Structured Gaussian Processes with Twin Multiple Kernel Learning

Çiğdem Ak CAK14@KU.EDU.TR
Graduate School of Sciences and Engineering, Koç University, İstanbul, Turkey

Önder Ergönül OERGONUL@KU.EDU.TR
Department of Infectious Diseases and Clinical Microbiology, School of Medicine, Koç University, İstanbul, Turkey

Mehmet Gönen MEHMETGONEN@KU.EDU.TR
Department of Industrial Engineering, College of Engineering, Koç University, İstanbul, Turkey
School of Medicine, Koç University, İstanbul, Turkey

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Vanilla Gaussian processes (GPs) have prohibitive computational needs for very large data sets. To overcome this difficulty, special structures in the covariance matrix, if exist, should be exploited using decomposition methods such as the Kronecker product. In this paper, we integrated the Kronecker decomposition approach into a multiple kernel learning (MKL) framework for GP regression. We first formulated a regression algorithm with the Kronecker decomposition of structured kernels for spatiotemporal modeling to learn the contribution of spatial and temporal features as well as learning a model for out-of-sample prediction. We then evaluated the performance of our proposed computational framework, namely, structured GPs with twin MKL, on two different real data sets to show its efficiency and effectiveness. MKL helped us extract relative importance of input features by assigning weights to kernels calculated on different subsets of temporal and spatial features.

Keywords: spatiotemporal modeling, regression, knowledge extraction, structured Gaussian processes, multiple kernel learning

1. Introduction

The kernel functions are the basic building blocks of kernel-based algorithms, and they directly affect the prediction performance and allow to try different levels of model complexities without changing the inference and/or training procedures. The standard training procedure is to select the best single kernel using, for example, a cross-validation step before testing. Instead, combinations of kernel functions have also been proposed to capture the relative importance of input features/representations (Gönen and Alpaydın, 2011).

For large data sets, Gaussian processes (GPs) might become computationally intensive. That is why several decomposition algorithms have been previously proposed to make the inference faster such as Nyström approximation (Rasmussen and Williams, 2006), approximation using Hadamard and diagonal matrices (Le et al., 2013), or Kronecker methods (Bonilla et al., 2007; Finley et al., 2009; Saatçi, 2011; Stegle et al., 2011; Riihimäki and Vehtari, 2014; Wilson et al., 2014; Gilboa et al., 2015).

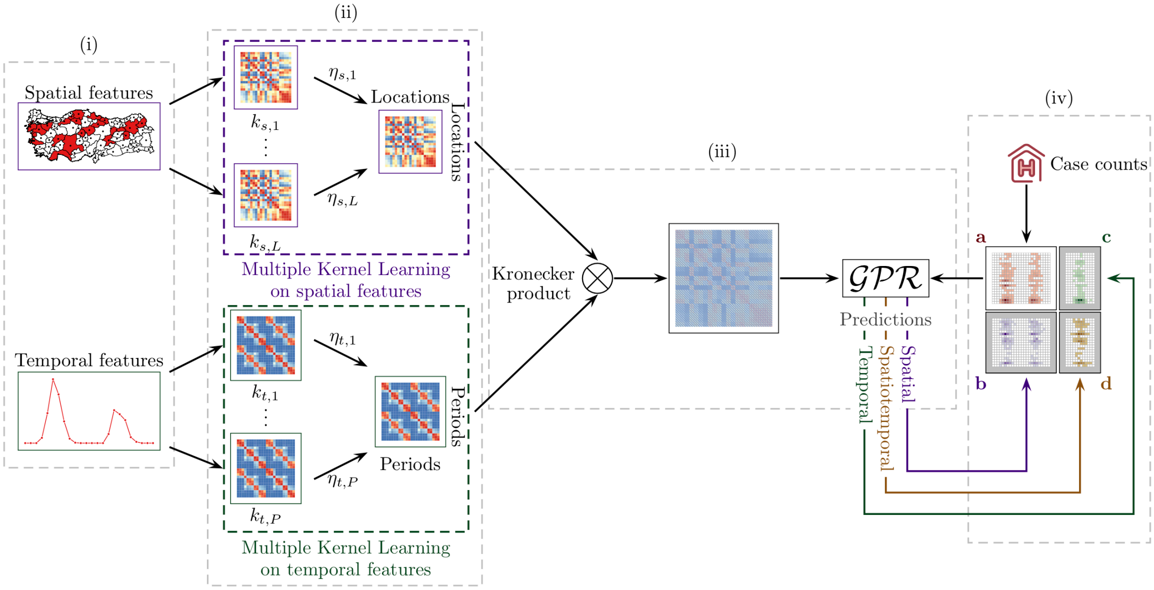

Figure 1: Our computational framework for spatiotemporal inference: (i) temporal and spatial feature extraction, (ii) twin multiple kernel learning, (iii) Kronecker product based GP regression ($\mathcal{GPR}$), and (iv) prediction scenarios: (a) Given response values for observed location and time pairs to make inference in three different scenarios: (b) spatial prediction, (c) temporal prediction, and (d) spatiotemporal prediction.

GPs have been used in many applications for temporal and spatial prediction such as environmental surveillance (Nguyen et al., 2017), reconstruction of sea surface temperatures (Luttinen and Ilin, 2012), drug–target interaction prediction (Airola and Pahikkala, in press), global land-surface precipitation prediction (Wang and Chaib-draa, 2013), and wind power forecasting (Chen et al., 2013) as well as spatiotemporal modeling (Särkkä and Hartikainen, 2012; Andrade-Pacheco, 2015). There is also a significant number of studies on GPs with application to epidemiology (Vanhatalo et al., 2010; Andrade-Pacheco et al., 2014; Senanayake et al., 2016; Bhatt et al., 2017).

1.1. Our Contributions

In this study, we proposed a GP approach with Kronecker decomposition for spatiotemporal regression problems to learn combinations of kernels for both pattern discovery and fast inference. We performed experiments under three prediction scenarios on two real-life data sets from two different domains.

Figure 1 illustrates the overview of our proposed computational framework with three possible prediction scenarios. Our framework has four main components: (i) extracting spatial and temporal features using the input data, (ii) calculating multiple kernels for both spatial and temporal features, (iii) using Kronecker product-based spatiotemporal GP formulation for prediction, and (iv) three different prediction scenarios that can be seen in real-life applications.

We first begin with a review of GPs and introduce structured GPs (SGPs) in Section 2. In Section 3, we describe a multiple kernel learning (MKL) approach for inference and hyper-parameter learning in SGPs. Finally, in Section 4, we elaborate on the model specifications that we used for computational experiments and report the empirical results obtained by comparing our proposed approach against other machine learning algorithms.

2. Background on Structured Gaussian Processes

2.1. Gaussian Processes

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be given input vectors and target outputs of a data set. GPs model the relationship between inputs and outputs as follows:

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\xi},$$

where $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}^\top$ is the vector of outputs, $\boldsymbol{f} = \begin{bmatrix} f_1 & f_2 & \cdots & f_N \end{bmatrix}^\top$ is the vector of underlying true outputs, and $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_N \end{bmatrix}^\top$ is the noise vector. Both $\boldsymbol{f}$ and $\boldsymbol{\xi}$ assumed to be normally distributed:

$$\begin{aligned} p(\boldsymbol{f}) &\sim \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K}), \\ p(\boldsymbol{\xi}) &\sim \mathcal{N}(\boldsymbol{\xi}|\mathbf{0}, \sigma_y^2\mathbf{I}), \end{aligned}$$

where $\mathbf{K} = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i=1,j=1}^{N,N}$ is a positive semi-definite kernel matrix (i.e., covariance matrix), and σ_y^2 is noise variance. Then, the likelihood can be written as

$$p(\boldsymbol{y}|\mathbf{X}, \sigma_y^2) \sim \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathbf{K} + \sigma_y^2\mathbf{I}),$$

where $\mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N \end{bmatrix}$ is the input data matrix.

The predictive distribution of the target output $y_\star$ of a given new data point $\boldsymbol{x}_\star$ conditioned on the training data has also a Gaussian density:

$$\begin{aligned} p(y_\star|\boldsymbol{x}_\star, \mathbf{X}, \boldsymbol{y}, \sigma_y^2) &\sim \mathcal{N}(y_\star|\mu_\star, \sigma_\star^2), \\ \mu_\star &= k(\boldsymbol{x}_\star, \mathbf{X})(\mathbf{K} + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}, & (1) \\ \sigma_\star^2 &= k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) - k(\boldsymbol{x}_\star, \mathbf{X})(\mathbf{K} + \sigma_y^2\mathbf{I})^{-1}k(\mathbf{X}, \boldsymbol{x}_\star). & (2) \end{aligned}$$

Note that $k(\boldsymbol{x}_\star, \mathbf{X}) = k(\mathbf{X}, \boldsymbol{x}_\star)^\top$ is a row vector.

2.2. Structured Gaussian Processes

GPs have intensive computational and memory requirements for large data sets. GP inference requires evaluating $(\mathbf{K} + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}$ for Equations (1) and (2). For this operation, the most common approach is to take the Cholesky decomposition of $(\mathbf{K} + \sigma_y^2\mathbf{I})$, which is also computationally demanding. However, by exploiting the structure of the covariance matrix $\mathbf{K}$, this step can be performed very efficiently.

In this section, we describe an approach to exploit the special structure of the kernel matrix to speed up inference, which allows us to efficiently determine the singular values of the covariance matrix $\mathbf{K}$ and enables us to efficiently compute $(\mathbf{K} + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}$ for faster training and prediction.

We consider data sets such that each input data point $\boldsymbol{x}_i$ is defined as a pair of spatial and temporal information $(\boldsymbol{s}_l, \boldsymbol{t}_p)$, where l indexes locations, and p indexes time periods. Let L be the number of locations and P be the number of time periods. The response matrix $\mathbf{Y}$ is then a matrix of size $L \times P$, and the output $y_{l,p}$ corresponds to the input $(\boldsymbol{s}_l, \boldsymbol{t}_p)$. In such a case, the covariance function is separable as follows:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k((\boldsymbol{s}_l, \boldsymbol{t}_p), (\boldsymbol{s}_m, \boldsymbol{t}_q)) = k_s(\boldsymbol{s}_l, \boldsymbol{s}_m) k_t(\boldsymbol{t}_p, \boldsymbol{t}_q),$$

where k_s and k_t functions are defined on the spatial and temporal features, respectively. The kernel matrix $\mathbf{K}$ is of size $LP \times LP$, which can be written as a Kronecker product:

$$\mathbf{K} = \mathbf{K}_s \otimes \mathbf{K}_t,$$

where $\mathbf{K}_s$ and $\mathbf{K}_t$ are $L \times L$ and $P \times P$ kernel matrices for spatial and temporal features obtained using k_s and k_t functions, respectively. Kronecker decomposition was first used within GP to model data, where inputs lie on a Cartesian grid (Saatçi, 2011). We can replace this more complex kernel formulation into standard GP Equations (1) and (2), and obtain SGPs to exploit spatiotemporal structures.

$$\begin{aligned} p(y_\star | \boldsymbol{x}_\star, \mathbf{X}, \mathbf{Y}, \sigma_y^2) &\sim \mathcal{N}(y_\star | \mu_\star, \sigma_\star^2), \\ \mu_\star &= (k_{s,\star} \otimes k_{t,\star})^\top (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})^{-1} \operatorname{vec}(\mathbf{Y}), & (3) \\ \sigma_\star^2 &= k_s(s_\star, s_\star) k_t(t_\star, t_\star) - (k_{s,\star} \otimes k_{t,\star})^\top (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})^{-1} (k_{s,\star} \otimes k_{t,\star}), & (4) \end{aligned}$$

where $\operatorname{vec}(\cdot)$ converts the input matrix into a column vector. Fortunately, these matrix computations can be performed efficiently using the following properties:

$$\begin{aligned} (\mathbf{A} \otimes \mathbf{B})^{-1} &= \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, & (5) \\ (\mathbf{A}\mathbf{B}) \operatorname{vec}(\mathbf{X}) &= \operatorname{vec}(\mathbf{B}\mathbf{X}\mathbf{A}^\top), & (6) \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= (\mathbf{A}\mathbf{C}) \otimes (\mathbf{B}\mathbf{D}). & (7) \end{aligned}$$

Equation (5) helps efficient computation of the inverse of $\mathbf{K}_s \otimes \mathbf{K}_t$ even though it is size of $LP \times LP$. This property is easy to implement if there is no noise term in the inverse using singular value decomposition (SVD). We can also develop an efficient implementation to take the inverse of $(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})$ as follows:

$$\begin{aligned} \mathbf{K}_s &= \mathbf{U}_s \mathbf{D}_s \mathbf{U}_s^\top, \\ \mathbf{K}_t &= \mathbf{U}_t \mathbf{D}_t \mathbf{U}_t^\top, \end{aligned}$$

where the left-singular vectors and right-singular vectors are identical since the kernel matrices are positive semi-definite. Hence, Kronecker product has the following decomposition:

$$\mathbf{K}_s \otimes \mathbf{K}_t = (\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t)(\mathbf{U}_s \otimes \mathbf{U}_t)^\top.$$

The matrix inversion operation can be replaced by the following formula:

$$(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})^{-1} = (\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2 \mathbf{I})^{-1} (\mathbf{U}_s \otimes \mathbf{U}_t)^\top. \quad (8)$$

We can rewrite mean and variance of SGPs using Equation (8). After this change, mean and variance calculations in Equations (3) and (4) can be performed very efficiently using Equations (6) and (7) without explicitly storing the inverse of $(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})$. In this step, we calculate the SVDs of smaller matrices $\mathbf{K}_s$ and $\mathbf{K}_t$, which have complexities $\mathcal{O}(L^3)$ and $\mathcal{O}(P^3)$, respectively. At the end, we have to take the inverse of the diagonal matrix $(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2 \mathbf{I})$ in Equation (8), which has $\mathcal{O}(LP)$ complexity. These steps make the overall complexity of our algorithm $\mathcal{O}(L^3 + P^3)$.

3. Structured Gaussian Processes with Twin Multiple Kernel Learning

In the previous section, we proposed a computational framework using SGP regression for spatiotemporal modeling, which is suitable to capture highly complex dependencies between input and output variables thanks to its nonlinear nature brought by kernel functions. In this section, we show how to combine SGP with an MKL approach to conjointly perform knowledge extraction and prediction, which we named as SGPs with twin MKL (SGP2MKL). In our formulation, each spatial and temporal feature is fed into a kernel function, and then MKL provides us with the relative importance of these features by assigning weights to their respective kernels.

Our main hypothesis about the spatiotemporal processes is that response values depend on both time and location. We need a kernel function, such that nearby observations in time and/or space, should produce similar values. The squared exponential covariance function (Rasmussen and Williams, 2006), which is also known as Gaussian kernel function, between two data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be defined as

$$k_{\mathcal{G}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2s^2}\right),$$

where s is the kernel width, and $\|\cdot\|_2$ is the ℓ_2-norm. We chose to use the Gaussian kernel for both spatial and temporal features.

3.1. Twin Multiple Kernel Learning

To identify the importance of individual and pairwise interaction effects of features, we defined both spatial and temporal kernels as linear combinations of Gaussian kernels and their pairwise interactions:

$$\mathbf{K}_s = \eta_{s,1}\mathbf{K}_{s,1} + \cdots + \eta_{s,P_s}\mathbf{K}_{s,P_s} + \eta_{s,P_s+1}\underbrace{(\mathbf{K}_{s,1} \circ \mathbf{K}_{s,2})}_{\mathbf{K}_{s,P_s+1}} + \cdots + \eta_{s,\frac{P_s(P_s+1)}{2}}\underbrace{(\mathbf{K}_{s,P_s-1} \circ \mathbf{K}_{s,P_s})}_{\mathbf{K}_{s,\frac{P_s(P_s+1)}{2}}},$$

$$\mathbf{K}_t = \eta_{t,1}\mathbf{K}_{t,1} + \cdots + \eta_{t,P_t}\mathbf{K}_{t,P_t} + \eta_{t,P_t+1}\underbrace{(\mathbf{K}_{t,1} \circ \mathbf{K}_{t,2})}_{\mathbf{K}_{t,P_t+1}} + \cdots + \eta_{t,\frac{P_t(P_t+1)}{2}}\underbrace{(\mathbf{K}_{t,P_t-1} \circ \mathbf{K}_{t,P_t})}_{\mathbf{K}_{t,\frac{P_t(P_t+1)}{2}}},$$

where $\circ$ is Hadamard product of two given matrices, and P_s and P_t are the total numbers of spatial and temporal features, respectively.

3.2. Inference Procedure

Here, we explain how we infer the noise variance σ_y^2, spatial and temporal kernel weights $\{\eta_{s,m}\}_{m=1}^{P_s(P_s+1)/2}$ and $\{\eta_{t,n}\}_{n=1}^{P_t(P_t+1)/2}$. We can learn them using a maximum likelihood approach because the required computations (integrals over the parameters) are analytically tractable for standard GPs. The marginal likelihood and its partial derivatives with respect to the hyper-parameters of a GP are given as follows (Rasmussen and Williams, 2006):

$$\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top \mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi, \tag{9}$$

$$\frac{\partial \log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_m} = \frac{1}{2}\boldsymbol{y}^\top \mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_m}\mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\operatorname{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_m}\right), \tag{10}$$

where $\boldsymbol{\theta}$ is the vector of the parameters of the covariance function, and $\boldsymbol{\alpha} = \mathbf{K}^{-1}\boldsymbol{y}$. In our case, $\boldsymbol{\theta} = (\{\eta_{s,m}\}, \{\eta_{t,n}\}, \sigma_y)$, and $\mathbf{K} = \mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I}$.

To learn the model parameters, we need to take the derivatives of $\mathbf{K}$ with respect to the spatial kernel weights $\{\eta_{s,m}\}$, temporal kernel weights $\{\eta_{t,n}\}$, and noise deviation σ_y:

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \eta_{s,m}} = \frac{\partial \mathbf{K}_s}{\partial \eta_{s,m}} \otimes \mathbf{K}_t, \tag{11}$$

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \eta_{t,n}} = \mathbf{K}_s \otimes \frac{\partial \mathbf{K}_t}{\partial \eta_{t,n}}, \tag{12}$$

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \sigma_y} = 2\sigma_y\mathbf{I}, \tag{13}$$

where the derivatives of spatial and temporal kernels with respect to the weight parameters are just the Gaussian kernels or the Hadamard products of two Gaussian kernels: $\partial\mathbf{K}_s/\partial\eta_{s,m} = \mathbf{K}_{s,m}$ and $\partial\mathbf{K}_t/\partial\eta_{t,n} = \mathbf{K}_{t,n}$. We first plugged these derivatives into Equations (11)–(13) and then plugged these resulting equations into the gradient calculation in Equation (10). The first term of the gradient can be computed efficiently using partial derivatives in Equations (11)–(13) and Kronecker properties in Equations (5)–(7). The second term of the gradient can also be computed efficiently by exploiting the cyclic property of trace function and the SVD decompositions as follows:

$$\operatorname{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_m}\right) = \operatorname{diag}(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma^2\mathbf{I})^{-1}\operatorname{diag}\left((\mathbf{U}_s \otimes \mathbf{U}_t)^\top\left(\frac{\partial \mathbf{K}}{\partial \theta_m}\right)(\mathbf{U}_s \otimes \mathbf{U}_t)\right)$$

where the latter term can be computed efficiently as a Kronecker product since the partial derivatives are Kronecker product and its diagonal as a Kronecker product of the diagonals of each factor in the product. As a result, we obtained three general gradient equations for the spatial kernels weights, temporal kernel weights, and noise deviation parameters.

We estimated the parameters using a constrained optimization method in R package `alabama` (Varadhan, 2015). We used the function `constrOptim.nl`, which uses an objective function to be optimized (i.e., likelihood function in Equation (9)), the gradient of the objective function evaluated at the argument (i.e., gradient in Equation (10)), constraints on parameters, and starting values for parameters (i.e., uniform kernel weights) as inputs. We constrained the parameters as follows: (a) They all should be non-negative: $\eta_{s,m} \geq 0$, $\eta_{t,n} \geq 0$, and $\sigma_y > 0$. (b) Kernel weights for spatial and temporal features should sum up to one: $\sum \eta_{s,m} = 1$ and $\sum \eta_{t,n} = 1$.

4. Experiments

We performed experiments on two real-life data sets: (a) an infectious disease surveillance data set and (b) a monthly average surface temperature data set. We compared SGP and SGP2MKL against two other machine learning algorithms used in ecological and epidemiological applications for spatial and temporal prediction scenarios, namely, boosted regression tree (BRT) and random forest regression (RFR) algorithms. These two algorithms are frequently used machine learning algorithms in this type of applications (Bhatt et al., 2013; Hay et al., 2013; Kane et al., 2014), and they are readily available as R software packages (Liaw and Wiener, 2015; Ridgeway, 2017). Our implementations of SGP and SGP2MKL in R and source codes to reproduce the experimental results reported are publicly available at `https://github.com/cigdemak/sgp2mkl`.

Two performance measures were used to evaluate the predictive accuracy of the proposed approaches: the Pearson's correlation coefficient (PCC) and the normalized root mean square error (NRMSE). Predictive performances of the algorithms were tested under three different prediction scenarios: (i) temporal prediction scenario (i.e., predicting future time points by looking at historical data), see Figure 1(c), (ii) spatial prediction scenario (i.e., predicting historical data for new locations using data for observed locations), see Figure 1(b), (iii) spatiotemporal prediction scenario (i.e., predicting future time points in new locations), see Figure 1(d).

In all experiments, instead of learning kernel hyper-parameters using type-II maximum likelihood (Rasmussen and Williams, 2006), we used a well-known heuristic for kernel hyper-parameter tuning, where we set the width parameter to the average pairwise Euclidean distance between training instances for each kernel. In SGP experiments, the noise deviation σ_y was chosen as the standard deviation of the training case counts, and all single and pairwise kernels were used with uniform weights.

Last one sixth of time periods for each data set was taken as the test set, and remaining time periods were used as training set. Half of the geographical locations were sampled randomly as the training set. For temporal scenario, since we have an ordered training and test sets, we had a single experiment, whereas, for spatial and spatiotemporal scenarios, we repeated the experiments 100 times with randomly sampled training sets to minimize the effect of sampling and to get more robust results.

4.1. Predicting Crimean–Congo Hemorrhagic Fever Infection Case Counts

Crimean–Congo hemorrhagic fever (CCHF) is a fatal viral infection mostly seen in parts of Africa, Asia, Eastern Europe, and Middle East. The virus causes severe complications in humans with the reported mortality rate of 5–40%. CCHF is the most widely spread infectious disease among tick-borne diseases (Ergönül, 2006). Humans might get infected through the bites of the ticks carrying the virus, direct contact with the bodily fluids of a patient with CCHF during the acute phase of infection, or contact with blood or tissues from viremic livestock.

The surveillance data set consists of monthly infected case counts for each province in Turkey (81 provinces) between January 2004 to December 2015. Thus, there are 81 locations and 144 (12 × 12) time periods. Figure 2 reports the yearly CCHF case counts between 2004 and 2015 for 81 provinces.

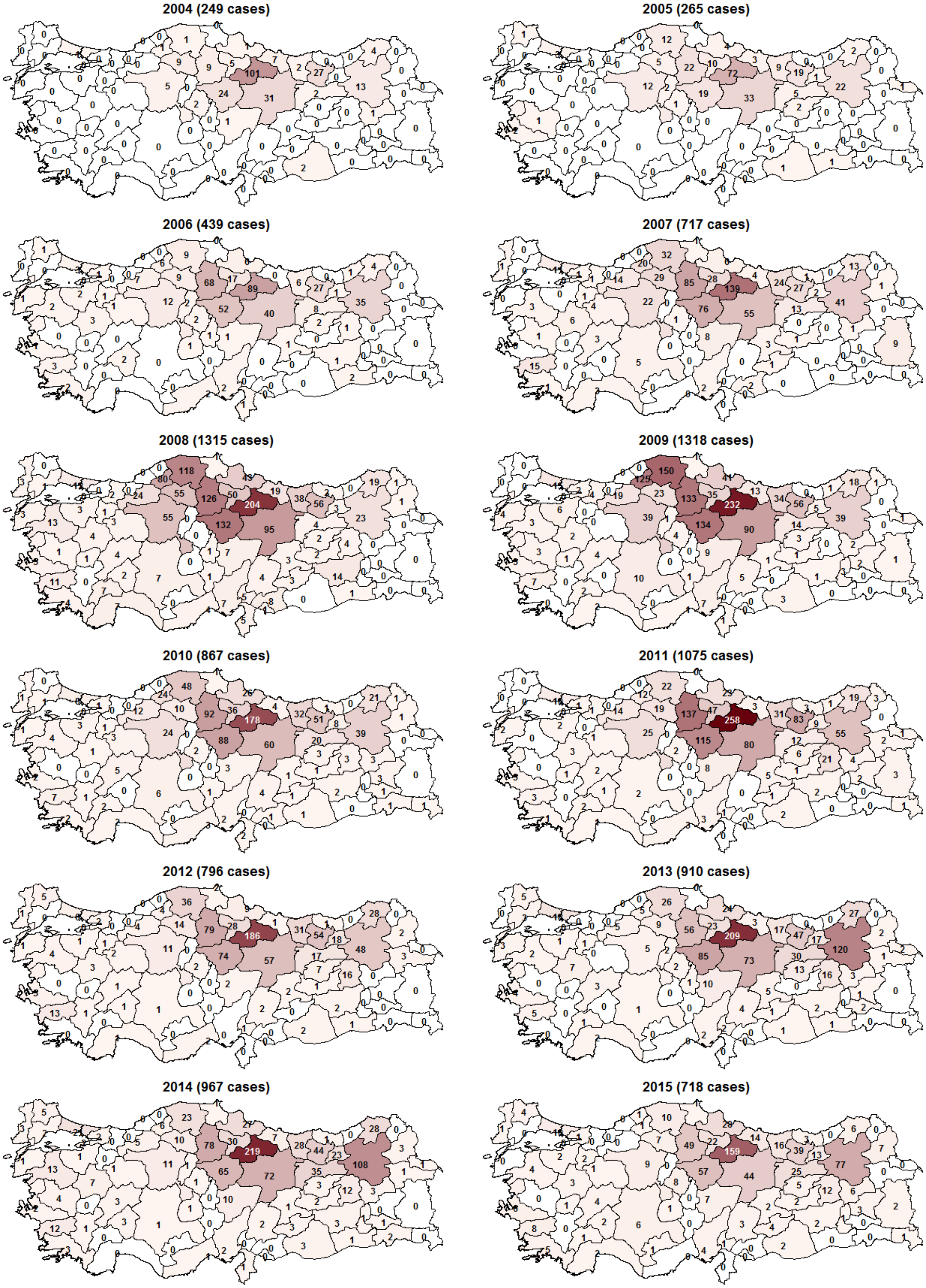

Figure 2: Yearly CCHF case counts between years 2004 and 2015 for 81 provinces of Turkey.

To be able to model case counts using Gaussian distribution, we first $\log_2$-scaled the CCHF surveillance data set. Using a Gaussian model on the logarithm of the case count data has been used in previous GP research (Andrade-Pacheco, 2015). First ten years (i.e., 2004–2013) were used as temporal training set and last two years (i.e., 2014 and 2015) as test set. 41 out of 81 locations were randomly chosen as spatial training set and the remaining 40 locations were used as the spatial test set. Hence, we had 9,720 ($81 \times 10 \times 12$) instances, 5,904 ($41 \times 12 \times 12$) instances, and 4,920 ($41 \times 12 \times 12$) instances for training; 1,944 ($81 \times 2 \times 12$) instances, 5,760 ($40 \times 12 \times 12$) instances, and 960 ($40 \times 2 \times 12$) instances for testing in temporal, spatial, and spatiotemporal prediction scenarios, respectively.

CCHF cases had been observed frequently during hot months (e.g., May, June, and July), moderately during warm months (e.g., April, August, and September) and rarely during cold months (e.g., October, November, December, January, February, and March). We encoded each time period by three temporal covariates: the year, month, and seasonal group (i.e., hot, warm, or cold) it belongs to.

Latitude and longitude coordinate information of province centers were used as spatial covariates, and each time period is encoded with its year, month, and season information. The model had 10 parameters to learn, namely, the noise variance σ_y and nine kernel weights, which are the weights of the kernels of individual spatial features `Lat.` and `Lon.`, the weights of the kernels of individual temporal features `Year`, `Month`, and `Season`, the weight of the spatial pairwise interaction kernel `Lat.` × `Lon.`, and the weights of the temporal pairwise interaction kernels `Year` × `Month`, `Year` × `Season`, and `Month` × `Season`.

The spatial interaction kernel had the highest weight in all of the prediction scenarios, approximately one in spatial and spatiotemporal scenarios (see Figure 3). For spatial and spatiotemporal scenarios, the month feature was the most informative temporal covariate with coefficient about 0.5, whereas the year feature was the least informative temporal covariate. On the other hand, for temporal prediction scenario, temporal pairwise interaction kernel weights were mostly significantly larger than the weights of kernels of individual features, contrary to the results for spatial and spatiotemporal prediction scenarios. We note that interactions of the season feature with the other features were more important in temporal prediction scenario.

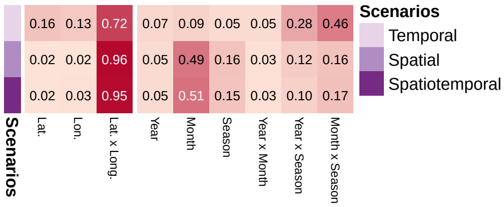

Figure 3: Averaged kernel weights found by SGP2MKL on CCHF data set.

Table 1 reports PCC and NRMSE values for temporal prediction scenario. The proposed SGP2MKL performed best, and RFR was the worst in terms of both PCC and NRMSE. SGP and SGP2MKL had comparable results, but RFR and BRT were quite separated especially in NRMSE values. Performance comparison for spatial and spatiotemporal scenarios are given in Figure 4. SGP2MKL had the best result followed by SGP. RFR performed better than BRT, contrary to the temporal scenario results. We observed a consistent ranking in all of the prediction scenarios, where SGP2MKL outperformed all other methods.

Table 1: Pearson's correlation coefficients (PCC) and normalized root mean squared errors (NRMSE) of four algorithms on CCHF data for temporal prediction scenario together with ranks in parentheses.

Algorithm	PCC	NRMSE
RFR	0.7480 (4)	0.8754 (4)
BRT	0.8460 (3)	0.7465 (3)
SGP	0.9027 (2)	0.4364 (2)
SGP2MKL	0.9124 (1)	0.4131 (1)

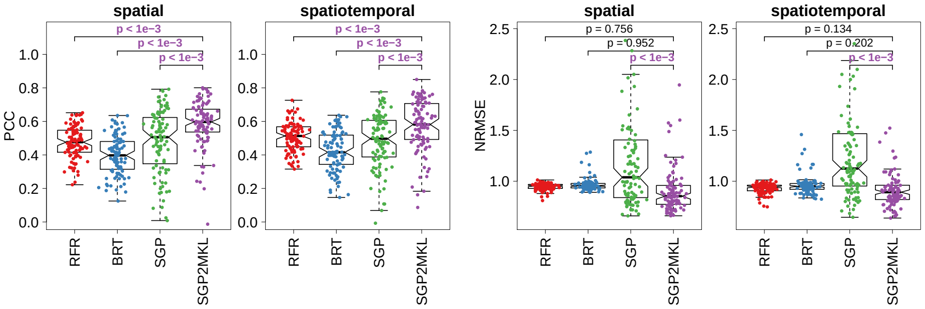

Figure 4: Pearson's correlation coefficients (PCC) and normalized root mean squared errors (NRMSE) of four algorithms on CCHF data set for spatial and spatiotemporal prediction scenarios. SGP2MKL was compared against each competitor using a two-sided paired *t*-test to check whether the predictive performances were statistically significantly different, and *P*-value for each comparison was also reported. If the *P*-value is less than 0.05, it is typeset with the color of the winning algorithm.

Figure 5 shows the comparison between observed and predicted cases of years 2014 and 2015 for temporal scenario (monthly predictions are summed over each province for illustration purposes). For most of the provinces, the predicted case counts are very close to the observed case counts, which shows that SGP2MKL was able to capture the temporal dynamics of the disease.

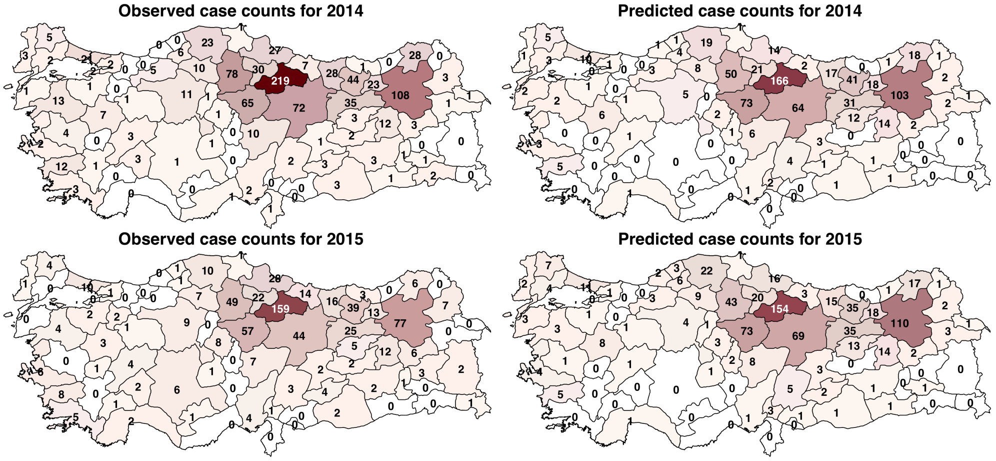
Observed case counts for 2014
Predicted case counts for 2014
Observed case counts for 2015
Predicted case counts for 2015

Figure 5: Country-wide observed versus predicted case counts of years 2014 and 2015 for temporal scenario. Observed and predicted case counts of 81 provinces aggregated yearly after prediction for illustration purposes.

4.2. Predicting Monthly Average of Surface Temperature

We used monthly average surface temperature observations from January 1995 to December 2000 in Central America. This data set comes from the NASA 2007 data expo, http://stat-computing.org/dataexpo/2006/, which contains geographic and atmospheric measures on a very coarse 24 by 24 grid covering Central America (see Figure 6). Thus, there are 576 spatial locations and 72 time periods.

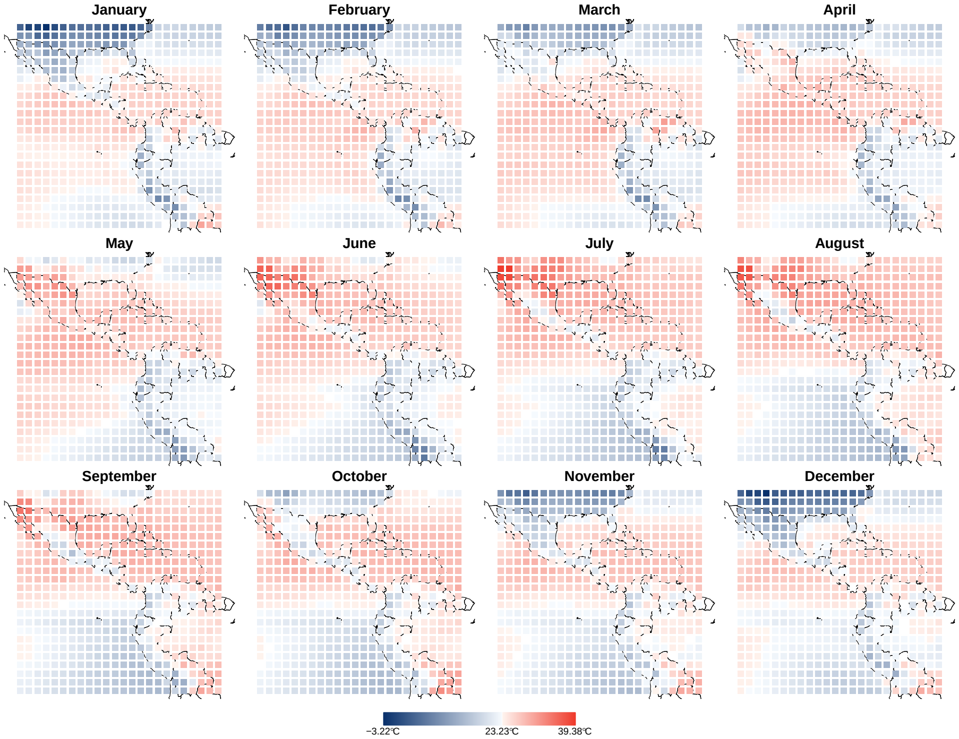

Figure 6: Observed monthly averages of surface temperature on 24 by 24 grid locations between years 1995 and 2000 over the central America. Here, we show the mean of monthly averages in each grid location over all years. We color the overall mean temperature (23.23 °C) with white, and temperatures lower (higher) than this mean with blue (red).

The first five years (i.e., 1995–1999) were used as the temporal training set, and the last year (i.e., 2000) was the test set. Half of the 576 spatial regions were randomly chosen as spatial training set, and the remaining 288 regions were the spatial test set. Hence, we had 34,320 (572×5×12) instances, 20,736 (288×6×12) instances, and 17,280 (288×5×12) instances for training; 6,864 (572×1×12) instances, 20,736 (288×6×12) instances, and 3,456 (288×1×12) instances for testing in temporal, spatial, and spatiotemporal prediction scenarios, respectively.

Latitude and longitude coordinate information of regional centers were used as spatial covariates, and year and month information of each time period were used as temporal covariates. Thus, the model had seven parameters to learn, namely, the noise deviation σ_y and six kernel weights, which are the weights of the kernels of individual spatial features `Lat.` and `Lon.`, the weights of the kernels of individual temporal features `Year` and `Month`, the weight of the spatial pairwise interaction kernel `Lat.` × `Lon.`, and the weight of the temporal pairwise interaction kernel `Year` × `Month`.

Learned kernel weights are shown in Figure 7. Spatial interaction kernels had the highest weights, approximately one in all scenarios. Month feature had the first rank among the temporal covariates with weights between 0.7 and 0.8, and year feature had the least weight, i.e., almost zero, in all scenarios.

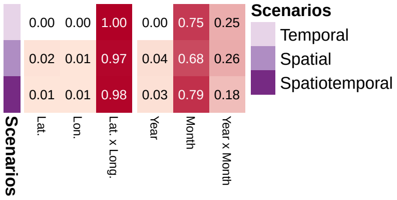

Figure 7: Averaged kernel weights found by SGP2MKL on NASA's surface temperature data set.

Table 2 reports PCC and NRMSE values for temporal prediction scenario. Our proposed method SGP2MKL performed best followed by SGP, and RFR was the worst in terms of both metrics. SGP and SGP2MKL were comparable in NRMSE values. Figure 8 shows PCC and NRMSE values for spatial and spatiotemporal scenarios. SGP2MKL had the best results followed by SGP. RFR performed better than BRT in terms of PCC values contrary to the temporal scenario results, but its NRMSE values were significantly the worst.

Table 2: Pearson's correlation coefficients (PCC) and normalized root mean squared errors (NRMSE) of four algorithms on NASA's surface temperature data for temporal prediction scenario together with ranks in parentheses.

Algorithm	**PCC**	**NRMSE**
RFR	0.8328 (4)	0.7019 (4)
BRT	0.8499 (3)	0.5286 (3)
SGP	0.8856 (2)	0.5068 (2)
SGP2MKL	0.9071 (1)	0.4975 (1)

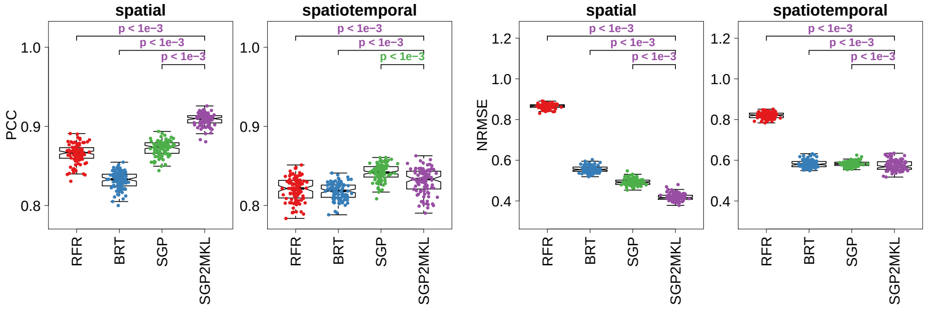

Figure 8: Pearson's correlation coefficients (PCC) and normalized root mean squared errors (N-RMSE) of four algorithms on NASA's surface temperature data set for spatial and spatiotemporal prediction scenarios. SGP2MKL was compared against each competitor using a two-sided paired t-test to check whether the predictive performances are statistically significantly different, and P-value for each comparison was also reported. If the P-value is less than 0.05, it is typeset with the color of the winning algorithm.

5. Conclusions

We proposed a joint framework that couples SGP and MKL. By doing this, we were able to benefit from the special structure of kernel matrices to increase efficiency and from the kernel weights in MKL to increase interpretability. We were able to improve the predictive accuracy of SGP and to provide greater insight about which components are more informative thanks to the MKL component.

We used two data sets from two different domains to show the validity of our proposed method SGP2MKL in real-life applications. Infectious diseases, especially vector borne-diseases, and surface temperature have strong spatial and temporal dependencies, due to the environmental factors. If we are able to learn these dependencies and integrate them into our model, we would be able to improve our characterization of the disease and the temperature dynamics to develop even better tools for forecasting.

In this study, we tried to understand if the geographical dependency is affected by the latitude or longitude information or both. We noted that latitude and longitude define spatial dynamics usually together. Similarly, for temporal features, we investigated year, month, and season information and found out that month information alone is strong enough for the temporal dynamics for these particular data sets except, in some experiments, season information may be needed along with the month information (e.g., temporal prediction scenario of CCHF). We showed that our proposed method SGP2MKL improved predictive accuracy over the alternatives in all experiments.

The use of spatiotemporal modeling tools might help us better understand the characteristics of diseases to develop different types of interventions to prevent and treat vector-borne diseases, such as vector or larva control, or timely treatment (World Health Organization, 2014). The success of such interventions depend on how well the case counts can be predicted and how fast the health care policy makers react to it. Within this context, mathematical modeling can be a powerful companion for decision making and health care services planning. Our proposed method SGP2MKL can be used for modeling infectious diseases other than CCHF.

The decomposition approach we used over two separate feature sets (e.g., locations and time periods in our case) is applicable to many different problems in different domains such as econometrics, gene expression, geostatistics, ensemble learning, multi-output regression, time series, image repainting, texture extrapolation, and video extrapolation.

Acknowledgments

Mehmet Gönen was supported by the Turkish Academy of Sciences (TÜBA-GEBİP; The Young Scientist Award Program) and the Science Academy of Turkey (BAGEP; The Young Scientist Award Program).

References

Antti Airola and Tapio Pahikkala. Fast Kronecker product kernel methods via generalized vec trick. *IEEE Transactions on Neural Networks and Learning Systems*, in press.

Ricardo Andrade-Pacheco. *Gaussian Processes for Spatiotemporal Modelling.* PhD thesis, The University of Sheffield, 2015.

Ricardo Andrade-Pacheco, Martin Mubangizi, John Quinn, and Neil Lawrence. Consistent mapping of government malaria records across a changing territory delimitation. *Malaria Journal*, 13(Suppl 1):P5, 2014.

Samir Bhatt, Peter W. Gething, Oliver J. Brady, Jane P. Messina, Andrew W. Farlow, Catherine L. Moyes, John M. Drake, John S. Brownstein, Anne G. Hoen, Osman Sankoh, Monica F. Myers, Dylan B. George, Thomas Jaenisch, G. R. William Wint, Cameron P. Simmons, Thomas W. Scott, Jeremy J. Farrar, and Simon I. Hay. The global distribution and burden of dengue. *Nature*, 496(7446):504–507, 2013.

Samir Bhatt, Ewan Cameron, Seth R. Flaxman, Daniel J. Weiss, David L. Smith, and Peter W. Gething. Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalisation. *Journal of the Royal Society Interface*, 14(134):20170520, 2017.

Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, pages 153–160, 2007.

Niya Chen, Zheng Qian, Xiaofeng Meng, and Ian T. Nabney. Short-term wind power forecasting using Gaussian processes. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2790–2796, 2013.

Önder Ergönül. Crimean–Congo haemorrhagic fever. *Lancet Infectious Diseases*, 6(4):203–214, 2006.

Andrew O. Finley, Sudipto Banerjee, Patrik Waldmann, and Tore Ericsson. Hierarchical spatial modeling of additive and dominance genetic variance for large spatial trial datasets. *Biometrics*, 65(2):441–451, 2009.

Elad Gilboa, Yunus Saatci, and John P. Cunningham. Scaling multidimensional inference for structured Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):424–436, 2015.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

Simon I. Hay, Dylan B. George, Catherine L. Moyes, and John S. Brownstein. Big data opportunities for global infectious disease surveillance. *PLoS Medicine*, 10(4):e1001413, 2013.

Michael J. Kane, Natalie Price, Matthew Scotch, and Peter Rabinowitz. Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics*, 15(1):276, 2014.

Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood — Computing Hilbert space expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, pages 244–252, 2013.

Andy Liaw and Matthew Wiener. *randomForest: Breiman and Cutler's random forests for classification and regression*, 2015. R package version 4.6-12.

Jaakko Luttinen and Alexander Ilin. Efficient Gaussian process inference for short-scale spatio-temporal modeling. In *Prooceedings of the 15th international conference on Artificial Intelligence and Statistics*, pages 741–750, 2012.

Linh Nguyen, Guoqiang Hu, and Costas J. Spanos. Spatio-temporal environmental monitoring for smart buildings. In *Proceedings of the 13th IEEE International Conference on Control and Automation*, pages 277–282, 2017.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

Greg Ridgeway. *gbm: Generalized Boosted Regression Models*, 2017. R package version 2.1.3.

Jaakko Riihimäki and Aki Vehtari. Laplace approximation for logistic Gaussian process density estimation and regression. *Bayesian Analysis*, 9(2):425–448, 2014.

Yunus Saatçi. *Scalable Inference for Structured Gaussian Process Models.* PhD thesis, University of Cambridge, 2011.

Simo Särkkä and Jouni Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 993–1001, 2012.

Ransalu Senanayake, Simon O. Callaghan, and Fabio Ramos. Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, pages 3901–3907, 2016.

Oliver Stegle, Christoph Lippert, Joris Mooij, Neil Lawrence, and Karsten Borgwardt. Efficient inference in matrix-variate Gaussian models with iid observation noise. In *Advances in Neural Information Processing Systems 24*, pages 630–638, 2011.

Jarno Vanhatalo, Ville Pietilainen, and Aki Vehtari. Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine*, 29(15):1580–1607, 2010.

Ravi Varadhan. *alabama: Constrained nonlinear optimization*, 2015. R package version 2015.3-1.

Yali Wang and Brahim Chaib-draa. A KNN based Kalman filter Gaussian process regression. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1771–1777, 2013.

Andrew Gordon Wilson, Galboa Elad, Arye Nehorai, and John P. Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems 27*, pages 3626–3634, 2014.

World Health Organization. *World Malaria Report 2014*. Geneva, 2014.