev spaces and partial differential equations. Springer Science & Business Media, 2010. [2] Noel Cressie and Christopher K Wikle. Statistics for spatio-temporal data. John Wiley & Sons, 2011. [3] Lehel Csat? and Manfred Opper. Sparse on-line gaussian processes. Neural computation, 14(3):641?668, 2002. [4] Sahil Garg, Amarjeet Singh, and Fabio Ramos. Learning non-stationary space-time models for environmental monitoring. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada., 2012. [5] David Higdon. A process-convolution approach to modelling temperatures in the north atlantic ocean. Environmental and Ecological Statistics, 5(2):173?190, 1998. [6] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015. [7] Chunsheng Ma. Nonstationary covariance functions that model space?time interactions. Statistics & Probability Letters, 61(4):411?419, 2003. [8] Kanti V Mardia, Colin Goodall, Edwin J Redfern, and Francisco J Alonso. The kriged kalman filter. Test, 7(2):217?282, 1998. [9] C Paciorek

and M Schervish. Nonstationary covariance functions for gaussian process regression. Advances in neural information processing systems, 16:273?280, 2004. [10] Fernando P?rez-Cruz, Steven Van Vaerenbergh, Juan Jos? Murillo-Fuentes, Miguel L?zaroGredilla, and Ignacio Santamaria. Gaussian processes for nonlinear signal processing: An overview of recent advances. Signal Processing Magazine, IEEE, 30(4):40?50, 2013. [11] Tobias Pfingsten, Malte Kuss, and Carl Edward Rasmussen. Nonstationary gaussian process regression using a latent extension of the input space, 2006. [12] Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In Machine learning and knowledge discovery in databases, pages 204?219. Springer, 2008. [13] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In NIPS, pages 1177?1184, 2007. [14] Carl E. Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. The MIT Press, December 2005. [15] Alexandra M Schmidt and Anthony O?Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 65(3):743?758, 2003. [16] Amarjeet Singh, Fabio Ramos, H Durrant-Whyte, and William J Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 5490?5497. IEEE, 2010. [17] Christopher K Wikle. A kernel-based spectral model for non-gaussian spatio-temporal processes. Statistical Modelling, 2(4):299?314, 2002. [18] Christopher Williams and Matthias Seeger. Using the Nystr?m method to speed up kernel machines. In NIPS, pages 682?688, 2001. [19] W Murray Wonham. Linear multivariable control. Springer, 1974. [20] Kemin Zhou, John C. Doyle, and Keith Glover. Robust and Optimal Control. Prentice Hall, Upper Saddle River, NJ, 1996. 9