

---

# Learning Localized Spatio-Temporal Models From Streaming Data

---

Muhammad Osama<sup>1</sup> Dave Zachariah<sup>1</sup> Thomas Schön<sup>1</sup>

## Abstract

We address the problem of predicting spatio-temporal processes with temporal patterns that vary across spatial regions, when data is obtained as a stream. That is, when the training dataset is augmented sequentially. Specifically, we develop a localized spatio-temporal covariance model of the process that can capture spatially varying temporal periodicities in the data. We then apply a covariance-fitting methodology to learn the model parameters which yields a predictor that can be updated sequentially with each new data point. The proposed method is evaluated using both synthetic and real climate data which demonstrate its ability to accurately predict data missing in spatial regions over time.

## 1. Introduction

Many real-world processes of interest, ranging from climate variables to brain signals, are spatio-temporal in nature, cf. Cressie & Wikle (2011). That is, they can be described as a random quantity that varies over some fixed spatial and temporal domain. Suppose we obtain  $n$  training points from a real-valued spatio-temporal process,

$$\mathcal{D}_n = \{ (s_1, t_1, y_1), \dots, (s_n, t_n, y_n) \},$$

where  $y_i$  denotes the quantity of interest observed at the  $i^{\text{th}}$  training point, with spatial coordinate  $s_i$  and time  $t_i$ . For notational convenience, let  $(s, t, y)$  denote an unobserved test point in space-time where  $y$  is unknown. Then a common goal is to predict  $y$  in unobserved space-time regions  $(s, t)$  using  $\mathcal{D}_n$ . Specifically, certain spatial regions may have limited data coverage over extended periods of time, as illustrated in Figure 1. In real-world applications,  $\mathcal{D}_n$  need not be gathered in a single batch but obtained in parts over time from various sensors, stations, satellites,

---

<sup>\*</sup>Equal contribution <sup>1</sup>Uppsala University, Sweden. Correspondence to: Muhammad Osama <muhammad.osama@it.uu.se>, Dave Zachariah <dave.zachariah@it.uu.se>.

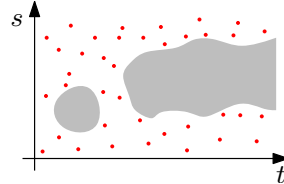


Figure 1. Example of training points (dots) in  $\mathcal{D}_n$  over a bounded space-time domain  $\mathcal{S} \times \mathcal{T}$  with one spatial dimension. Note that the sampling pattern may be irregular and that it is not possible to provide complete spatial coverage at all times. A typical problem is to predict the process in unobserved regions (shaded).

etc. That is, the dataset is augmented sequentially, i.e.,  $n = 1, 2, \dots, N$ . In these streaming data scenarios, we are interested in continuous refinement of the prediction of  $y$  at  $(s, t)$  as new data is augmented into  $\mathcal{D}_{n+1}$ .

The unknown data-generating process is often assumed to belong to a class of data models indexed by a parameter  $\theta$ . Each model  $\theta$  in the class yields a predictor  $\hat{y}_\theta(s, t)$  of  $y$  at test point  $(s, t)$ . A specific set of model parameters  $\hat{\theta}$  is learned using  $\mathcal{D}_n$ . Examples of commonly used model classes include Gaussian Processes (GP) (Rasmussen & Williams, 2006), spatio-temporal random effects models (Cressie et al., 2010), dynamic factor analysis models (Lopes et al., 2008; Fox & Dunson, 2015), spatial random effect models extended to incorporate time as an additional dimension (Zammit-Mangion & Cressie, 2017) (cf. related work section below). For many spatio-temporal applications, the model class should be capable of expressing temporal patterns that change across different spatial regions. Moreover, for streaming data scenarios, the learned parameter  $\hat{\theta}$  and the resulting predictor  $\hat{y}_{\hat{\theta}}(s, t)$  should be updated in a sequential manner.

Our contribution in this paper is two-fold:

- we develop a non-stationary, localized covariance model capable of capturing temporal patterns that change across space, as illustrated in Figure 2 below.
- we show how to sequentially learn the covariance model parameters and update the predictor from streaming spatio-temporal data, with a runtime that is linear in  $n$ .

In Section 2, we relate our work to already existing approaches and introduce a commonly used model class in Section 3. In Section 4 we develop a localized spatio-temporal covariance model to be used in conjunction with a covariance-fitting learning approach. Finally, the proposed method is evaluated using synthetic and real climate data in Sections 5 and 6, respectively.

**Notation:**  $\text{col}\{\mathbf{s}_1, \mathbf{s}_2\}$  stacks both elements into a single column vector.  $\otimes$ ,  $\delta(\cdot)$ ,  $\|\cdot\|_{\mathbf{W}}$  and  $\dagger$  denote the Kronecker product, Kronecker delta function, weighted  $\ell_2$ -norm and Moore-Penrose inverse, respectively. Finally, the sample mean is denoted by  $\widehat{\mathbf{E}}[\mathbf{s}_i] = \frac{1}{n} \sum_{i=1}^n \mathbf{s}_i$ .

## 2. Related work

A popular model class is the family of GPs, specified by a mean and covariance function (Rasmussen & Williams, 2006). This approach is computationally prohibitive in its basic form since both learning the model parameters  $\theta$  and implementing the predictor  $\widehat{y}_\theta(\mathbf{s}, t)$  requires a runtime on the order of  $\mathcal{O}(N^3)$ , where  $N$  is typically large in spatio-temporal applications. The predictor implementation can be approximated using various techniques. One popular approach is to approximate the training data using  $m \ll N$  inducing points which reduces the runtime to  $\mathcal{O}(m^2N)$  (Quiñonero-Candela & Rasmussen, 2005; Bijl et al., 2015). Moreover, by assuming Kronecker covariance functions it is possible to obtain even shorter runtimes by utilizing the Kronecker structure of the GP covariance matrix (Saatçi, 2012). If the model class is restricted to stationary covariance functions, the runtimes can be reduced further, cf. (Saatçi, 2012; Wilson et al., 2014). In the space-time domain, such models are also equivalent to dynamical system models so that  $\widehat{y}_\theta(\mathbf{s}, t)$  can be approximated using a basis expansion and implemented by a Kalman smoother (Särkkä et al., 2013). In the above cases, however,  $\theta$  and  $\widehat{y}_\theta(\mathbf{s}, t)$  are not updated jointly when obtaining streaming data.

The restriction to stationary covariance models is, moreover, not always adequate to capture temporal patterns that differ across spatial regions. This modeling limitation is addressed by Cressie et al. (2010), where a discrete-time model class is partially specified using a spatial basis function expansion with time-varying expansion coefficients. These are modeled as a first-order vector auto-regressive process. The coefficients thus determine a spatial pattern of the process that evolves at each discrete time-instant. This model class can capture patterns localized to specific regions in space, unlike stationary covariance models. The predictor  $\widehat{y}_\theta(\mathbf{s}, t)$  can be viewed as a spatial fixed-rank kriging method that is updated via a Kalman filter and thus applicable to streaming data (cf. Cressie & Johannesson (2008)). The model parameter  $\theta$ , however, is learned us-

ing a moment-fitting approach and operates on batch rather than streaming datasets. Other work using dynamic factor analysis models (Lopes et al., 2008; Fox & Dunson, 2015) similarly allow for time-varying coefficients but with more flexible data-adaptive basis. However, they are implemented using Markov Chain Monte Carlo methods which are computationally prohibitive for the scenarios considered herein.

Moreover, a first-order auto-regressive structure may not accurately capture more complex temporal patterns observed in real spatio-temporal processes. The approach taken by Zammit-Mangion & Cressie (2017) circumvents this limitation using basis functions that are localized in both space and time. Time locality cannot, however, capture periodic patterns or trends necessary for interpolation over longer periods. The model parameters are learned using an expectation-maximization method which is not readily applicable to streaming data scenarios.

## 3. Spatio-temporal model class

We begin by defining the data vector  $\mathbf{y} = \text{col}\{y_1, y_2, \dots, y_n\}$  obtained from  $\mathcal{D}_n$ . For the test point  $(\mathbf{s}, t)$ , we consider the unbiased predictor of  $y$  as a linear combination of the data (Stein, 2012):

$$\widehat{y}(\mathbf{s}, t) = \boldsymbol{\lambda}^\top(\mathbf{s}, t)\mathbf{y}, \quad (1)$$

where  $\boldsymbol{\lambda}^\top(\mathbf{s}, t)$  is a vector of  $n$  weights which naturally depend on the test point  $(\mathbf{s}, t)$ . The weight vector is defined as the minimizer of the conditional mean square prediction error. That is,

$$\boldsymbol{\lambda}(\mathbf{s}, t) \triangleq \arg \min_{\boldsymbol{\lambda}} \mathbb{E} \left[ (y - \boldsymbol{\lambda}^\top \mathbf{y})^2 \mid \mathbf{s}, t \right]. \quad (2)$$

Since the conditional error is determined by the unknown distribution  $p(y, \mathbf{y} \mid \mathbf{s}, t, \mathbf{s}_1, t_1, \dots, \mathbf{s}_n, t_n)$ , we specify a class of data-generating models, using only the mean and covariance (Cressie & Wikle, 2011):

$$\begin{cases} \mathbb{E}[y] &= \mathbf{u}^\top(\mathbf{s}, t)\boldsymbol{\eta}, \\ \text{Cov}[y, y'] &= \boldsymbol{\phi}^\top(\mathbf{s}, t)\boldsymbol{\Theta}\boldsymbol{\phi}(\mathbf{s}', t') + \theta_0\delta(\mathbf{s}, \mathbf{s}')\delta(t, t'). \end{cases} \quad (3)$$

The function  $\mathbf{u}(\mathbf{s}, t)$  captures the expected trend of the entire spatio-temporal process  $y$ , and when there is no such general trend we set  $\mathbf{u}(\mathbf{s}, t) \equiv 1$ . The function  $\boldsymbol{\phi}(\mathbf{s}, t)$  captures the smoothness of the process in space-time and is of dimension  $p \times 1$ . The parameter matrix  $\boldsymbol{\Theta}$  is diagonal and specifies the relevance of each dimension of  $\boldsymbol{\phi}(\mathbf{s}, t)$  similar to the way in which automatic relevance determination is sometimes used within the GP (Tipping, 2001; Faul & Tipping, 2002). Taken together, (3) specifies a class of models, each of which is indexed by the parameters  $(\boldsymbol{\eta}, \boldsymbol{\Theta}, \theta_0)$ . The  $p+1$  covariance parameters  $(\boldsymbol{\Theta}, \theta_0)$ , which we collectively

denote by  $\boldsymbol{\theta} = \text{col}\{\theta_0, \theta_1, \dots, \theta_p\}$  for notational convenience, determine the spatio-temporal covariance structure  $\text{Cov}_\theta[y, y']$  which depends on the function  $\phi(\mathbf{s}, t)$ . In the next section, we will specify  $\phi(\mathbf{s}, t)$  to develop a suitable covariance model to capture local spatial and periodic temporal patterns.

For a given model in the class, the optimal weights (2) are given in closed form (Stein, 2012) as

$$\boldsymbol{\lambda}_\theta(\mathbf{s}, t) = \mathbf{K}^{-1} \mathbf{1} (\mathbf{1}^\top \mathbf{K}^{-1} \mathbf{1})^\dagger + \mathbf{K}^{-1} \boldsymbol{\Pi}^\perp \boldsymbol{\Phi} \boldsymbol{\Theta} \phi(\mathbf{s}, t),$$

where the subindex highlights the model parameter dependence. The quantities in  $\boldsymbol{\lambda}_\theta(\mathbf{s}, t)$  are determined by the regressor matrix

$$\boldsymbol{\Phi} = [\phi(\mathbf{s}_1, t_1) \quad \dots \quad \phi(\mathbf{s}_n, t_n)]^\top$$

and the following covariance matrix

$$\mathbf{K}_\theta = \text{Cov}[\mathbf{y}, \mathbf{y}] = \boldsymbol{\Phi} \boldsymbol{\Theta} \boldsymbol{\Phi}^\top + \theta_0 \mathbf{I} \succ \mathbf{0}, \quad (5)$$

with  $\boldsymbol{\Pi}^\perp = \mathbf{I} - \mathbf{1} (\mathbf{1}^\top \mathbf{K}^{-1} \mathbf{1})^\dagger \mathbf{1} \mathbf{K}^{-1}$  being an oblique projector onto  $\text{span}(\mathbf{1})^\perp$ .

The optimal weights are invariant to the mean parameters  $\sim \boldsymbol{\eta}$  and to uniform scaling of the  $p + 1$  covariance parameters  $\boldsymbol{\theta}$ . By learning  $\boldsymbol{\theta}$  up to an arbitrary scale factor, the predictor (1) is given by the linear combiner weights  $\boldsymbol{\lambda}_\theta(\mathbf{s}, t)$ . If we assume that the process is Gaussian, the model can be learned using the maximum likelihood framework. However, this yields neither a convex problem nor one that is readily solved in a sequential manner as  $\mathcal{D}_n$  is augmented sequentially. In the next section, we apply a convex covariance-fitting framework to learn the spatio-temporal model using streaming data.

## 4. Proposed method

Below we specify the function  $\phi(\mathbf{s}, t)$  in (3) such that the spatio-temporal covariance structure  $\text{Cov}_\theta[y, y']$  can express local spatial patterns with varying temporal periodicities as illustrated in Figure 2. Subsequently, we apply a covariance-fitting methodology for learning the model parameters such that the predictor (1) can be updated sequentially for each new observation (Zachariah et al., 2017).

### 4.1. Local-periodic space-time basis

The function  $\phi(\mathbf{s}, t)$  varies over a space-time domain  $\mathcal{S} \times \mathcal{T} \subset \mathbb{R}^{d+1}$  and its elements can be thought of as basis functions. It is formulated as a Kronecker product of a time and space bases,

$$\phi(\mathbf{s}, t) = \boldsymbol{\psi}(t) \otimes \boldsymbol{\varphi}(\mathbf{s}), \quad (6)$$

for compactness.

We begin by specifying the spatial function as

$$\boldsymbol{\varphi}(\mathbf{s}) = \boldsymbol{\varphi}_1(s_1) \otimes \dots \otimes \boldsymbol{\varphi}_d(s_d), \quad (7)$$

where the basis vector for the  $i^{\text{th}}$  spatial dimension,

$$\boldsymbol{\varphi}_i(s_i) = \text{col}\{\varphi_{i,1}(s_i), \dots, \varphi_{i,N_s}(s_i)\} \quad (8)$$

is composed of  $N_s$  localized components with a finite support  $L$ . For notational simplicity, we consider  $N_s$  and  $L$  to be same for each dimension  $i$ . Based on their computational attractiveness and local approximation properties we use a cubic spline basis (Rasmussen & Williams, 2006; Wasserman, 2006). Then (8) is given by (4), where  $c$  determines the location of a component. Figure 3a illustrates the components as a function of its spatial dimension. We place the centers  $c$  of each component uniformly across the spatial dimensions.

Using  $\boldsymbol{\varphi}(\mathbf{s})$  allows for covariance structures that are localized in space in such a way that neighbouring points have a nonnegative correlation and points far from each other have no correlation as determined by the support size  $L$ . Hence for a given  $L$ , the resulting covariance structure can capture local spatial patterns of a certain scale and can easily be extended to cover multiple scales by replacing (6) with for example

$$\phi(\mathbf{s}, t) = \boldsymbol{\psi}(t) \otimes \begin{bmatrix} \boldsymbol{\varphi}_{L_1}(\mathbf{s}) \\ \boldsymbol{\varphi}_{L_2}(\mathbf{s}) \end{bmatrix}$$

that accommodates two different support sizes  $L_1$  and  $L_2$ . The number of basis functions  $N_s$  is chosen such that adjacent localized components  $\varphi_i(s)$  have overlapping support to cater for points in between them. This requirement is

$$\boldsymbol{\varphi}(s) = \begin{cases} \frac{1}{6} f(s)^3 & \frac{(c-2)L}{4} \leq s < \frac{(c-1)L}{4} \\ \frac{-1}{2} f(s)^3 + 2f(s)^2 - 2f(s) + \frac{2}{3} & \frac{(c-1)L}{4} \leq s < \frac{Lc}{4} \\ \frac{1}{2} f(s)^3 - 4f(s)^2 + 10f(s) - \frac{22}{3} & \frac{Lc}{4} \leq s < \frac{(c+1)L}{4} \\ \frac{-1}{6} f(s)^3 + 2f(s)^2 - 8f(s) + \frac{32}{3} & \frac{(c+1)L}{4} \leq s \leq \frac{(c+2)L}{4} \\ 0 & \text{otherwise} \end{cases} \quad \text{where } f(s) = \frac{4s}{L} - c + 2 \quad (4)$$

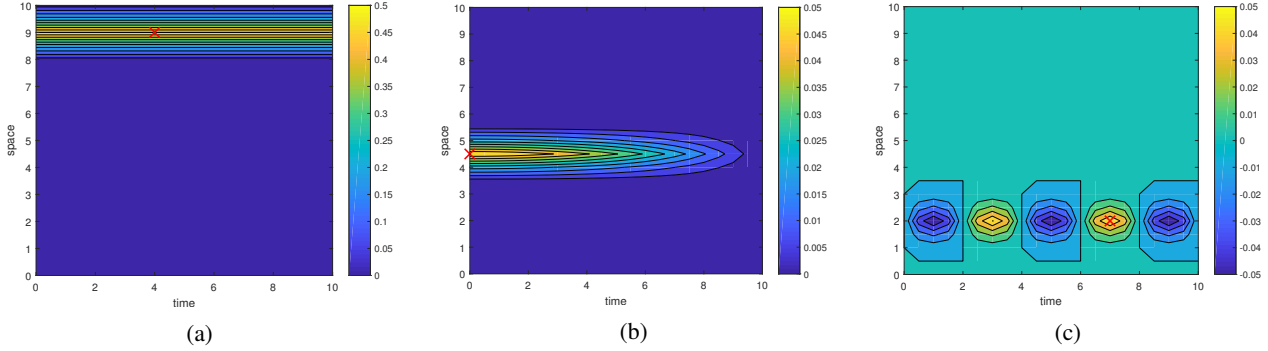


Figure 2. Example illustrations of spatio-temporal covariance structures that are possible using the proposed function  $\phi(\mathbf{s}, t)$ . Contour plots of  $\text{Cov}_\theta[y, y']$  for  $y$  at a test point  $(s, t)$  (red cross) and  $y'$  at all other coordinates  $(s', t')$  in the space-time domain. For sake of illustration, the spatial dimension is  $d = 1$ . (a) Test point  $(s, t) = (9, 4)$ . The test point is positively correlated with only neighbouring points in space and has a constant covariance with points across time. (b) Test point  $(s, t) = (4.5, 0)$ . The covariance is local in space as in (a) but across time the covariance decays slowly. (c) Test point  $(s, t) = (2, 7)$ . The covariance is periodic.

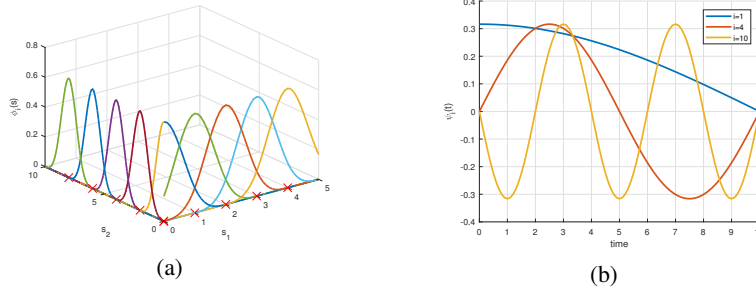


Figure 3. (a) Components of the local spatial basis  $\varphi_1(s_1)$  and  $\varphi_2(s_2)$ , respectively. Each component is centered at red crosses on the spatial axes  $s_1$  and  $s_2$ . (b) Components of the periodic temporal basis  $\psi(t)$ .

fulfilled by choosing  $N_s > \frac{R_s}{L}$  where  $R_s$  is the range of the spatial dimension. For example when  $N_s = \frac{2R_s}{L}$ , the adjacent component  $\varphi_i(s)$  have 50 percent overlap. The maximum value of  $N_s$  is limited by the number of training points and the computational resources that are available.

The temporal function  $\psi(t)$  is also specified by a basis

$$\psi(t) = \text{col}\{\psi_0(t), \psi_1(t), \dots, \psi_{N_t}(t)\}. \quad (9)$$

However, to be able to predict missing data of the type illustrated in Figure 1 we cannot rely on a localized basis for extended interpolations over space-time. Due to its good approximating properties we instead apply the periodic basis developed by Solin & Särkkä (2014) defined over a range  $\mathcal{T} = [0, R_t]$ :

$$\psi_k(t) = \begin{cases} 1, & k = 0, \\ \frac{1}{\sqrt{R_t}} \sin(k\pi \frac{t+R_t}{2R_t}), & \text{otherwise,} \end{cases} \quad (10)$$

Similar to a Fourier basis,  $\psi(t)$  allows for periodic covariance structures that capture both fixed and periodic patterns in the data along time with different frequencies. Moreover, as  $N_t$  grows, any temporally stationary covariance structure can be captured, cf. (Solin & Särkkä, 2014). Using (10), the maximum frequency in the model is  $\frac{N_t}{4R_t}$ . Hence,

depending on the data and the highest frequency periodic patterns we may expect in it, an appropriate value of  $N_t$  can be chosen.

In summary, the proposed spatio-temporal basis  $\phi(\mathbf{s}, t)$  in (6) is of dimension  $p = N_s^d(N_t + 1)$  and yields a covariance function  $\text{Cov}_\theta[y, y']$  that may vary temporally with different frequencies specific to different spatial regions, as illustrated in Figure 2. The covariance structure is determined by the parameter  $\theta$ , which we learn using a covariance-fitting methodology considered next.

## 4.2. Learning method for streaming data

We describe a covariance-fitting approach for learning the model parameter  $\theta$ , up to an arbitrary scale factor, from streaming data. Given a training dataset  $\mathcal{D}_n$ , this approach enables us to update the predictor  $\hat{y}_\theta(s, t) = \lambda_\theta^\top(s, t)y$  from (1) in a streaming fashion as  $n = 1, 2, \dots$ . We consider fitting the model covariance structure of the training data  $\mathbf{y}$ , which is parameterized by  $\theta$  in (5), to the empirical structure. Let us first define a normalized sample covariance matrix of the training data,

$$\tilde{\mathbf{K}} = \frac{(\mathbf{y} - \mathbf{1}\eta)(\mathbf{y} - \mathbf{1}\eta)^\top}{\|\mathbf{y} - \mathbf{1}\eta\|_2}.$$

Here  $\mathbf{1}$  corresponds to using  $\mathbf{u}(\mathbf{s}, t) \equiv \mathbf{1}$ . Then the optimal model parameters are given by a covariance-fitting criterion (cf. (Cressie, 1985; Anderson, 1989; Cressie & Johannesson, 2008; Stoica et al., 2011)) with minimizer:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\tilde{\mathbf{K}} - \mathbf{K}_{\boldsymbol{\theta}}\|_{\mathbf{K}_{\boldsymbol{\theta}}^{-1}}^2 \quad (11)$$

Here the matrix norm corresponds to a weighted norm which penalizes correlated residuals. The learned parameter  $\hat{\boldsymbol{\theta}}$  is invariant with respect to the mean parameter  $\eta$  and can be rescaled by an arbitrary scale factor (Zachariah et al., 2017). Moreover, the resulting predictor corresponding to  $\hat{\boldsymbol{\theta}}$  in equation (1) can be written in the equivalent form:

$$\hat{y}_{\hat{\boldsymbol{\theta}}}(\mathbf{s}, t) \equiv \boldsymbol{\alpha}^{\top}(\mathbf{s}, t) \mathbf{w}^* \quad (12)$$

where  $\boldsymbol{\alpha}(\mathbf{s}, t) = \text{col}\{1, \phi(\mathbf{s}, t)\}$ . The  $(p+1)$ -dimensional weight vector  $\mathbf{w}^*$  is defined as the minimizer

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sqrt{\hat{\mathbf{E}}[|y_i - \boldsymbol{\alpha}^{\top}(\mathbf{s}_i, t_i) \mathbf{w}|^2]} + \frac{1}{\sqrt{N}} \|\boldsymbol{\zeta} \odot \mathbf{w}\|_1 \quad (13)$$

where the elements of  $\boldsymbol{\zeta}$  are given by

$$\zeta_j = \begin{cases} \frac{1}{\sqrt{N}} \|\boldsymbol{\Phi}_{j-1}\|_2, & j > 1, \\ 0, & \text{otherwise,} \end{cases}$$

For proofs of these relations and a derivation of its computational properties, see Zachariah et al. (2017).

The resulting predictor in (12) is called the SPICE (sparse iterative covariance-based estimation) predictor. It is computed via a convex and sparsifying regularized minimization problem that can be solved using coordinate descent with recursively updated quantities at each new training point  $(\mathbf{s}_n, t_n, y_n)$ . By exploiting this structure, our predictor  $\hat{y}_{\hat{\boldsymbol{\theta}}}(\mathbf{s}, t)$  can now be updated with streaming data as  $n = 1, 2, \dots$ . A pseudocode implementation is provided in Algorithm 1. The key recursively updated quantities passed from one update to the next are the symmetric matrix  $\boldsymbol{\Gamma}$  and the vectors  $\boldsymbol{\rho}$  and  $\tilde{\mathbf{w}}$  of dimension  $p+1$  along with the scalar  $\sim \kappa$ . Here  $\tilde{\mathbf{w}}$  is the weight vector at sample  $n-1$ , which is initialized at zero along with the above variables in Algorithm 1. The runtime is linear in  $n$  and constant in memory. That is, for a fixed training data size  $N$ , the total runtime of the algorithm is on the order  $\mathcal{O}(Np^2)$  and its memory requirement is  $\mathcal{O}(p^2)$ . For further details, we refer the reader to the supplementary material. Code available at [github](#).

## 5. Synthetic data

The proposed method has been derived for predictions using large and/or streaming data sets. We now demonstrate its predictive properties using synthetic data and for the sake of reference compare it with a GPR (Gaussian process regression) method using different covariance functions  $\text{Cov}[y, y']$ .

---

### Algorithm 1 Learning from streaming datasets

---

Input:  $(\mathbf{s}_n, t_n, y_n)$  and  $\tilde{\mathbf{w}}$   
 $\boldsymbol{\Gamma} := \boldsymbol{\Gamma} + \boldsymbol{\alpha}(\mathbf{s}_n, t_n) \boldsymbol{\alpha}^{\top}(\mathbf{s}_n, t_n)$   
 $\boldsymbol{\rho} := \boldsymbol{\rho} + \boldsymbol{\alpha}(\mathbf{s}_n, t_n) y_n$   
 $\kappa := \kappa + y_n^2$   
 $\epsilon := \kappa + \tilde{\mathbf{w}}^{\top} \boldsymbol{\Gamma} \tilde{\mathbf{w}} - 2\tilde{\mathbf{w}}^{\top} \boldsymbol{\rho}$   
 $\boldsymbol{\tau} := \boldsymbol{\rho} - \boldsymbol{\Gamma} \tilde{\mathbf{w}}$   
**repeat**  
 $j = 1, \dots, p+1$   
 $c_j := \tau_j + \Gamma_{jj} \tilde{w}_j$   
**if**  $j = 1$  **then**  
 $w_j := \frac{c_j}{\Gamma_{jj}}$   
**else**  
 $a_j := \epsilon + \Gamma_{jj} \tilde{w}_j^2 + 2\tilde{w}_j \tau_j$   
 $\hat{s}_j := \text{sign}(c_j)$   
 $\hat{r}_j := \frac{|c_j|}{\Gamma_{jj}} - \frac{1}{\Gamma_{jj}} \sqrt{\frac{a_j \Gamma_{jj} - |c_j|^2}{n-1}}$   
 $w_j := \begin{cases} \hat{s}_j \hat{r}_j & \sqrt{n-1} |c_j| > \sqrt{a_j \Gamma_{jj} - |c_j|^2} \\ 0 & \text{otherwise} \end{cases}$   
**end if**  
 $\epsilon := \epsilon + \Gamma_{jj} (\tilde{w}_j - w_j^*)^2 + 2(\tilde{w}_j - w_j^*) \tau_j$   
 $\boldsymbol{\tau} := \boldsymbol{\tau} + [\boldsymbol{\Gamma}]_j (\tilde{w}_j - w_j^*)$   
**until** number of iterations equal L  
 Output:  $\mathbf{w}^* = \tilde{\mathbf{w}}$

---

### 5.1. Damped planar wave

To illustrate a dynamically evolving process, we consider planar a wave in one-dimensional space and time, cf. Figure 4a. The unknown process is generated according to:

$$y(\mathbf{s}, t) = \cos\left(\frac{2\pi}{\lambda_s}(s - v_s t)\right) \exp\left(-\frac{s}{20}\right) + \epsilon, \quad (14)$$

where  $v_s$  is the speed of the wave along space in units per second,  $\lambda_s$  is the wavelength in units of space and  $\epsilon$  is a zero-mean white Gaussian process with standard deviation  $\sigma$ .

Note that the process decays exponentially as it propagates through space. For our experiments, we set  $v_s = 3$  [spatial units/sec],  $\lambda_s = 9$  [spatial units] and  $\sigma = 0.3$ . Synthetic data is generated over a uniform grid and a subset of  $N = 700$  training points are used. Different contiguous space-time blocks are selected as test regions to resemble realistic scenarios in which the coverage of sensors, satellites or other measurement equipment is incomplete. For example, the dashed white boxes in Figure 4 emulate cases where data over a small region is missing most of the time. By contrast, the dashed black boxes correspond to cases when data over large spatial regions is missing some of the time.

The process in these test regions as well as at other randomly missing points is predicted using the proposed

method with  $N_t = 25$ ,  $N_s = 15$  and a spatial basis support set to  $L = 5$  spatial units. This results in  $\phi(s, t)$  being of dimension  $p = N_s(N_t + 1) = 390$ . The mean-square error (MSE) of the prediction is shown in Figure 4b and evaluated using 25 Monte Carlo simulations. The region in the white box extends over almost the entire time dimension, hence there are very few neighbouring training points in time to draw upon for prediction and no information about the periodicities in the region. Instead our method leverages the neighbouring spatial information to obtain a good prediction resulting in a low MSE. Both black boxes are test regions that have neighbouring training points that provide temporal information about the process. However, left region has training points both before and after whereas the right region only has points before, yielding a more challenging prediction problem. Nevertheless, the proposed method is able to learn both the periodic and the local damping patterns to provide accurate predictions in both regions.

We include also the MSE of GPR using two different covariance functions learned by a numerical maximum likelihood search. While this method is not applicable to the streaming data of interest here, it provides a performance reference. First, we use a Matérn ARD covariance model (Rasmussen & Nickisch, 2010) to carefully adapt both space and time dimensions. In Figure 4c it is seen that the resulting prediction errors are markedly worse for the large missing spatial regions and the method naturally fails to capture the periodic pattern of the process. Next, we use a periodic Matérn ARD covariance model to also capture space-time periodicity. However, the MSE (Figure 4d) is degraded throughout, which is possibly due to the non-convex optimization problem used to learn the model parameters. It may lead to local minima issues, including learning erroneous periods.

## 5.2. Varying seasonalities across space

Here we generate a process that emulates scenarios of temporal periodicities which may vary across spatial regions. This occurs e.g. in climate data. Figure 5a shows a realization of a process generated according to

$$y(s, t) = \cos\left(\frac{2\pi}{T(s)}t\right) + \varepsilon \quad (15)$$

where the period  $T(s)$  differs across space and  $\varepsilon$  is zero-mean white Gaussian process with standard deviation  $\sigma = 0.3$ . In the upper region of the spatial domain  $T(s) = \infty$ , i.e., the process has a constant mean. In the middle and bottom regions  $T(s)$  is large and small, respectively. The data is generated over a uniform grid and a subset of  $N = 600$  points is used for training. A contiguous space-time block, marked by the dashed black box in Figure 5, forms a test region to emulate scenarios where data can be missing

over a large spatial region for some time.

For the proposed method we use  $N_t = 35$ ,  $N_s = 15$  and a support of  $L = 3$  for the spatial basis, so that  $p = N_s(N_t + 1) = 540$ . For the GPR we use the periodic Matérn ARD kernel. Figures 5b and 5c show the MSE performance of the proposed method and GPR respectively which were obtained using 25 Monte Carlo simulations. The MSE of the proposed method is overall lower than that of GPR, both in the dashed test region as well as outside it. Unlike the proposed method, GPR has one parameter to fit to an overall periodic pattern and is thus unable to learn spatially localized patterns. Thus after learning, the process is predicted to be nearly constant along time for all parts of the spatial region.

## 6. Real data

We now demonstrate the proposed method for much larger, and possibly streaming, real-world datasets.

### 6.1. Pacific Sea Surface Temperature

As a first application example, we use tropical Pacific Sea Surface Temperature (SST) data (Wikle, 2011). These data represent gridded monthly SST anomalies, in  $^{\circ}\text{C}$ , from January 1970 through March 2003 over a spatial region from  $29^{\circ}\text{S}$  to  $29^{\circ}\text{N}$  and  $124^{\circ}\text{E}$  to  $70^{\circ}\text{W}$ . The spatial resolution of the data is  $2^{\circ}$  in both latitude and longitude.

Here we consider data from the first 36 months, making the total number of space-time data points equal to  $36 \times 2520 = 90720$ . In the first experiment, training points are sampled randomly across space-time and the missing data constitute the test points. Here we set  $N = 63503$  as the number of training points. For the proposed method we set  $N_t = 100$ ,  $N_s = 8$  and the spatial support  $L$  to be half of each spatial dimension. Then  $p = N_s^2(N_t + 1) = 6464$ . Figure 6a shows the prediction error histogram of all test points across the spatio-temporal domain. We see that it is centered around zero and its dispersion is considerably narrower than the dynamic range of the data.

In the second experiment, we select a contiguous space-time block as a test region in addition to other test points to evaluate the performance in scenarios where data over entire spatial regions are missing for a period of time. Data falling within the spatial region marked by the black dashed box in Figure 6c is missing beyond month 26, as indicated by the black dashed line in Figure 6d. Here  $N = 18144$  are the number of training points. The prediction error histogram for this second experiment is shown in Figure 6b and remains fairly narrow. Figure 6c illustrates the predicted SST anomalies [ $^{\circ}\text{C}$ ] for a spatial slice at month  $t = 30$ . We pick a spatial point in a region where the El Niño effect, i.e., the periodic warming of the equatorial Pa-

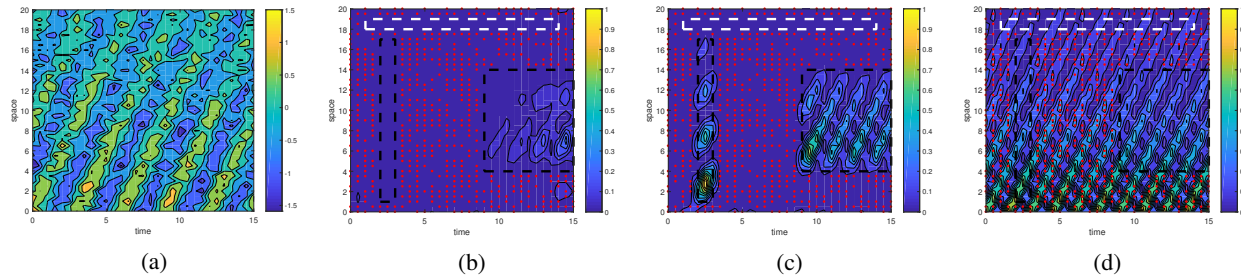


Figure 4. (a) Realization of the damped planar wave process  $y(s, t)$  defined in (14). (b) MSE of the proposed method. The red dots denote training points. The white and black dashed boxes represent contiguous space-time test regions where the data is missing. (c) MSE of GPR using the Matérn ARD covariance model. (d) MSE of GPR using periodic Matérn ARD covariance function.

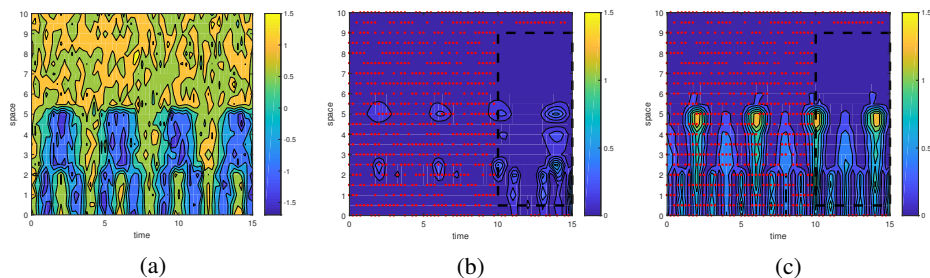


Figure 5. (a) Realization of the process  $y(s, t)$  defined in (15) with varying periods across space. (b) MSE of the proposed method which is able to learn different periodic temporal patterns across space. The red dots denote training points. The black dashed box marks a contiguous test region. (c) MSE of GPR using periodic Matérn ARD covariance model.

cific Sea Surface (Sarachik & Cane, 2010), is known to be noticeable. The prediction of the SST anomalies at this spatial location across time along with the true SST is illustrated with Figure 6d. Note that the predictor is able to track the rising temperature deviation also for the missing data.

## 6.2. Precipitation data

As a second application example, we use precipitation data from the Climate Research Unit (CRU) time series datasets of climate variations (Jones & Harris, 2013). The precipitation data consists of monthly rainfall in millimeter over a period from 1901 to 2012 obtained with high spatial resolution (0.5 by 0.5 degree) over the whole planet. Here we consider a five year period from 2001 to 2005 and between spatial coordinates  $95^\circ\text{W}$  to  $107^\circ\text{W}$  and  $40^\circ\text{N}$  to  $50^\circ\text{N}$ . This yields a total number of 28 800 data points.

The spatial region indicated by the black dashed box in Figure 7b beyond month  $t = 47$ , as seen in Figure 7c, constitutes a contiguous test region, in addition to other randomly selected test points. The remaining  $N = 14\,400$  points are used for training.

For the proposed method we set  $N_t = 300$ ,  $N_s = 6$  and the spatial support  $L$  to be half of each spatial dimension. Then  $p = N_s^2(N_t + 1) = 10\,836$ . Figure 7a shows the

prediction error histogram for the precipitation test data. It is centered around zero and its dispersion is narrower than the dynamic range of the data. Figure 7b shows the contour plot of predicted precipitation for a spatial slice at month  $t = 54$ . The red cross and plus marker indicate spatial points whose actual and predicted time series are compared in Figures 7c and 7d, respectively. Note that the estimated precipitation tracks the true precipitation well everywhere even to the right of the black dashed line where the data was not seen during training. Note the ability of the predictor to track the different seasonal patterns in the missing regions.

## 7. Conclusion

We proposed a method in which a spatio-temporal predictor  $\hat{y}_{\hat{\theta}}(s, t)$  can be learned and updated sequentially as spatio-temporal data is obtained as a stream. It is capable of capturing spatially varying temporal patterns, using a non-stationary covariance model that is learned using a covariance-fitting approach. We demonstrated, using both simulated and real climate data, that it is capable of producing accurate predictions in large unobserved space-time test regions. In future work, we intend to further improve the computational efficiency of the method by exploiting the spatially localized structure of the covariance model.

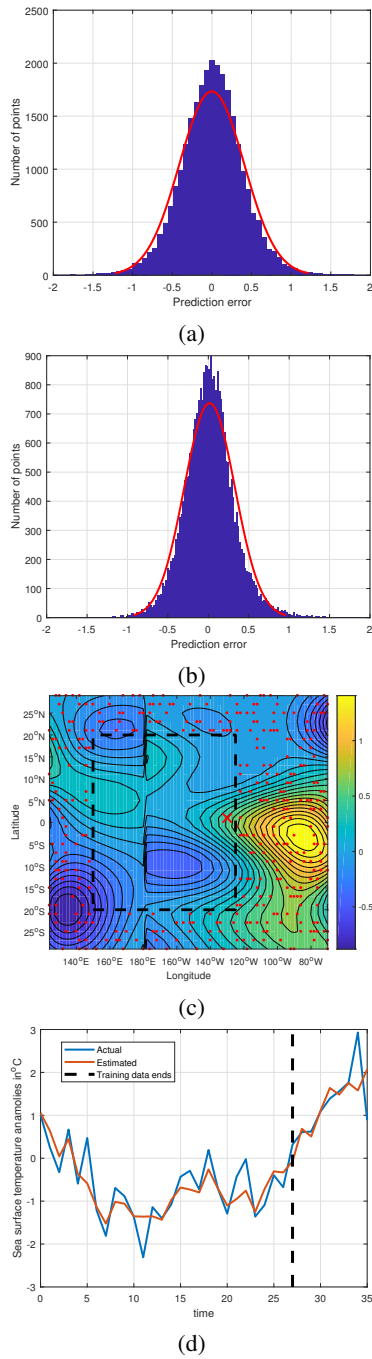


Figure 6. (a) Histogram of prediction error of all test points in the first scenario with randomly sampled training points. Note that the dynamic range of the data is  $[-3.2, 3.2]$  °C. (b) Histogram of prediction error of all test points for the second scenario with a contiguous space-time block as test region. The plots in red are fitted Gaussian distributions (c) Contour plot of predicted SST for a single spatial slice at time  $t = 30$ . The red dots denote training points and the black dashed box indicates a contiguous test region. The red cross denotes a point of interest in which the El Niño effect can be observed. (d) Comparison of time series of actual and estimated SST for the point marked by the red cross in 6c. The data to the right of the black dashed line is not seen during training.

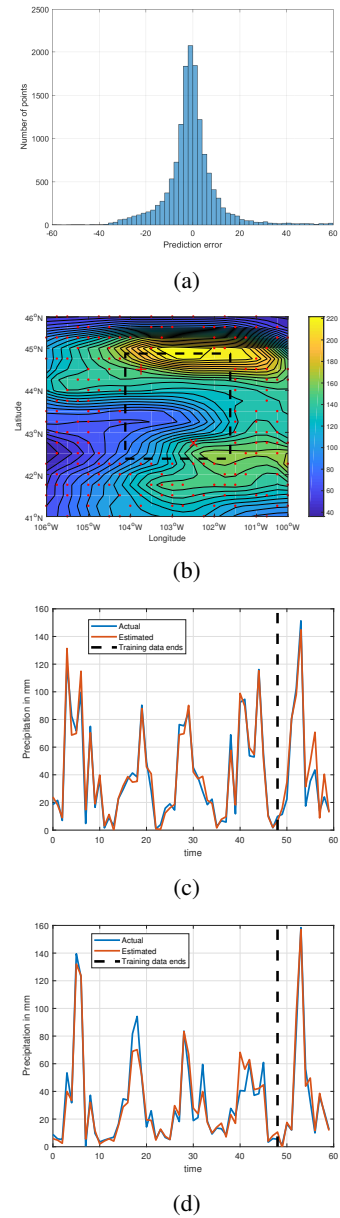


Figure 7. (a) Histogram of prediction error for all test points of precipitation data. The dynamic range of the data is  $[0, 226]$  millimeters (b) Contour plot of predicted precipitation for a single spatial slice at time  $t = 54$ . Data inside the black dashed box marks the contiguous test region and is not seen during training. The red dots denote training points. (c) The actual and estimated precipitation time series for the the spatial point marked by red cross in figure 7b. (d) The actual and estimated precipitation time series for the the spatial point marked by red addition sign in figure 7b. The data to the right of the black dashed line is part of the contiguous test region and is not used during training.



## References

- Anderson, T. W. Linear latent variable models and covariance structures. *Journal of Econometrics*, 41(1):91–119, 1989.
- Bijl, H., van Wingerden, J.-W., Schön, T. B., and Verhaegen, M. Online sparse Gaussian process regression using FITC and PITC approximations. *IFAC-PapersOnLine*, 48(28):703–708, 2015.
- Cressie, N. Fitting variogram models by weighted least squares. *Journal of the International Association for Mathematical Geology*, 17(5):563–586, 1985.
- Cressie, N. and Johannesson, G. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1): 209–226, 2008.
- Cressie, N. and Wikle, C. K. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.
- Cressie, N., Shi, T., and Kang, E. L. Fixed rank filtering for spatio-temporal data. *Journal of Computational and Graphical Statistics*, 19(3):724–745, 2010.
- Faul, A. C. and Tipping, M. E. Analysis of sparse Bayesian learning. In *Advances in neural information processing systems*, pp. 383–389, 2002.
- Fox, E. B. and Dunson, D. B. Bayesian nonparametric covariance regression. *Journal of Machine Learning Research*, 16:2501–2542, 2015.
- Jones, P. and Harris, I. Climatic research unit (CRU) time-series (ts) version 3.21 of high resolution gridded data of month-by-month variation in climate (jan. 1901–dec. 2012). *NCAS British Atmospheric Data Centre*, 2013.
- Lopes, H. F., Salazar, E., and Gamerman, D. Spatial dynamic factor analysis. *Bayesian Analysis*, 3(4):759–792, 2008.
- Quiñonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.
- Rasmussen, C. E. and Nickisch, H. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.
- Rasmussen, C. E. and Williams, C. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.
- Saatçi, Y. *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer, 2012.
- Sarachik, E. S. and Cane, M. A. *The El Nino-southern oscillation phenomenon*. Cambridge University Press, 2010.
- Särkkä, S., Solin, A., and Hartikainen, J. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.
- Solin, A. and Särkkä, S. Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*, 2014.
- Stein, M. L. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.
- Stoica, P., Babu, P., and Li, J. New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sampled data. *IEEE Trans. Signal Processing*, 59(1):35–47, 2011.
- Tipping, M. E. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.
- Wasserman, L. *All of Nonparametric Statistics*. Springer Texts in Statistics. New York: Springer-Verlag, 2006.
- Wikle, C. K. Sea surface temperature anomaly data, 2011. URL <http://faculty.missouri.edu/~wiklec/datasets.html>.
- Wilson, A. G., Gilboa, E., Nehorai, A., and Cunningham, J. P. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, pp. 3626–3634, 2014.
- Zachariah, D., Stoica, P., and Schön, T. B. Online learning for distribution-free prediction. *arXiv preprint arXiv:1703.05060*, 2017.
- Zammit-Mangion, A. and Cressie, N. FRK: An R package for spatial and spatio-temporal prediction with large datasets. *arXiv preprint arXiv:1705.08105*, 2017.

## 8. Acknowledgements

*NewLEADS - New Directions in Learning Dynamical Systems* (contract number: 621-2016-06079), funded by the Swedish Research Council and the project *ASSEMBLE* (contract number: RIT15-0012), funded by the Swedish Foundation for Strategic Research (SSF).