Spatiotemporal Modeling Using Machine Learning

by

Çiğdem Ak

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Computational Sciences and Engineering

KOÇ
UNIVERSITY

2019

Spatiotemporal Modeling Using Machine Learning

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Çiğdem Ak

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Mehmet Gönen (Advisor)

_____

Prof. Önder Ergönül (Co-Advisor)

_____

Prof. Füsun Can

_____

Assoc. Prof. Mehmet Sayar

_____

Assoc. Prof. Arzucan Özgür

_____

Asst. Prof. Mustafa Gökçe Baydoğan

Date: _____

To my dearest family.

ABSTRACT

An ideal machine learning algorithm for spatiotemporal modeling should be able (i) to integrate both temporal and spatial data from different sources, (ii) to discover patterns and, (iii) to make inference, without human intervention. Gaussian processes provide a Bayesian framework for analyzing spatiotemporal data, which were used widely to estimate values across space and time, yet their computational and storage complexity have been a limiting factor when it comes to application. In this thesis, we proposed computational frameworks that integrate Gaussian processes into spatiotemporal modeling scenarios with a particular focus on scalable inference by exploiting the structure of the covariance matrix generated by matrix multiplication of spatial and temporal covariance matrices. We also aimed to increase the interpretability using kernel methods, which have deep connections with spatial statistics. With the combination of multiple kernel learning and structured Gaussian processes, we increased both accuracy and expressiveness of the inference. We showed the power of these methods on real-world regression data sets. Our proposed methods were applied to a spatiotemporal data set of a vector-borne disease using official patient records in Turkey. We showed our proposed machine learning algorithms were better than their counterparts in terms of accuracy. In addition, our developed methods are also more interpretable, which means that they are able to answer questions drawn from the domain of public health and give insight to policy makers for quick response planning and resource allocation.

ÖZETÇE

Uzam-zamansal modelleme için ideal bir yapay öğrenme algoritması, (i) farklı kaynaklardan hem uzamsal hem de zamansal verileri bir araya getirerek bütünleştirebilmeli, hiç bir insan müdahalesi olmadan (ii) model yapısını ortaya çıkarabilmeli ve (iii) çıkarımda bulunabilmelidir. Gauss süreçleri uzam-zamansal verileri analiz etmek için Bayesçi bir yaklaşım sağlarlar, fakat hesaplama hızı ve saklama alanı karmaşıklıkları, pratikte sınırlayıcı olmuştur. Bu tezde, uzamsal ve zamansal kovaryans matrislerinin çarpımlarıyla oluşan kovaryans matrisinin yapısından faydalanarak, ölçekli çıkarım üzerinde odaklanarak, Gauss süreçlerini uzam-zamansal modelleme senaryolarına entegre eden hesaplamalı methodlar önerdik. Uzamsal istatistikle derin bağları olan çekirdek methodlarını kullanarak, yorumlanabilirliği arttırmayı da amaçladık. Çoklu çekirdek öğrenmesinin yapısal Gauss süreçleriyle birleşmesiyle, çıkarımların hem doğruluğunu hem de ifade gücünü arttırdık. Bu methodların kabiliyetlerini geçek regresyon veri kümeleri üzerinde gösterdik. Önerdiğimiz methodları, Türkiye'nin resmi hasta kayıtlarından oluşan, vektörle bulaşan bir hastalığın uzam-zamansal veri kümesine uyguladık. Önerdiğimiz algoritmaların emsallerinden doğruluk açısından daha iyi olduklarını gösterdik. Ayrıca, geliştirdiğimiz methodlar daha açıklayıcıdır, yani, halk sağlığı alanından gelen soruları cevaplayabilirler ve karar alması gereken kişilere planlamada ve kaynakların atanmasında çabuk müdahale için fikir verebilirler.

ACKNOWLEDGMENTS

I would like to thank first of all my supervisors Assoc. Prof. Mehmet Gönen and Prof. Önder Ergönül for all of their advise. I feel extremely fortunate to have worked with both of them who are such inspiring advisors with never-ending energy and motivation. I am thankful to Prof. Önder Ergönül for believing in me and made this interdisciplinary work possible in the first place. I cannot express my deep and sincere gratitude enough to Assoc. Prof. Mehmet Gönen for teaching me how to do research and also how to achieve in life as well. The completion of this thesis would not be possible without his invaluable guidance and support.

I would like to acknowledge the jury members and all professors who gave their time to discuss and give advise Assist. Prof. Mustafa Gökçe Baydoğan, Prof. Füsun Can, Assoc. Prof. Elvan Ceyhan, Prof. Ayşen Gargılı, Assoc. Prof. Arzucan Özgür, and Assoc. Prof. Mehmet Sayar.

Thanks to the members of the Machine Intelligence and Data Analysis in Science Laboratory (MIDASLAB) and the colleagues that I have met during my PhD who made my time memorable at Koç. It is not possible to mention all the names here but I want to thank especially Veli Oğuzalp Bakır, Onur Dereli, Angi Nazih Ghanem, Serap Gümüş, Banu Ulusoy. I also would like to thank my longtime friends and my lovely flate-mates for their support and encouragement both in academic and social life Gizem Özdermir, Yiğit Özel, Tuğba Uçar, Suraya Yusof, and Elif Yunt.

Finally, I would like to thank my mother Fatma, my sister Gizem, and my father Tarık Ak for their support during all these years. I would like to dedicate this thesis to my family, particularly to my mother who always supported and advocated my education more than anything. I would not be here writing this very sentence if it were not for them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Abbreviation	Meaning
BRT	Boosted Regression Tree
CCHF	Crimean–Congo Hemorrhagic Fever
GP	Gaussian Process
GPR	Gaussian Process Regression
MKL	Multiple Kernel Learning
NASA	National Aeronautics and Space Administration
NRMSE	Normalized Root Mean Squared Error
PCC	Pearson's Correlation Coefficient
RFR	Random Forest Regression
SGP	Structured Gaussian Process
SGP2MKL	Structured Gaussian Process with Twin Multiple Kernel Learning
SVD	Singular Value Decomposition

Notation	Meaning
$\lvert\mathbf{K}\rvert$	Determinant of matrix $\mathbf{K}$
$\circ$	Hadamard product
$\otimes$	Kronecker product
$\lVert\boldsymbol{x}\rVert_2$	The ℓ_2-norm of vector $\boldsymbol{x}$
$\propto$	Proportional to
$\sim$	Distributed according to
$\boldsymbol{x}^\top$	Transpose of vector $\boldsymbol{x}$

Notation	Meaning
$\mathbf{1}$	Vector of all ones
D	Number of features
$\mathcal{D}$	Data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
$\det(\mathbf{K})$	Determinant of matrix $\mathbf{K}$
$\text{diag}(\mathbf{D})$	Vector containing the diagonal elements of matrix $\mathbf{D}$
$\text{E}[\boldsymbol{x}]$	Expectation of the random variable $\boldsymbol{x}$
$\boldsymbol{f}$	Gaussian process or vector of latent function values, $\boldsymbol{f} = [f(\boldsymbol{x}_1) \cdots f(\boldsymbol{x}_N)]^\top$
$\mathcal{GP}$	Gaussian Process: $\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$, the function $\boldsymbol{f}$ is distributed as a Gaussian process with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{K}$
$\mathbf{I}$	Identity matrix
$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Kernel function between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
$\mathbf{K}$	Kernel matrix: kernel function evaluated at all pairs of training data, $\{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i=1,j=1}^{N,N}$
$\log(\cdot)$	Natural logarithm (base e)
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$
N	Number of data points
$\mathcal{O}(\cdot)$	Big Oh; $f(n) = \mathcal{O}(g(n))$ for the functions f and g on the domain of natural numbers, if the ratio $f(n)/g(n)$ remains bounded for sufficiently large values of n
$p(x)$	Probability density function of random variable x
$p(y\|x)$	Conditional probability density function of random variable y given x
$\phi(\boldsymbol{x})$	Feature map of the data point $\boldsymbol{x}$
$\mathbb{R}^N$	Real $N \times 1$ vectors

Notation	Meaning
σ_y^2	Noise variance
$\boldsymbol{\theta}$	Vector of hyper-parameters
$\text{tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$
$\boldsymbol{w}$	Weight coefficients
$\mathbf{X}$	$D \times N$ matrix of data points, $\mathbf{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N]$
$\boldsymbol{x_i}$	i-th data point
$x_\star$	Test data point
$\text{Var}[\boldsymbol{x}]$	Variance of the random variable $\boldsymbol{x}$
y	Output value of the input data point $\boldsymbol{x}_i$
$\hat{y}$	Predicted value of y for the data point $\boldsymbol{x}_i$

Chapter 1

INTRODUCTION

An observation in a spatiotemporal data set is a spatial and temporal event that exists at a certain time and location. In other words, spatiotemporal means being dependent on both space and time. Spatiotemporal modeling helps solving problems that we want to examine and predict how observations vary over space and/or time. Such problems exist at very different spatial and temporal scales, for example, tracking of moving objects, weather forecasting, medical imaging, monitoring ecosystems, and mapping disease spread over time. Spatiotemporal analyses have more power over only spatial or time-series analyses with the inclusion of space-time interaction terms.

The dependency of a time series generally defined with patterns such as trends (i.e., long term change in the mean of the observations), cyclic effects (i.e., variations around the trend, e.g., seasonal effect), or irregular fluctuations (i.e., residual variation). We can define the spatial dependence by Tobler's first law of geography [Tobler, 1970]:

"Everything is related to everything else, but near things are more related than distant things".

We may not simulate the real life processes identically, but we can approximate very closely to the reality. Even though we may not be able to explain the complete process at the end, we may still learn some patterns, which may be useful for understanding the process and making predictions. Main assumption is that the future (e.g., an unseen sample data point) will not be very different from the past (e.g., the collected sample data).

Machine learning makes inference from observations by building mathematical models and/or using statistics. Nevertheless, machine learning is not just making use

of data, but also being intelligent, which should have the ability to learn in changing circumstances. To be able to learn and make inference without human intervention, the model should be discovering patterns, and extrapolating them to new conditions. When the model learns and adapts to such changes, the modeler need not to think for every possible case and specify these in the model/algorithm.

Statistics and machine learning communities have most of the base theories and many algorithms in common. The principal differences are the motivation for modeling, the type of problems tackled, and the questions asked for learning. Broadly speaking, main focus in statistics is usually understanding the data and the relationships. On the other hand, the principal aim in machine learning algorithms is to make very accurate predictions and find out the behavior of learning algorithms.

These two fields evolved around different purposes mentioned above. For instance, in machine learning, even though the prediction power of neural network algorithms has been proven excessively, many statisticians are not satisfied because these models perform as a black-box and interpreting them is difficult.

Gaussian process (GP) satisfies the objective of two communities. GP offers a probabilistic, practical, and principled approach, thus GPs may be easier to handle and interpret unlike the its couterparts e.g., neural networks. For example, neural networks are not very easy to apply in practice because of many reasons. One needs to decide the number of hidden layers, number of neurons in each layer, the activation functions, the learning rate, etc. In addition, there is a lack of a principled framework to answer these questions. It also has been shown that Gaussian processes are mathematically equivalent to large neural networks (under some conditions) [Neal, 2012]

In this thesis, we are concerned with solving spatiotemporal regression problems using Gaussian process. Regression is a problem of learning input-(numeric) output mappings from training data set. In the next section, we describe Gaussian process regression in detail.

This thesis contains work from three publications:

In the first publication [Ak et al., 2018a], we developed a computational framework based on Gaussian processes to perform spatiotemporal prediction. Vanilla Gaussian processes 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. Therefore, we exploited the special structure of similarity matrices in our formulation to obtain a very efficient implementation. We then tested our framework on the problem of modeling Crimean–Congo hemorrhagic fever cases between years 2004 and 2015 in Turkey.

We showed that our Gaussian process formulation obtained better results than two frequently used standard machine learning algorithms (i.e., random forests and boosted regression trees) under temporal, spatial, and spatiotemporal prediction scenarios. These results showed that our framework has the potential to make an important contribution to public health policy makers.

Infectious diseases cause important health problems worldwide and create difficult challenges for public health policy makers. That is why they need reliable computational tools to better understand disease and to predict case counts. They will benefit from such computational tools to make more informed decisions in developing control and prevention strategies. We formulated a computational framework that can be used to model spatial, temporal, or spatiotemporal dynamics of infectious diseases. We showed the utility of our framework on the problem of modeling Crimean–Congo hemorrhagic fever (CCHF) in Turkey.

In the second paper [Ak et al., 2018b], 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.

In the third study [Ak et al., 2019], we aimed to develop a prospective prediction tool on CCHF to identify geographic regions at risk. The tool could support public health decision makers in implementation of an effective control strategy in a timely manner. We used monthly surveillance data between 2004 and 2015 to predict case counts between 2016 and 2017 prospectively. Turkish nationwide surveillance data set collected by Ministry of Health contained 10,411 confirmed CCHF cases. We collected potential explanatory covariates about climate, land use, and animal and human population at risk to capture spatiotemporal transmission dynamics. We developed a structured GP algorithm and prospectively tested this tool predicting the future year's cases given past years' cases. We were also able to rank the covariates with respect to their relative importance.

We predicted the annual cases in 2016 and 2017 as 438 and 341, whereas the observed cases were 432 and 343, respectively. Pearson's correlation coefficient and normalized root mean squared error values for 2016 and 2017 predictions were (0.83; 0.58) and (0.87; 0.52), respectively. The most important covariates were found to be the number of settlements with fewer than 25,000 inhabitants, latitude, longitude, and potential evapotranspiration (evaporation and transpiration).

Main driving factors of CCHF dynamics were human population at risk in rural areas, geographical dependency, and climate effect on ticks. Our model was able to prospectively predict the numbers of CCHF cases. Our proof of concept study also provided insight for understanding possible mechanisms of infectious diseases and found the important directions for practice and policy to combat against emerging infectious diseases.

In the following sections of this introduction chapter, we define GP regression problem with two different points of views and also describe multiple kernel learning as preliminary for the next sections. Then, one chapter is dedicated to each publication mentioned above. Finally, we conclude with a conclusion chapter where we also make a mention of possible future studies.

1.1 Gaussian Process

GP is a Bayesian machine learning approach. It has 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, 2018], 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].

GP is a stochastic process, which places a prior over the function space. A stochastic process is roughly a generalization of a probability distribution to functions. One advantage is that computations required for inference and learning is easier for GPs. The main modeling power of GPs is the covariance/kernel function. The kernel function k directly specifies the covariance between a pair of data points: $k(\boldsymbol{x}, \boldsymbol{x}') = cov(f(\boldsymbol{x}), f(\boldsymbol{x}'))$. In other words, kernel functions define the nature of the resulting models (e.g., linear, periodic, etc.). This means that we do not need to modify our algorithm considerably to change the model but the covariance function. We can also learn properties of the problem modeled by estimating the hyper-parameters of the kernels.

In this thesis, we use the terms kernel functions and covariance functions interchangeably. An example of a kernel function is the covariance function of a GP. For a kernel $k(x, x')$ to be a covariance function, any matrix $\mathbf{K}$ with elements $\mathbf{K}_{ij} = k(x_i, x_j)$ must be positive semi-definite (i.e., $\boldsymbol{z}^\top \mathbf{K} \boldsymbol{z} \geq 0$ for all $\boldsymbol{z} \in \mathbb{R}^N$). This is satisfied because the covariance matrix in a multivarite Gaussian distribution must be positive semi-definite.

There are more than one way of interpreting GP regression (GPR). There is an alternative way to the definition we gave above (i.e., GP is a prior over functions), which is easier to grasp initially. That is why we start with this easier description,

weight–space view. The main idea of the weight–space view subsection is that, we can construct a GP defined solely in terms of scalar products in input space by extending the Bayesian linear model with the inputs, which are projected into a high-dimensional feature space. Then, we continue with the function space view in second subsection.

1.1.1 Bayesian Linear Regression with Infinite Basis Functions

In this subsection, we start with Bayesian treatment of a standard linear regression problem, then apply linear regression after projecting the inputs into a high–dimensional feature space. Finally applying the kernel trick, we show that the resulting model is a GP.

We denote the input as $\boldsymbol{x}$ and the output as y. In general, the input is represented as a vector $\boldsymbol{x}$ because there may be more than one feature of an input. The output y is continuous since we will work on regression problems. For a given data set of N observations, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, we want to make predictions for a new input $\boldsymbol{x}_\star$, which is not observed yet. This is a problem of finding the function f that makes predictions for all possible inputs. We model observations as noisy realizations of a linear combination of the features:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{w} + \boldsymbol{\xi},$$

where $\mathbf{X}$ is a $D \times N$ matrix of the training inputs $\{\boldsymbol{x}_i\}_{i=1}^N$, where D is the dimension of the input space, $\boldsymbol{w}$ is a vector of weights (i.e., parameters) of the linear model, and $\boldsymbol{\xi}$ is the vector of noise variables. We assume that the vector of measurement noise values follow an isotropic multivariate normal distribution with the zero mean vector and the variance matrix $\sigma_y^2\mathbf{I}$, that is

$$p(\boldsymbol{\xi}) \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\mathbf{I}).$$

The linear model with the noise assumption leads to the probability density of the observations given the inputs and the parameters as follows:

$$p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \sigma_y^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{w}, \sigma_y^2\mathbf{I}). \tag{1.1}$$

In Bayesian approaches, a prior is specified over the parameters. This is our beliefs about the model before seeing the training data. We put a Gaussian prior over model parameters with covariance matrix $\mathbf{\Sigma_w}$, that is

$$p(\boldsymbol{w}) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_w}).$$

We can calculate the posterior probability distribution over the weights using the Bayes' rule:

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|\mathbf{X})}, \tag{1.2}$$

where $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$ is the posterior density (i.e., the distribution over parameters after observing data), $p(\boldsymbol{y}|\mathbf{X})$ is the likelihood (i.e., measure of fitness), $p(\boldsymbol{w})$ is the prior density (i.e., anything we know about parameters before we see any data), and $p(\boldsymbol{y}|\mathbf{X})$ is the marginal likelihood, which is independent of weights hence is a normalization constant that ensures $\int p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{w} = 1$.

Ignoring the non-$\boldsymbol{w}$ terms, the right hand side of Equation (1.2), the likelihood multiplied by the prior is

$$\begin{aligned}
\text{posterior} &\propto p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w}) \\
&\sim \exp\left(-\frac{1}{2\sigma_y^2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{w})\right)\exp\left(-\frac{1}{2}\boldsymbol{w}^\top {\mathbf{\Sigma_w}}^{-1}\boldsymbol{w}\right) \\
&\sim \exp\left(-\frac{1}{2}\left(\boldsymbol{w}^\top\left[\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X} + {\mathbf{\Sigma_w}}^{-1}\right]\boldsymbol{w} - \frac{2}{\sigma_y^2}\boldsymbol{w}^\top\mathbf{X}^\top\boldsymbol{y}\right)\right).
\end{aligned}$$

Because both prior and likelihood are Gaussians, the posterior is also found to be a Gaussian distribution with the mean $\boldsymbol{\mu}$ and the covariance $\mathbf{\Sigma}$. Ignoring normalizing constants, the left hand side of Equation (1.2) is

$$\begin{aligned}
p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \sigma_y^2) &\propto \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\boldsymbol{w} - \boldsymbol{\mu})\right) \\
&\sim \exp\left(-\frac{1}{2}(\boldsymbol{w}^\top\mathbf{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu} + \boldsymbol{\mu}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu})\right) \\
&\sim \exp\left(-\frac{1}{2}(\boldsymbol{w}^\top\mathbf{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu})\right).
\end{aligned}$$

Equating both sides, we have

$$\exp\left(-\frac{1}{2}(\boldsymbol{w}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{w}-2\boldsymbol{w}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right)=\exp\left(-\frac{1}{2}\left(\boldsymbol{w}^\top\left[\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}+{\boldsymbol{\Sigma}_\mathbf{w}}^{-1}\right]\boldsymbol{w}-\frac{2}{\sigma_y^2}\boldsymbol{w}^\top\mathbf{X}^\top\boldsymbol{y}\right)\right).$$

By equating individual terms on each side, we obtain the mean and the covariance as follows

$$\boldsymbol{\Sigma} = \left(\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}+{\boldsymbol{\Sigma}_\mathbf{w}}^{-1}\right)^{-1}, \tag{1.3}$$

$$\boldsymbol{\mu} = \frac{1}{\sigma_y^2}\boldsymbol{\Sigma}\mathbf{X}^\top\boldsymbol{y}. \tag{1.4}$$

To conclude, we showed that the posterior probability distribution over the weights is Gaussian with mean and covariance given in Equations (1.3)-(1.4), that is $p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y},\sigma_y^2)=\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$. Now, we can make prediction for a test point by averaging Equation (1.1) over all possible parameter values weighted by their posterior, as follows

$$\begin{aligned} p(y_\star|\mathbf{X},\boldsymbol{y},\boldsymbol{x}_\star,\sigma_y^2) &= \int p(y_\star|\mathbf{X},\boldsymbol{y},\boldsymbol{x}_\star,\sigma_y^2,\boldsymbol{w})p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y},\sigma_y^2)d\boldsymbol{w} \\ &= \int \boldsymbol{x}_\star^\top\boldsymbol{w}p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y})d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{x}_\star^\top\boldsymbol{\mu},\sigma_y^2+\boldsymbol{x}_\star^\top\boldsymbol{\Sigma}\boldsymbol{x}_\star). \end{aligned}$$

The predictive distribution is also Gaussian. The mean is the test input multiplied by the mean of the posterior of the weights (Equation (1.3)). The variance is a quadratic form of the test input with the covariance of the posterior of the weights (Equation (1.4)).

The problem with the Bayesian linear model is, it is not capable of modeling accurately when the data is not following a linear model, which is often the case in real life applications.

To overcome this difficulty, we create more features by projecting the inputs to a higher dimensional feature space, then perform the linear model on the new features created instead of applying linear model only on original inputs. For example, suppose the input x is scalar (i.e., one dimensional) and by projecting it into the space of powers of x, $\phi(x)=[1\ x\ x^2\ x^3\ \ldots\ ]^\top$ and, then, implementing a linear model on

the new vector $\phi(x)$ is just a polynomial regression, which allows us to use a linear model for non-linear data. The model is still linear in parameters because projection is independent of $\boldsymbol{w}$. Now, we generalize this example. Imagine transforming the inputs using a set of P functions, $\boldsymbol{x} \rightarrow \phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \cdots, \phi_P(\boldsymbol{x}))^\top$. The $\phi_\cdot$ functions are also known as basis functions. Define matrix $\boldsymbol{\Phi}$ as follows

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \dots & \phi_P(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \dots & \phi_P(\boldsymbol{x}_N) \end{bmatrix}.$$

After applying Bayesian linear regression on the transformed features, $\boldsymbol{\Phi}$, we have the posterior as $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \sigma_y^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean and covariance as follows

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{\sigma_y^2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top \boldsymbol{y} \\ \boldsymbol{\Sigma} &= \left(\frac{1}{\sigma_y^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + {\boldsymbol{\Sigma}_{\mathbf{w}}}^{-1}\right)^{-1}. \end{aligned}$$

Predictive distribution can be obtained the same way before as follows

$$p(y_\star|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}_\star, \sigma_y^2) = \mathcal{N}(\boldsymbol{\Phi}_\star^\top\boldsymbol{\mu}, \sigma_y^2 + \boldsymbol{\Phi}_\star^\top\boldsymbol{\Sigma}\boldsymbol{\Phi}_\star). \tag{1.5}$$

The problem is that there are infinitely many set of basis functions, polynomials, trigonometric functions, etc. How do we decide which set of basis functions among infinitely many options to use when we are going to apply this method? If we could use all the basis functions possible, our method would be able to model anything, but then the cost of storing and taking the inverse of a matrix of infinite size wouldn't be practically possible since the scalar product $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ in the covariance matrix $\boldsymbol{\Sigma}$, is a matrix of size infinite by infinite. Actually there is no need to choose basis functions because GPs work implicitly with an infinite set of basis functions thanks to kernels and learn a probabilistic combination of these.

We are going to show that predictions can be expressed exclusively in terms of scalar products as follows

$$k(\boldsymbol{x}, \boldsymbol{x}') = \psi(\boldsymbol{x})^\top\psi(\boldsymbol{x}').$$

Replacing the inner products with the kernel function $k(\cdot, \cdot)$ is called kernel trick. This method is advantageous when computing the kernel (i.e., covariance matrix) instead of the feature matrix $\mathbf{\Phi}$. In other words, this allows us to work either $k(\cdot, \cdot)$ or $\psi(\cdot)$. This useful because working with $\psi(\cdot)$ costs $\mathcal{O}(P^2)$ storage, $\mathcal{O}(P^3)$ time. Working with $k(\cdot, \cdot)$ costs $\mathcal{O}(N^2)$ storage, $\mathcal{O}(N^3)$ time. We can pick the one that makes computations faster. However, it is possible to pick $k(\cdot, \cdot)$ so that $\psi(\cdot)$ is infinite dimensional. Now, we are going to demonstrate this for an example of a Gaussian kernel.

It is possible to show that for

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||_2^2}{2s^2}\right)$$

there exists a corresponding $\psi(\cdot)$ that is infinite dimensional.

We prove that the Gaussian kernel induces an infinite dimensional $\psi(\cdot)$ for one dimensional inputs x and x' and $s = 1$ for simplicity, but this proof can easily be extended to higher dimensions and arbitrary s. Expand the Gaussian kernel $k(x, x')$ as

$$\exp\left(-\frac{(x - x')^2}{2}\right) = \exp\left(-\frac{(x)^2}{2}\right) \exp\left(-\frac{(x')^2}{2}\right) \exp(xx').$$

Focusing on the last term and applying the Taylor expansion of the $\exp(\cdot)$ function, we obtain

$$\exp(xx') = 1 + (xx') + \frac{(xx')^2}{2!} + \frac{(xx')^3}{3!} + \frac{(xx')^4}{4!} + \cdots,$$

which completes the proof.

In order to show that Bayesian linear regression can be formulated through scalar products only, we need Woodbury identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \tag{1.6}$$

We can take the inverse in the variance of the posterior by setting $A = \mathbf{\Sigma}_{\mathbf{w}}$, $U = V^\top = \mathbf{\Phi}^\top$, and $C = \frac{1}{\sigma_y^2}\mathbf{I}$ in Equation (1.6) as follows

$$\begin{aligned}\boldsymbol{\Sigma} &= \left(\frac{1}{\sigma_y^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + {\boldsymbol{\Sigma}_\mathbf{w}}^{-1}\right)^{-1} \\ &= \boldsymbol{\Sigma}_\mathbf{w} - \boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\left(\sigma_y^2\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\right)^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}.\end{aligned}$$

Posterior mean $\boldsymbol{\mu}$ can also be rewritten using the same idea because it contains $\boldsymbol{\Sigma}$. Then, from the predictive probability (1.5), we have the following equality as the new covariance expressed with kernels

$$\phi_\star^\top\boldsymbol{\Sigma}_\mathbf{w}\phi_\star - \phi_\star^\top\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\left(\sigma_y^2\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\right)^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}\phi_\star = k_{\star\star} - \boldsymbol{k}_\star^\top(\sigma_y^2\mathbf{I} + \mathbf{K})^{-1}\boldsymbol{k}_\star,$$

where the mapping defining the kernel is

$$\psi(\boldsymbol{x}) = {\boldsymbol{\Sigma}_\mathbf{w}}^{1/2}\phi(\boldsymbol{x})$$

and

$$\begin{aligned}k_{\star\star} &= k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) = \psi(\boldsymbol{x}_\star)^\top\psi(\boldsymbol{x}_\star), \\ (\boldsymbol{k}_\star)_i &= k(\boldsymbol{x}_\star, \boldsymbol{x}_i) = \psi(\boldsymbol{x}_\star)^\top\psi(\boldsymbol{x}_i), \\ (\mathbf{K})_{ij} &= k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \psi(\boldsymbol{x_i})^\top\psi(\boldsymbol{x}_j).\end{aligned}$$

At the end, we have the mean and the variance of the predictions in terms of kernel evaluations as follows:

$$p(y_\star|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}_\star, \sigma_y^2) = \mathcal{N}(\boldsymbol{k}_\star^\top(\sigma_y^2\mathbf{I} + \mathbf{K})^{-1}\boldsymbol{y}, k_{\star\star} - \boldsymbol{k}_\star^\top(\sigma_y^2\mathbf{I} + \mathbf{K})^{-1}\boldsymbol{k}_\star), \tag{1.7}$$

where the size of $\mathbf{K}$ matrix is $N \times N$ and the size of vector $\boldsymbol{k}_\star$ is N and $k_{\star\star}$ is scalar.

In conclusion, we showed that GP is Bayesian linear regression with infinitely many basis functions (1.7).

1.1.2 Gaussian Process Regression Defined as Prior over Functions

For a given training data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, GP regression uses a probabilistic formulation to model the relationship between the input covariates and the output as follows

[Williams and Rasmussen, 2006]:

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

where $\boldsymbol{y} = [y_1\ y_2\ \cdots\ y_N]^\top$ is the vector of observed output values, $\boldsymbol{f} = [f_1\ f_2\ \cdots\ f_N]^\top$ is the vector of underlying true output values for the corresponding input data instances $\mathbf{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_N]$, $\boldsymbol{\xi} = [\xi_1\ \xi_2\ \cdots\ \xi_N]^\top$ is the vector of measurement noise values that are assumed to follow an isotropic multivariate normal distribution with the variance parameter σ_y^2, and $\mathbf{0}$ and $\mathbf{I}$ are the vector of zeros and the identity matrix of proper sizes, respectively.

The true output values $\boldsymbol{f}$ are assumed to follow a multivariate normal distribution with the mean $\mathbf{0}$ and the covariance $\mathbf{K}$, that is

$$\begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_1) \\ k(\boldsymbol{x}_1, \boldsymbol{x}_2) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_1, \boldsymbol{x}_N) & k(\boldsymbol{x}_2, \boldsymbol{x}_N) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \right)$$

where $k(\cdot, \cdot)$ is a kernel function that calculates a similarity measure between two data instances. By integrating out the true output values $\boldsymbol{f}$, it can be shown that the observed output values $\boldsymbol{y}$ have the following form:

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

We could also use the property of sum of two Gaussian distributions, which is also a Gaussian with mean and covariance, that are the sum of the means and the covariances respectively, since $\boldsymbol{y}$ is the sum of $\boldsymbol{f}$ and $\boldsymbol{\xi}$ that are both Gaussian. Using the properties of the multivariate normal distribution, we can find the predictive distribution of an unknown output value $y_\star$ for an unseen data instance $\boldsymbol{x}_\star$. We first

write the joint distribution of $(\boldsymbol{y}, y_\star)$ using marginalization property (see, Appendix, Equation (6.3)) as follows

$$\begin{bmatrix} \boldsymbol{y} \\ y_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K} + \sigma_y^2 \mathbf{I} & \boldsymbol{k}_\star \\ \boldsymbol{k}_\star^\top & k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) + \sigma_y^2 \end{bmatrix} \right),$$

where $\boldsymbol{k}_\star = [k(\boldsymbol{x}_\star, \boldsymbol{x}_1)\ k(\boldsymbol{x}_\star, \boldsymbol{x}_2)\ \cdots\ k(\boldsymbol{x}_\star, \boldsymbol{x}_N)]^\top$.

Then, we can easily find the conditional distribution (see Appendix, Equation (6.2)) of $y_\star$ to obtain its predictive distribution, which is also a multivariate normal distribution with the following mean and variance:

$$\mathrm{E}[y_\star | \boldsymbol{x}_\star, \mathbf{X}, \boldsymbol{y}, \sigma_y^2] = \boldsymbol{k}_\star^\top (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \boldsymbol{y}, \tag{1.8}$$

$$\mathrm{Var}[y_\star | \boldsymbol{x}_\star, \mathbf{X}, \boldsymbol{y}, \sigma_y^2] = k(x_\star, x_\star) - \boldsymbol{k}_\star^\top (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \boldsymbol{k}_\star. \tag{1.9}$$

1.1.3 Computational Burden of Gaussian Process Regression

A big advantage of GPs is that exact predictions can be made with closed-form equations as given in Equations (1.8)-(1.9). This comes with a cost of taking the inverse of the matrix $(\mathbf{K} + \sigma_y^2 \mathbf{I})$ and store it. The computational cost of making a GP regression, given N training data points, is $\mathcal{O}(N^3)$ in execution time and $\mathcal{O}(N^2)$ in memory. One of the most important objective of machine learning is making such algorithms that their computational cost would be acceptable with the growing data size N.

One of the aims of this thesis is to demonstrate that efficient and exact inference is possible for spatiotemporal GP regression by exploiting the special structure of the matrix $\mathbf{K}$. In chapter 2, we construct a special structure for spatiotemporal modeling and explain how we exploit this structure for fast inference.

1.2 Multiple Kernel Learning

In the past years, Multiple Kernel Learning (MKL) methods have been proposed, where multiple kernels are used instead of selecting only one kernel function. In

the previous section, we discussed how we should choose basis functions. Now, we should decide how we chose which kernels to use. Besides there are many kernels with different similarity functions, we may have multiple features from different data sources, thus multiple kernels calculated on these different features. In addition, we can construct new kernel functions from existing ones. For example, multiplication of a valid kernel with a constant is another valid kernel. Summation and multiplication of valid kernels results in valid kernels. Given two data points $\boldsymbol{x}$ and $\boldsymbol{x}'$

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= c\, k_m(\boldsymbol{x}, \boldsymbol{x}'), \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{m=1}^{P} k_m(\boldsymbol{x}, \boldsymbol{x}'), \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \prod_{m=1}^{P} k_m(\boldsymbol{x}, \boldsymbol{x}'), \end{aligned}$$

where $k_m(\cdot, \cdot)$ is a valid kernel for $m = 1, 2, \ldots, P$ and c is constant. With these properties, we now do not have to chose one kernel among all possible kernels. We can take their average for instance or we can take a weighted sum of the possible kernels, that is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{m=1}^{P} \eta_m\, k_m(\boldsymbol{x}, \boldsymbol{x}'),$$

where η_m is the weight of the kernel k_m, $\eta_m \geq 0$ and/or, $\sum_{m=1}^{P} \eta_m = 1$. Using a weighted sum of kernels instead of a single kernel is called multiple kernel learning [Gönen et al., 2011].

It is obvious that combining multiple kernels using different data sources or using different similarity functions is advantageous. However, the features that are being used might not be related to the problem which we want to model and data sources are often noisy. That is why, when combining multiple kernels, we should optimize and learn the kernel weights. We tackle this problem of learning multiple kernels for GPs in the Chapter 3.

Chapter 2

SPATIOTEMPORAL MODELING WITH STRUCTURED GAUSSIAN PROCESSES

In this chapter, we presented and applied the spatiotemporal modeling using structured GPs especially in the context of infectious diseases. Before describing our computational model, we first start with how and why spatiotemporal modeling is useful for infectious diseases and discuss the data sources used in the application.

Infectious diseases constitute a major part of healthcare burden worldwide, leading to millions of deaths annually, which are especially seen among poor and young populations in low and middle income countries [Harris et al., 2012]. In addition to pandemic infectious diseases such as influenza and tuberculosis, there are also emerging infectious diseases such as Ebola virus disease and Zika fever, which require a worldwide effort to combat. Thus, predicting the case counts of infectious diseases is of great importance in developing control and prevention strategies. In particular, there might be spatial dependencies (e.g., humid conditions for malaria) and temporal dependencies (e.g., seasonal effects for influenza) that control the emergence and spread of such diseases [Jones et al., 2008].

To be able to develop protective measures against infectious diseases, it is very important (i) to clearly identify the disease spread and (ii) to make reliable predictions for future cases. When the disease spread is known, policy makers can develop preventive strategies against, for instance, environmental factors that promote the disease. Once we have reliable predictions for future cases, policy makers can make informed decisions on, for example, vaccine purchases, public awareness campaigns and training programs for health care workers.

Machine learning algorithms can contribute to the control of infectious diseases

by addressing aforementioned two aims. In the literature, standard machine learning algorithms such as random forests [Breiman, 2001] and boosted regression trees [Friedman, 2001, Friedman, 2002] were frequently used in ecological and epidemiological applications [Cappelle et al., 2010, Bhatt et al., 2013, Kane et al., 2014, Ducheyne et al., 2015, Messina et al., 2016]. These algorithms have been picked by the applied researchers mainly because they have a relatively simple interface for nonspecialists. However, they might fail to capture highly complex dependencies in disease modeling scenarios. Thus, we used GPs [Williams and Rasmussen, 2006] to be able to identify highly nonlinear dependencies and to make more reliable predictions.

We proposed a computational framework that uses GPs as the basic building block to perform spatiotemporal prediction of infectious diseases. We first noted that the kernel matrices have a special structure owing to their dependencies on both spatial and temporal covariates and, then, exploited this special structure to obtain a very efficient inference algorithm. We tested our proposed framework on Turkey's country-wide surveillance data set of a vector-borne infectious disease Crimean–Congo hemorrhagic fever, which is a widespread endemic infectious disease seen in Africa, the Balkans, the Middle East, and Asia with a case fatality rate of 3–30% [Ergönül and Whitehouse, 2007].

We present the overview of our proposed computational framework with three possible prediction scenarios in Figure 2.1. We assume that the reported case counts of location and time period pairs have been recorded with additional information about their spatial and temporal properties. We first extract spatial and temporal features for each location and time period, respectively, from these properties. We then calculate two similarity matrices among locations and time periods, respectively, using the extracted features. These two similarity matrices are combined to obtain a larger similarity matrix between location and time period pairs. Using the combined similarity matrix and reported cases counts, we train a Gaussian process regression model to be able to make predictions under three different scenarios: (i) temporal prediction (i.e., predicting case counts for future time periods, leading to predicting

disease prevalence for each location in the future), (ii) spatial prediction (i.e., predicting case counts for unseen locations, leading to predicting disease spread within the same time frame in other locations), which can be used to complete missing case counts for the locations that we could not obtain historical data, and (iii) spatiotemporal prediction (i.e., predicting case counts for unseen location and future time period pairs, leading to predicting disease spread to new locations in the future), which is especially important to be able to prepare against emerging infectious diseases since there will be no historical data for the locations that experience the disease for the first time. In this study, we proposed a computational framework to perform spa-

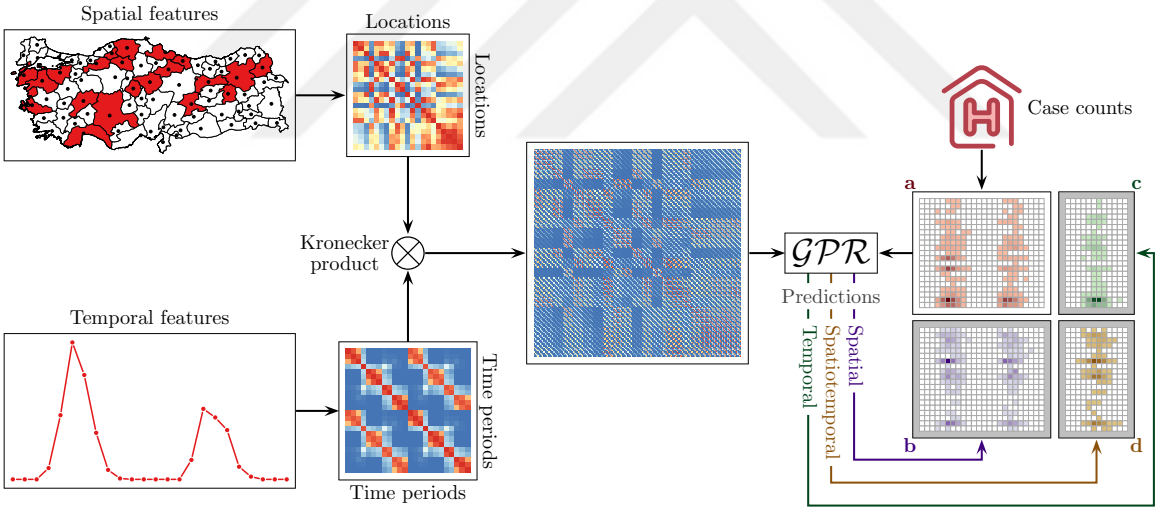

Figure 2.1: **Overview of our proposed computational framework to perform spatiotemporal prediction of infectious diseases.** (**a**) Reported case counts are given for location and time period pairs. The proposed framework can be used for three different prediction scenarios: (**b**) spatial prediction, (**c**) temporal prediction, and (**d**) spatiotemporal prediction.

tiotemporal prediction of infectious diseases. To test this framework, we addressed an important public health problem in Turkey, namely, Crimean–Congo hemorrhagic fever (CCHF), which is a vector-borne infectious disease transmitted by infected tick bites and exposure to blood or bodily fluids of the infected cases.

2.1 Materials

We used an unpublished surveillance data set of 9,636 CCHF infection cases reported in Turkey between years 2004 and 2015, which was collected by the Ministry of Health of Turkey. The reported cases were mainly because of infected tick bites, and they were diagnosed with clinical symptoms such as fever, myalgia, and bleeding from various sites. These infected cases were also confirmed with blood tests.

The Ministry of Health of Turkey provided us with spatial information (province, district, and town names) and temporal information (year and month) for each case, which made this data set suitable for studying spatiotemporal characteristics of CCHF. The data set does not include clinical covariates of infected cases, which forces our study to investigate only spatial and temporal covariates.

2.1.1 Spatial Covariates

We used the infected case counts of provinces to capture the spatial spread of CCHF since finer resolutions such as district or town level gives us very sparse case counts. Figure 2.2 shows the total numbers of infected cases reported in 81 provinces of Turkey between years 2004 and 2015, whereas annual numbers of infected cases can be seen in Appendix, Figure 7.1. CCHF cases had mainly been observed in northern and northeastern regions of Turkey (e.g., 2,046 of 9,636 infected cases were reported in a single northern province), and other regions had strikingly fewer infected cases (e.g., southern provinces had one to three infected cases per year). This confirmed that CCHF has a strong spatial dependency, which was reported by several earlier studies [Ergönül, 2006, Estrada-Peña et al., 2007b, Ergönül, 2012], owing to mainly spatial differences in wild-life and livestock animal populations carrying ticks. We extracted latitude and longitude coordinates of each province center, leading to two spatial covariates.

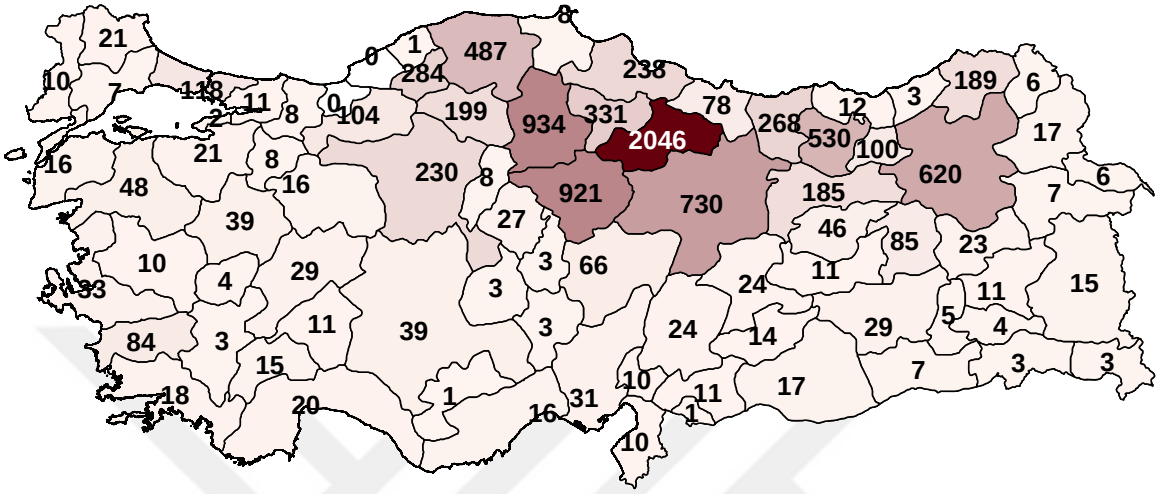

Figure 2.2: **The total numbers of infected cases reported in 81 provinces of Turkey between years 2004 and 2015.** Note that the northern and northeastern regions had strikingly high numbers of infected cases. The numbers were shown on the province centers. This map was generated using the Turkish administrative map downloaded from `https://www.gadm.org` and the R package `maps` version 3.3.0 at `https://cran.r-project.org/web/packages/maps`.

2.1.2 Temporal Covariates

We used the monthly infected case counts since we did not have data for finer resolutions and ticks become dormant (i.e., inactive) during cold weather, which makes periods longer than month unable to capture the temporal dynamics of CCHF. Figure 2.3 shows the numbers of country-wide infected cases for each month between years 2004 and 2015. 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). This confirmed that CCHF has a strong temporal dependency, which was again reported by several earlier studies [Ergönül et al., 2005, Randolph and Ergönül, 2008, Ince et al., 2014], owing to mainly life or sleep cycles of ticks. We encoded each time period by three temporal covariates: the year,

month, and seasonal group (i.e., hot, warm, or cold) it belongs to.

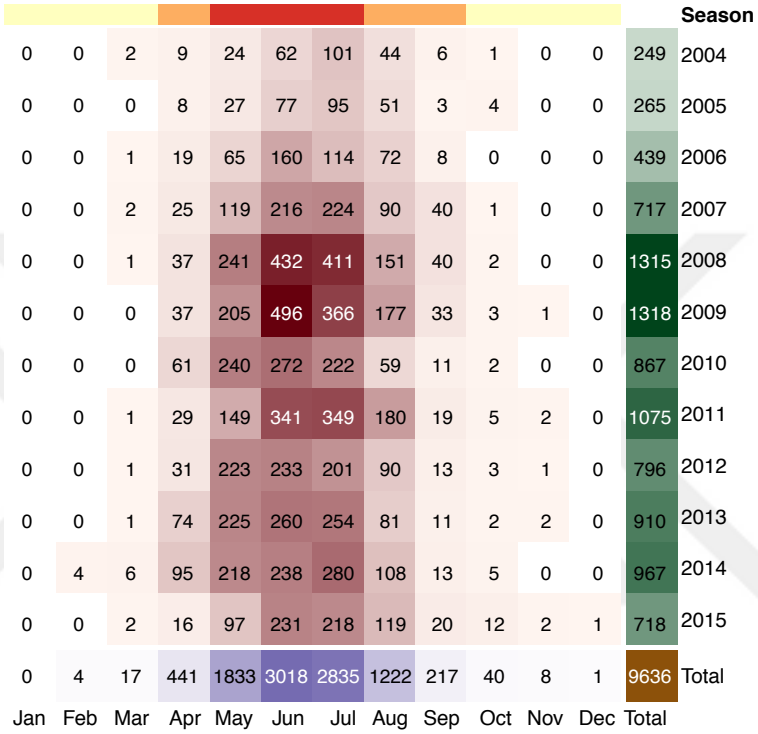

Figure 2.3: **The numbers of country-wide infected cases for each month between years 2004 and 2015.** The total numbers of infected cases for each month and each year were also reported as column and row sums, respectively. The columns were annotated by their seasonal group information at the top (yellow: cold; orange: warm; red: hot). Note that there is an annual periodicity of cases and a striking seasonal variation over infected cases.

2.2 Methods

Infectious disease spread is usually driven by both location and time, which means nearby locations and time periods have similar characteristics. The disease spreads to adjacent province much more easily than distant provinces due to spatial dependency. Case counts in consecutive time periods or in time periods within the same season are usually heavily correlated due to temporal dependency.

We suggest using GPR, which is suitable to capture highly complex dependencies between input and output variables thanks to its nonlinear nature brought by kernel functions. We propose a computational strategy based on GPR that enables us to perform predictions under spatial (i.e., predicting case counts for unseen locations), temporal (i.e., predicting case counts for future time periods) and spatiotemporal scenarios (i.e., predicting counts for unseen location and future time period pairs) for infectious diseases.

We first give a brief description of GPR. We then show how GPR can be modified for infectious disease modeling by introducing a structured kernel function based on two separate kernel functions over spatial and temporal covariates, respectively, and how this modified GPR formulation can be implemented very efficiently. We describe three different prediction scenarios encountered in spatiotemporal modeling of infectious diseases. We lastly discuss two baseline algorithms from the literature that will be used to benchmark against.

2.2.1 Structured Gaussian Process Regression

For large data sets, GPs might become computationally intensive. GPs have intensive computational and memory requirements. GP inference requires evaluating $(\mathbf{K} + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}$ for Equations (1.8) and (1.9). 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. Several decomposition algorithms have been previously proposed to make the inference faster such as Nyström approximation [Williams and Rasmussen, 2006], approximation using Hadamard and diagonal matrices [Le et al., 2013], or Kronecker methods [Bonilla et al., 2007, Finley et al., 2009a, Stegle et al., 2011, Riihimäki and Vehtari, 2014, Wilson et al., 2014, Gilboa et al., 2015, Airola and Pahikkala, 2018].

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.

In spatiotemporal modeling, we can represent each data instance $\boldsymbol{x}_i$ as a pair of location and time period vectors $(\boldsymbol{s}_l, \boldsymbol{t}_p)$, where l indexes locations, p indexes time periods, L is the number of locations, and P is the number of time periods. We can also form a response matrix $\mathbf{Y}$ of size $L \times P$ to store y_i values of these pairs.

In this case, the kernel function between data instances can be written as the multiplication of two separate kernel functions:

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

where $k_s(\cdot, \cdot)$ gives the similarity between geographical locations using spatial features, and $k_t(\cdot, \cdot)$ calculates the similarity between time periods using temporal features.

The kernel matrix calculated on the training instances can be written as the Kronecker product (see Appendix, Equation (6.1) for the definition) of two smaller kernel matrices calculated on the geographical locations and the time periods, respectively.

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

where $\mathbf{K}$, $\mathbf{K}_s$, and $\mathbf{K}_t$ are of sizes $LP \times LP$, $L \times L$, and $P \times P$, respectively. Similarly, the vector that stores kernel function outputs between the test instance and the training instances can be written as

$$\boldsymbol{k}_\star = \boldsymbol{k}_{s,\star} \otimes \boldsymbol{k}_{t,\star}.$$

Kronecker decomposition was first used within GP to model data, where inputs lie on a Cartesian grid [Saatçi, 2012]. We can replace this more complex kernel formulation into standard GP Equations (1.8) and (1.9), and obtain SGPs to exploit spatiotemporal structures.

$$\begin{aligned} p(y_\star|\boldsymbol{x}_\star, \mathbf{X}, \mathbf{Y}, \sigma_y^2) &\sim \mathcal{N}(\mu_\star, \sigma_\star^2), \\ \mu_\star &= (k_{s,\star} \otimes k_{t,\star})^\top (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1} \operatorname{vec}(\mathbf{Y}), \qquad (2.1) \\ \sigma_\star^2 &= k_s(s_\star, s_\star)k_t(t_\star, t_\star) \\ &\quad -(k_{s,\star} \otimes k_{t,\star})^\top (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}(k_{s,\star} \otimes k_{t,\star}), \qquad (2.2) \end{aligned}$$

where vec$(\cdot)$ converts the input matrix into a column vector.

Fortunately, these matrix computations can be performed efficiently. To benefit from the special structure of our kernel matrices, we will use the following properties of the Kronecker product as described in [Saatçi, 2012]:

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \tag{2.3}$$

$$(\mathbf{AB})\,\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{BXA}^{\top}), \tag{2.4}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \tag{2.5}$$

Equation (2.3) 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:

$$(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1} = (\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^{\top}. \tag{2.6}$$

We can rewrite mean and variance of SGPs using Equation (2.6). After this change, mean and variance calculations in Equations (2.1) and (2.2) can be performed very efficiently using Equations (2.4) and (2.5) 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 (2.6), which

has $\mathcal{O}(LP)$ complexity. These steps make the overall complexity of our algorithm $\mathcal{O}(L^3 + P^3)$.

Infectious disease modeling using structured GPR. In this study, we use structured GPR formulation to predict case counts under three different scenarios (Figure 2.4): (i) predicting case counts for a future time period $t_\star$, (ii) predicting case counts for an unseen location $s_\star$, and (iii) predicting case counts for an unseen location and future time period pair $(s_\star, t_\star)$. In all scenarios, we assume that we are given case counts within a list of locations for a number of time periods. **Predicting**

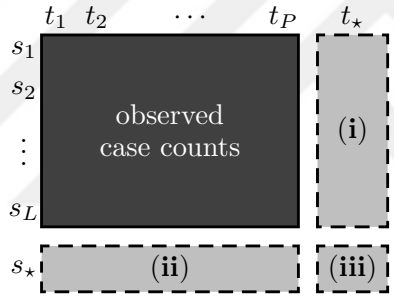

Figure 2.4: **Three prediction scenarios.** (i) temporal scenario to predict case counts of future time points on the training locations, (ii) spatial scenario to predict case counts of unseen locations at the training time points, and (iii) spatiotemporal scenario to predict case counts of unseen locations at future time points.

case counts for a future time period. In the first scenario, we are interested in finding case counts in the observed locations for a future time period. This amounts to making predictions for $(\boldsymbol{s}_l, \boldsymbol{t}_\star)$ pairs, where $\boldsymbol{s}_l$ is one of the locations in our training set.

Predicting case counts for an unseen location. In the second scenario, we are interested in finding case counts in an unseen location for the observed time periods. This amounts to making predictions for $(\boldsymbol{s}_\star, \boldsymbol{t}_p)$ pairs, where $\boldsymbol{t}_p$ is one of the time periods in our training set.

Predicting case counts for an unseen location and future time period pair. In the third scenario, we are interested in finding case counts in an unseen location

for a future time period. This amounts to making predictions for $(\boldsymbol{s}_\star, \boldsymbol{t}_\star)$ pairs.

Baseline algorithms. Several off-the-shelf machine learning algorithms can be used to perform spatiotemporal prediction of infectious diseases. In this study, we compared our method against two particular baseline algorithms, namely, random forests regression (RFR) and boosted regression trees (BRT). We have two main reasons for these particular choices: (i) Both RFR and BRT are frequently used and considered as the standard machine learning algorithms to capture temporal, spatial, and spatiotemporal dependencies in ecological and epidemiological applications [Cappelle et al., 2010, Bhatt et al., 2013, Kane et al., 2014, Ducheyne et al., 2015, Messina et al., 2016]. (ii) Both RFR and BRT are nonlinear algorithms as our structured GPR formulation.

Random forests regression. RFR algorithm combines several regression trees trained on different portions of the input covariates [Breiman, 2001]. As a result, the obtained regression trees give diverse decision rules, and combining several trees produces more robust results.

Boosted regression trees. BRT algorithm is based on the idea of combining weak learners to obtain better learners (i.e., boosting) and uses decision trees trained on different subsamples of training instances as weak learners [Friedman, 2001, Friedman, 2002].

2.2.2 Experimental Settings and Performance Metrics

We created three scenarios to perform experiments for temporal, spatial, and spatiotemporal prediction.

For temporal prediction, we took the first 10 years and the remaining two years as training and test sets, respectively. We first trained the three algorithms using case counts of 81 provinces over 10 years (120 months) as the observed response matrix, leading to a training set of 9,720 instances (81 provinces $\times$ 120 months). We then tested the trained models by predicting observed case counts of 81 provinces for the remaining two years (24 months), leading to a test set of 1,944 instances (81 provinces

× 24 months).

For spatial prediction, we divided 81 provinces into two groups by first ordering their total case counts and, then, taking odd- and even-numbered provinces as training and test sets, respectively (see Appendix, Figure 7.2). We first trained the three algorithms using case counts of 41 training provinces over 12 years (144 months) as the observed response matrix, leading to a training set of 5,904 instances (41 provinces × 144 months). We then tested the trained models by predicting observed case counts of 40 test provinces for the same time periods, leading to a test set of 5,760 instances (40 provinces × 144 months).

For spatiotemporal prediction, we took the intersection of training sets (respectively, test sets) of the first two scenarios as the training set (respectively, test set). We first trained the three algorithms using case counts of 41 training provinces over 10 years (120 months) as the observed response matrix, leading to a training set of 4,920 instances (41 provinces × 120 months). We then tested the trained models by predicting observed case counts of 40 test provinces for the last two years (24 months), leading to a test set of 960 instances (40 provinces × 24 months).

The observed case counts were mapped to logarithmic scale after adding one since they are count data and contain zero values. These mapped values were used as the response matrix for all three algorithms. After training the algorithms, their predictions were mapped back to the original scale by exponentiating first and, then, subtracting one.

For RFR algorithm, we used the `randomForest` R package version 4.6-12 [Liaw et al., 2002]. We set the formula parameter `formula` to "cases ~ year + month + season + latitude + longitude" to describe the model and set the number of trees to grow parameter `ntree` to 100,000, and other parameters were held at their default values.

For BRT algorithm, we used the `gbm` R package version 2.1.1 [Ridgeway et al., 2006]. We set the formula parameter `formula` to "cases ~ year + month + season + latitude + longitude" to describe the model, set the maximum number of iterations

(i.e., the maximum number of trees) parameter `n.trees` to 100,000, set the number of cross-validation folds parameter `cv.folds` to 5 and set the maximum depth of variable interactions parameter `interaction.depth` to 2, and other parameters were held at their default values.

We implemented our structured GPR algorithm in R and used the Gaussian kernel to define similarity functions on spatial and temporal covariates. The Gaussian kernel function $k_{\mathcal{G}}(\cdot, \cdot)$ between two data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be defined as

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

where $\|\cdot\|_2$ denotes the ℓ_2 norm, and s is the kernel width parameter. For spatial covariates of two data instances (i.e., latitude and longitude coordinates of two province centers), we defined the spatial kernel as $k_s(\boldsymbol{s}_l, \boldsymbol{s}_m) = k_{\mathcal{G}}(\boldsymbol{s}_l, \boldsymbol{s}_m)$ and picked the kernel width parameter as the mean of pairwise Euclidean distances between training instances. For temporal covariates of two time periods (i.e., years, months, and seasonal groups of two time periods), we defined the temporal kernel as the multiplication of three kernels, i.e., $k_t(\boldsymbol{t}_p, \boldsymbol{t}_q) = k_{\text{year}}(\boldsymbol{t}_p, \boldsymbol{t}_q) k_{\text{month}}(\boldsymbol{t}_p, \boldsymbol{t}_q) k_{\text{season}}(\boldsymbol{t}_p, \boldsymbol{t}_q)$, to capture the interaction effects between them, where we had three separate Gaussian kernels on year, month, and seasonal group covariates. The kernel width parameters were chosen as the means of pairwise Euclidean distances between training instances for all three kernels. We picked the standard deviation parameter of measurement noise values σ_y as the standard deviation of log-scaled observed case counts of training instances.

We used the Pearson's correlation coefficient (PCC) and normalized root mean squared error (NRMSE) to compare prediction performances of the three algorithms. PCC can be calculated as

$$\text{PCC} = \frac{(\boldsymbol{y} - \mathbf{1}y_{\cdot})^\top (\widehat{\boldsymbol{y}} - \mathbf{1}\widehat{y}_{\cdot})}{\sqrt{(\boldsymbol{y} - \mathbf{1}y_{\cdot})^\top (\boldsymbol{y} - \mathbf{1}y_{\cdot})} \sqrt{(\widehat{\boldsymbol{y}} - \mathbf{1}\widehat{y}_{\cdot})^\top (\widehat{\boldsymbol{y}} - \mathbf{1}\widehat{y}_{\cdot})}}$$

where $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$ denote the vectors of observed and predicted case counts, respectively, and $y_{\cdot}$ and $\widehat{y}_{\cdot}$ denote the averages of $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$, respectively. Larger PCC values correspond to better performance in capturing the trend in case counts. NRMSE can

be calculated as

$$\text{NRMSE} = \sqrt{\frac{(\boldsymbol{y} - \widehat{\boldsymbol{y}})^\top (\boldsymbol{y} - \widehat{\boldsymbol{y}})}{(\boldsymbol{y} - \mathbf{1}y_\cdot)^\top (\boldsymbol{y} - \mathbf{1}y_\cdot)}}.$$

Smaller NRMSE values correspond to better performance in capturing the scale of case counts.

2.3 Results

2.3.1 Performance Comparison

Table 2.1 reports PCC values of RFR, BRT, and GPR algorithms on our CCHF data set for three prediction scenarios. We see that GPR algorithm obtained the best PCC values by improving the results of temporal, spatial, and spatiotemporal prediction scenarios by 1.05%, 26.31%, and 16.45%, respectively. Note that RFR and BRT algorithms failed to capture the spatial spread of CCHF when predicting case counts for unseen provinces (i.e., in spatial and spatiotemporal scenarios), whereas GPR algorithm was able to capture this spread by obtaining more than 70% PCC for these two scenarios. All algorithms achieved PCC values around 75% and 85% for temporal scenario since capturing temporal dynamics is easier owing to annual periodicity of CCHF cases.

Table 2.1: **Pearson's correlation coefficients of three algorithms on CCHF data set for three prediction scenarios together with ranks in parentheses**

	Temporal	**Spatial**	**Spatiotemporal**
RFR	0.748 (3)	0.486 (2)	0.543 (2)
BRT	0.846 (2)	0.437 (3)	0.493 (3)
GPR	0.857 (1)	0.749 (1)	0.707 (1)

Table 2.2 shows NRMSE values of RFR, BRT, and GPR algorithms on our CCHF data set for temporal, spatial, and spatiotemporal prediction scenarios. We see that

GPR algorithm again obtained the best NRMSE values by improving the results of temporal, spatial, and spatiotemporal prediction scenarios by 21.39%, 20.38% and 15.65%, respectively. Even though BRT algorithm obtained a PCC value comparable to that of GPR algorithm for temporal scenario, GPR algorithm obtained considerably better NRMSE values than both RFR and BRT algorithms. This shows that GPR algorithm is better than the other two algorithms in terms of capturing the range of CCHF cases in the test sets as discussed below.

Table 2.2: **Normalized root mean squared errors of three algorithms on CCHF data set for three prediction scenarios together with ranks in parentheses**

	Temporal	**Spatial**	**Spatiotemporal**
RFR	0.875 (3)	0.927 (3)	0.894 (3)
BRT	0.746 (2)	0.900 (2)	0.876 (2)
GPR	0.532 (1)	0.697 (1)	0.720 (1)

Figure 2.5 shows the total observed and predicted case counts by RFR, BRT and GPR algorithms for years 2014 and 2015 over the five provinces with the highest case counts among 40 common test provinces of all scenarios. We see that all three algorithms captured the annual periodicity of CCHF cases, whereas GPR algorithm performed the best in terms of predicting the observed case counts. RFR algorithm was not able to predict the observed case counts owing to its lack of high order interactions between covariates, whereas BRT algorithm performed better owing to its second order interactions. The same results were also valid if we took the first 10, 15, and 20 provinces from 40 common test provinces (see Appendix, Figure 7.3, 7.4, and 7.5).

Appendix Figure 7.6 gives a detailed comparison between observed and predicted case counts of RFR, BRT, and GPR algorithms for the same five provinces reported in Figure 2.5. We see that GPR algorithm produced predictions mostly in agreement

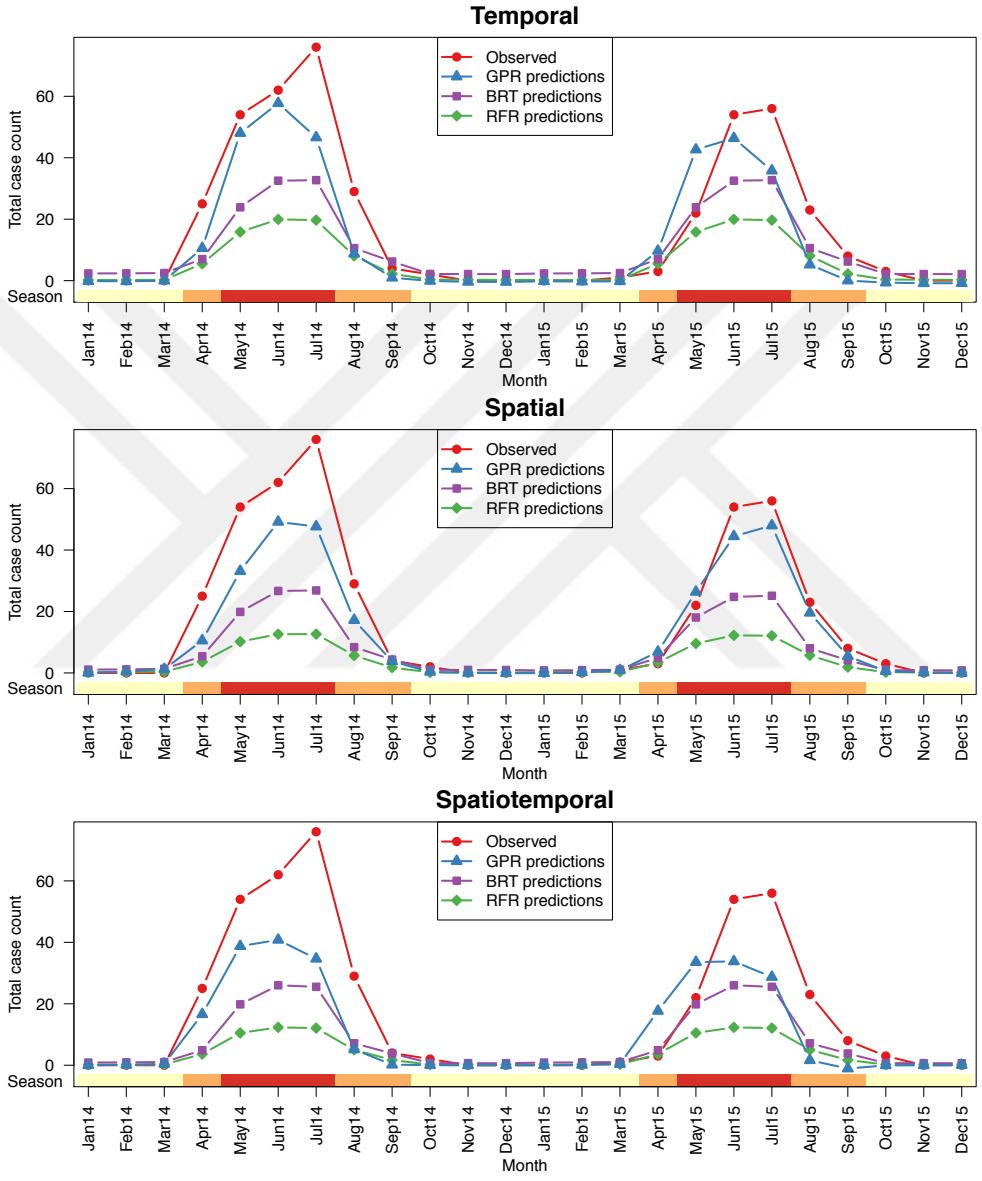

Figure 2.5: **The total observed and predicted case counts for years 2014 and 2015 over the five provinces with the highest case counts (i.e., endemic region) among 40 common test provinces of all scenarios.** The time periods were annotated by their seasonal group information at the bottom (yellow: cold; orange: warm; red: hot). Note that all three algorithms were able to capture the annual periodicity of CCHF cases in all scenarios, whereas the predicted case counts of GPR algorithm were closer to the observed CCHF cases.

with the range of observed CCHF case counts, whereas RFR and BRT algorithms underestimated CCHF case counts in most of the time periods. BRT algorithm obtained NRMSE value comparable to that of GPR algorithm for temporal scenario, whereas GPR algorithm reduced NRMSE values by 0.277 and 0.170 for spatial and spatiotemporal scenarios, respectively.

The results of the computational experiments reported in this study can be analyzed from different perspectives. We analyzed the results with respect to prediction scenarios, machine learning algorithms, computational complexity, dependency on training set size, and dependency on sampling over provinces.

2.3.2 Prediction Scenarios

We performed computational experiments under three different scenarios. As we can see from Table 2.1, Table 2.2, Figure 2.5, and Appendix Figure 7.6, making temporal predictions (i.e., predicting future time periods by looking at the historical data) is strikingly easier than making spatial and spatiotemporal predictions (i.e., generalizing to unseen locations). Most infectious disease outbreaks occur in cycles (i.e., ascending, plateau, and descending phases), and this structure makes temporal prediction easier. The disease we addressed is a vector-borne infectious disease mainly transmitted by infected tick bites, leading to a strong temporal dependency owing to the sleep cycles of ticks.

2.3.3 Machine Learning Algorithms

We used three machine learning algorithms for predicting case counts. As we discussed before, GPR algorithm was able to capture the range of CCHF case counts better than RFR and BRT algorithms. We think that this was mainly due to the capability of GPR algorithm to model highly complex dependencies between input and output covariates thanks to nonlinear kernel functions such as the Gaussian kernel we used. We also noted from Figure 2.5 and Appendix Figure 7.6 that the main improvement of GPR algorithm over the others was the ability to better capture the range of

case counts in the time periods with nonzero observed case counts. In the literature, RFR and BRT algorithms were frequently used as classification algorithms to predict whether there will be cases. In terms of classification performance, we would not expect major differences between three algorithms.

Computational complexity

Instead of using a naive version of GPR algorithm, we implemented an efficient variant that exploits the special structure of the kernel matrix to make inference very fast. We decomposed the kernel matrix into a Kronecker product of two smaller kernel matrices calculated on spatial and temporal covariates, respectively. By doing so, we were able to perform inference for our structured GPR formulation in the order of milliseconds, whereas RFR and BRT algorithms took several minutes to complete using drastically higher physical memory.

2.3.4 Dependency on Training Set Size

To show the dependency of GPR on training set size, we performed an additional set of experiments by changing the number of years used for training. We used CCHF case counts of the last two, four, six, eight, and ten years between 2004 and 2013, respectively. Table 2.3 shows PCC and NRMSE values of GPR algorithm for this new set of experiments. We can see that there was an increasing trend in predictive performance as we increased the training set size.

2.3.5 Dependency on Sampling over Spatial Dimension

Up to this point, we performed our experiments on a fixed training and test set split (Appendix Figure 7.2), which was designed to make training and test sets as similar as possible, to better illustrate the differences between machine learning algorithms. We also compared the predictive performances of RFR, BRT, and GPR on 100 different training and set set splits constructed by random sampling on 81 provinces. Figure 2.6 shows PCC and NRMSE values of the algorithms for spatial and spatiotemporal

Table 2.3: **Pearson's correlation coefficients and normalized root mean squared errors of GPR algorithm on CCHF data set with changing training set size (i.e., 2, 4, 6, 8, and 10 years)**

	Temporal		**Spatiotemporal**	
	PCC	NRMSE	PCC	NRMSE
2012–13	0.633	1.015	0.558	1.039
2010–13	0.749	0.830	0.636	0.960
2008–13	0.831	0.582	0.725	0.760
2006–13	0.791	0.637	0.745	0.671
2004–13	0.857	0.532	0.707	0.720

modeling scenarios. We see that our algorithm GPR was statistically significantly better (i.e., $p < 0.001$) than other two algorithms for both scenarios in terms of PCC values. In spatial prediction scenario, GPR achieved statistically significantly better NRMSE values than RFR (i.e., $p = 0.023$), but it obtained NRMSE values comparable to BRT (i.e., $p = 0.052$). In spatiotemporal prediction scenario, NRMSE values of GPR were statistically significantly better than those of BRT (i.e., $p < 0.001$), whereas NRMSE values were comparable between GPR and RFR (i.e., $p = 0.932$).

2.4 Discussion

Infectious diseases cause important health problems worldwide and create difficult challenges for public health policy makers. To be able to make correct and effective decisions, it is quite important to understand the characteristics of each infectious disease, which includes environmental factors such as climate and animal population in addition to molecular evolution of disease sources such as bacteria and viruses. In this study, we addressed to capture the effect of environmental factors on infectious diseases by modeling their spatial and temporal dependencies on these factors.

For this purpose, several computational methods have been proposed in the litera-

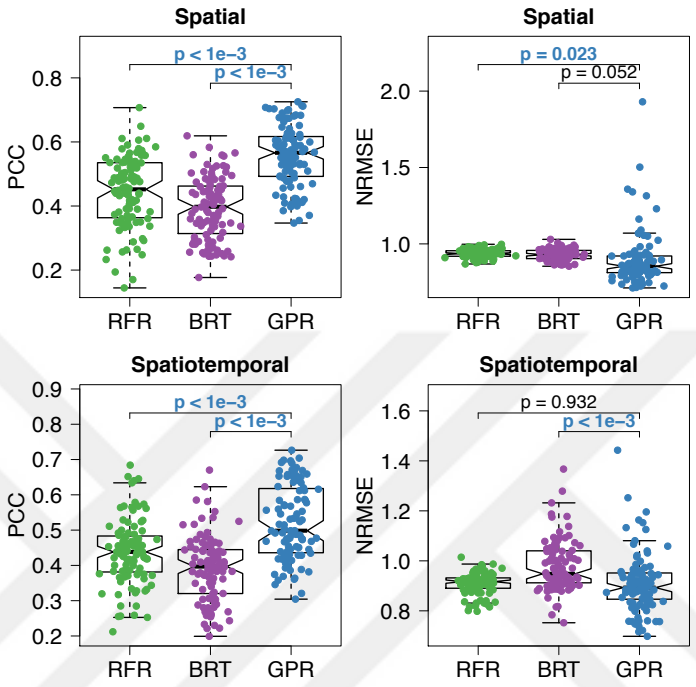

Figure 2.6: **Pearson's correlation coefficients and normalized root mean squared errors of three algorithms on CCHF data set for 100 different training and test set splits of 81 provinces for spatial and spatiotemporal modeling scenarios.** GPR was compared against RFR and BRT using a two-sided paired t-test to check whether the predictive performances are 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.

ture, whereas we focused only on machine learning algorithms applied to this problem. Easy-to-use machine learning algorithms such as random forests and boosted regression trees were frequently used in infectious disease modeling studies. However, GPs might capture highly complex dependencies better than these tree-based algorithms. Thus, we formulated a computational framework based on Gaussian processes that can be used to perform spatial, temporal, or spatiotemporal prediction of infectious diseases.

We integrated spatial features (such as geographical coordinates) and temporal

features (such as seasonal conditions) for location and time period pairs that were used as data instances in our GP formulation. However, a naive implementation of GPs would become computationally infeasible owing to very high numbers of pairs being modeled. We exploited the special structure (i.e., Kronecker) of similarity matrices in our formulation to obtain a very efficient implementation, which enabled us to train models for around 10,000 data instances in the order of milliseconds.

We applied our framework to the problem of predicting the case counts of a vector-borne infectious disease Crimean–Congo hemorrhagic fever using the data set of infected case counts between years 2004 and 2015 collected by the Ministry of Health of Turkey. We performed predictions under three different scenarios (Figure 2.1), which correspond to making predictions for unseen provinces (i.e., spatial prediction), future time periods (i.e., temporal prediction), or unseen province and time period pairs (i.e., spatiotemporal prediction) to show the suitability of our approach to distinct problems.

Predicting future cases of infectious diseases is very important for the control and prevention of the disease. The predicted case counts can be used to develop new public health policies and intervention mechanisms. It is more useful for public health policy makers to be able to predict the possible number of infected cases for a region and a time period pair rather than predicting whether there will be cases or not. Policy makers can make use of predicted number of infected cases to purchase vaccines around the right amount, to raise public awareness in the region, to educate health care workers, etc. From that perspective, GPR algorithm did a better job than RFR and BRT algorithms by predicting CCHF case counts more accurately (i.e., lower NRMSE values).

We tested our proposed formulation on a single disease, but the same framework can be extended towards other vector-borne infectious diseases (e.g., dengue fever, malaria, Zika fever) and as well as other infectious diseases (e.g., influenza, measles, tuberculosis). We also made the source code publicly available to enable other computational and applied researchers to make such extensions easily.

In the next chapter, we improve our model by integrating multiple kernel learning method to our computational framework for better accuracy as well as more expressiveness.

Chapter 3

STRUCTURED GAUSSIAN PROCESSES WITH MULTIPLE LEARNING

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 et al., 2011].

As mentioned before, GPs might become computationally intensive for large data sets. That is why several decomposition algorithms have been previously proposed to make the inference faster such as Nyström approximation [Williams and Rasmussen, 2006], approximation using Hadamard and diagonal matrices [Le et al., 2013], or Kronecker methods [Bonilla et al., 2007, Finley et al., 2009b, Saatçi, 2012, Stegle et al., 2011, Riihimäki and Vehtari, 2014, Wilson et al., 2014, Gilboa et al., 2015]. 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, 2018], 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].

In this study, we proposed a GP approach with Kronecker decomposition for spa-

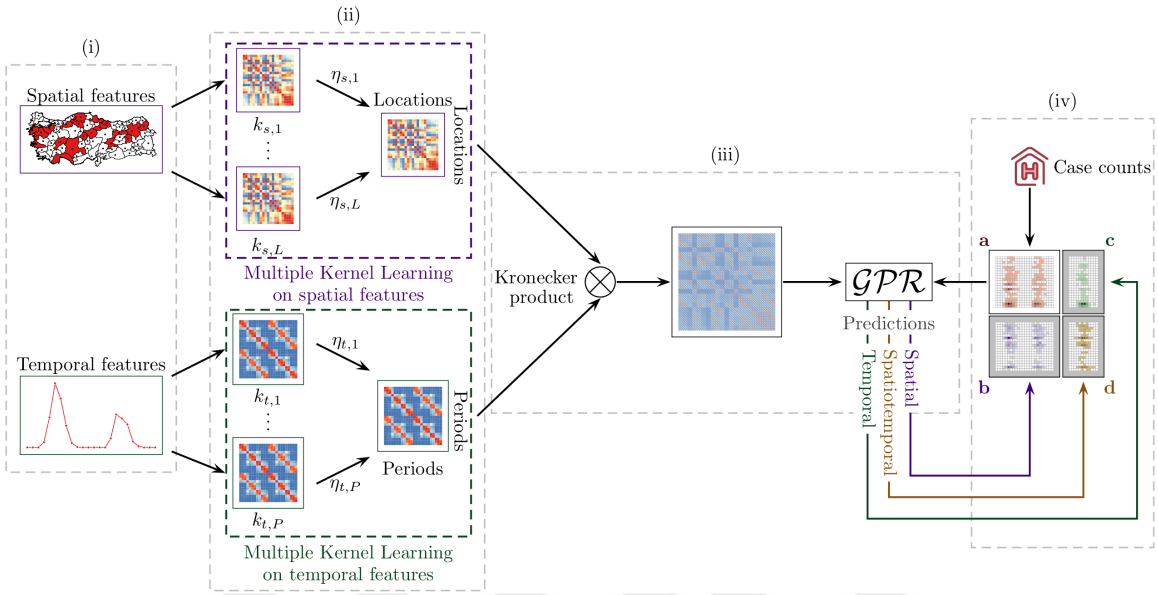

Figure 3.1: **Our computational framework for spatiotemporal inference with multiple kernel learning.** (i) Temporal and spatial feature extraction, (ii) Twin multiple kernel learning, (iii) Kronecker product based GP regression ($\mathcal{GPR}$), and (iv) Prediction scenarios: (a) Given response values for observed location and time pairs to make inference in three different scenarios: (b) spatial prediction, (c) temporal prediction, and (d) spatiotemporal prediction.

tiotemporal 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 3.1 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.

In the following Section 3.1, we describe a multiple kernel learning (MKL) approach for inference and hyper-parameter learning in SGPs. Finally, in Section 3.2,

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.

3.1 Structured Gaussian Processes with Twin Multiple Kernel Learning

In the previous chapter, 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 [Williams and Rasmussen, 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.

3.1.1 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.

3.1.2 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 [Williams and Rasmussen, 2006]:

$$\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top \mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi, \tag{3.1}$$

$$\frac{\partial \log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_m} = \frac{1}{2}\boldsymbol{y}^\top \mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_m}\mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\operatorname{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_m}\right), \tag{3.2}$$

where $\boldsymbol{\theta}$ is the vector of the parameters of the covariance function. 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:

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \eta_{s,m}} = \frac{\partial \mathbf{K}_s}{\partial \eta_{s,m}} \otimes \mathbf{K}_t, \tag{3.3}$$

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \eta_{t,n}} = \mathbf{K}_s \otimes \frac{\partial \mathbf{K}_t}{\partial \eta_{t,n}}, \tag{3.4}$$

$$\frac{\partial(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})}{\partial \sigma_y} = 2\sigma_y\mathbf{I}, \tag{3.5}$$

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 (3.3)–(3.5) and, then, plugged these resulting equations into the gradient calculation in Equation (3.2). The first term of the gradient can be computed efficiently using partial derivatives in Equations (3.3)–(3.5) and Kronecker properties in Equations (2.3)–(2.5). 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:

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

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. See Appendix, Section 6.2.3 for further details.

We estimated the parameters using a constrained optimization method in R package `alabama` [Varadhan, 2017]. We used the function `constrOptim.nl`, which uses an objective function to be optimized (i.e., likelihood function in Equation (3.1)), the gradient of the objective function evaluated at the argument (i.e., gradient in Equation (3.2)), 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 $\sum \eta_{t,n} = 1$.

3.2 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 et al., 2002, Ridgeway et al., 2006]. 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 3.1(c), (ii) spatial prediction scenario (i.e., predicting historical data for new locations using data for observed locations), see Figure 3.1(b), (iii) spatiotemporal prediction scenario (i.e., predicting future time points in new locations), see Figure 3.1(d).

In all experiments, instead of learning kernel hyper-parameters using type-II maximum likelihood [Williams and Rasmussen, 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.

3.2.1 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. In Appendix chapter, Figure 7.1 reports the yearly CCHF case counts between 2004 and 2015 for 81 provinces.

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.` $\times$ `Lon.`, and the weights of the temporal pairwise interaction kernels `Year` $\times$ `Month`, `Year` $\times$ `Season`, and `Month` $\times$ `Season`.

The spatial interaction kernel had the highest weight in all of the prediction scenarios, approximately one in spatial and spatiotemporal scenarios (see Figure 3.2). 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. Table 3.1 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 3.3. 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.

Figure 3.4 shows the comparison between observed and predicted cases of years

Figure 3.2: **Averaged kernel weights found by SGP2MKL on CCHF data set.**

Table 3.1: **PCC and NRMSE values of SGP2MKL and other 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)

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.

3.2.2 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

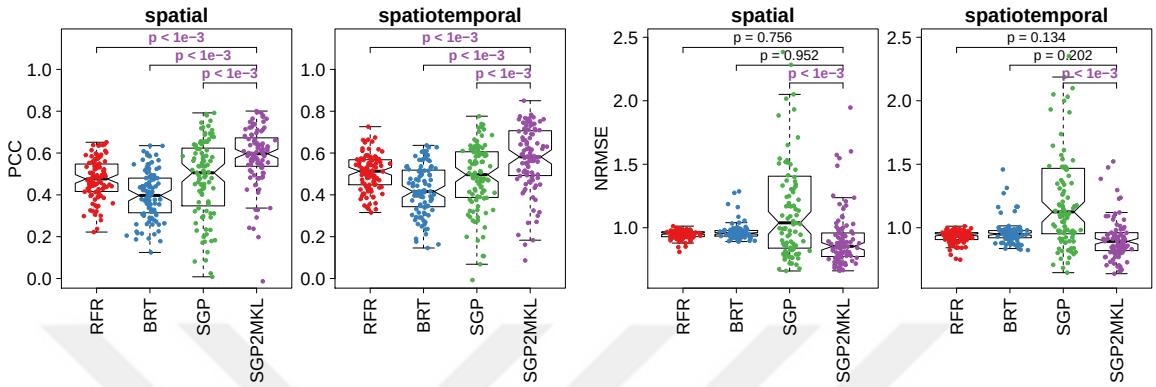

Figure 3.3: **PCC and NRMSE values of four algorithms on CCHF data set for spatial and spatiotemporal prediction scenarios.** SGP2MKL was compared against each competitor using a two-sided paired t-test to check whether the predictive performances were statistically significantly different, and P-value 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.

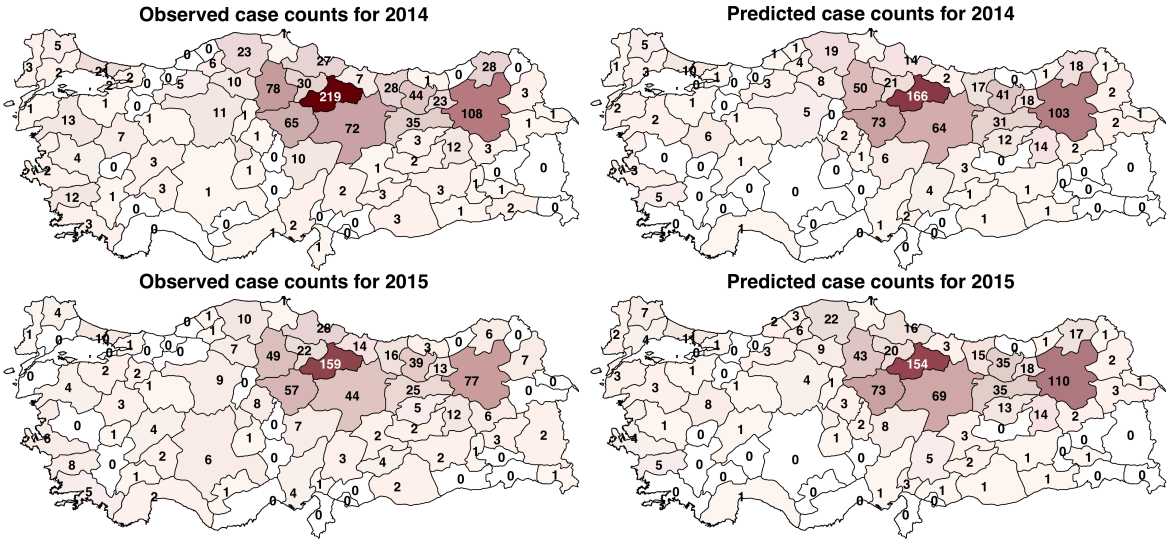

Figure 3.4: **Country-wide observed versus predicted case counts of years 2014 and 2015 for temporal scenario.** Observed and predicted case counts of 81 provinces aggregated yearly after prediction for illustration purposes.

Figure 3.5). Thus, there are 576 spatial locations and 72 time periods. The first five

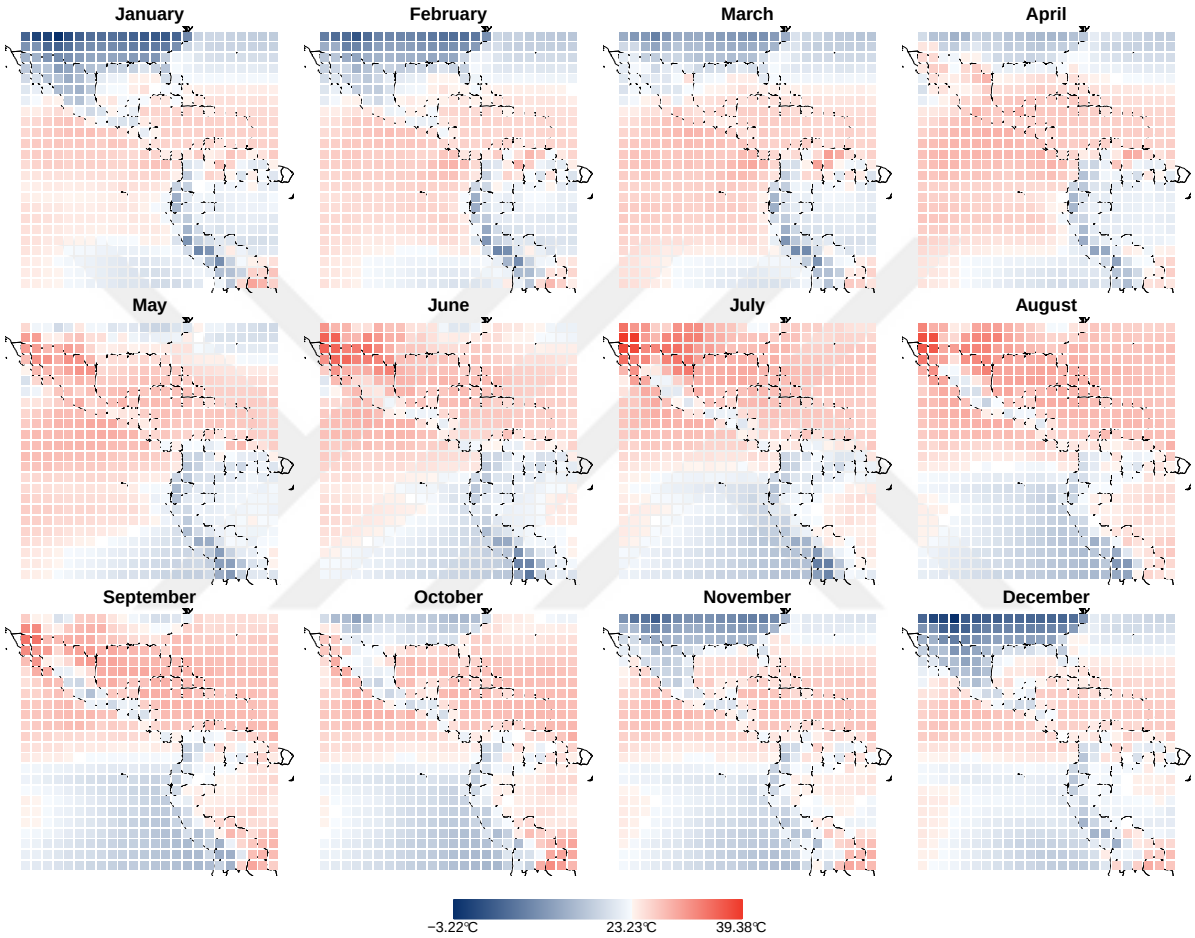

Figure 3.5: **Observed monthly averages of surface temperature on 24 by 24 grid locations between years 1995 and 2000 over the central America.** Here, we show the mean of monthly averages in each grid location over all years. We color the overall mean temperature (23.23 °C) with white, and temperatures lower (higher) than this mean with blue (red).

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 3.6. 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. Table 3.2 reports PCC and

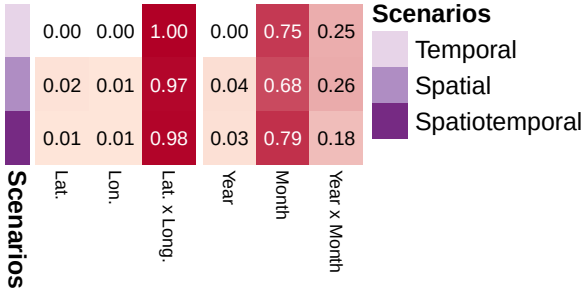

Figure 3.6: **Averaged kernel weights found by SGP2MKL on NASA's surface temperature data set.**

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 3.7 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 3.2: **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)

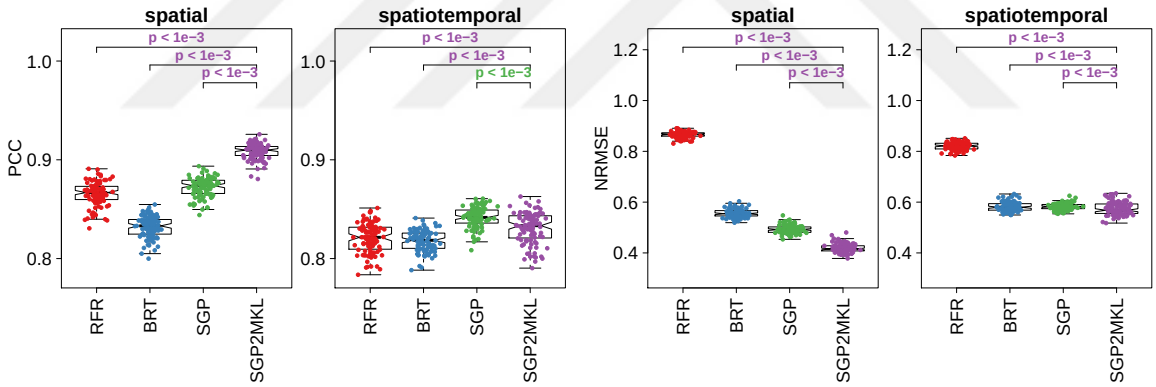

Figure 3.7: **PCC NRMSE values of four algorithms on NASA's surface temperature data set for spatial and spatiotemporal prediction scenarios.** SGP2MKL was compared against each competitor using a two-sided paired t-test to check whether the predictive performances are statistically significantly different, 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.

3.3 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 chapter, 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.

In the next chapter, we present another learning routine and incorporate different sources of data to improve the performance of the model both in accuracy and interpretability.

Chapter 4

A PROSPECTIVE TOOL TO PREDICT FUTURE CASES OF CRIMEAN–CONGO HEMORRHAGIC FEVER

In this chapter, we collected data from various data sources and made use of all possible information to better understand the underlying reasons of the spread of CCHF cases in Turkey. At the same time, we presented prospective CCHF case count prediction in collaboration with the Ministry of Health.

Turkey has the highest CCHF case counts among other countries where it remains endemic. *H. marginatum* ticks are the primer vectors, and they feed on animals at each developmental stage. Both wild and domesticated animals are important in the disease transmission cycle, serving as reservoirs for the continuation of tick reinfection.

People working or living close to livestock or to habitats of the vector ticks are particularly at risk. Human-to-human transmission is possible, typically among healthcare workers or caregivers having close contact and exposure to infectious blood or bodily fluids of those infected with CCHF virus (CCHFV). When the possibility for enzootic transmission exposure increases, the risk of CCHFV infection for humans increases as well [Kilpatrick and Randolph, 2012]. Environmental changes can influence both the survival and reproduction of *H. marginatum* ticks, then may trigger community outbreaks. For example, neglect of agricultural lands and agricultural reforms causing landscape alterations may be an important factor for the emergence of CCHF. Thus, the investigation of those environmental factors that may influence the cycle of CCHF is relevant for outbreak preparedness and response.

Some of the seasonal and climatic covariates were previously reported as important predictors of CCHFV infections [Mostafavi et al., 2013, Ansari et al., 2014, Vescio et al., 2012]. Areas with higher temperatures, precipitation, and humidity were linked

with the high CCHF occurrence in Bulgaria and Iran [Ansari et al., 2014, Vescio et al., 2012]. Suitable habitat for H. marginatum ticks was reported as fragmented agricultural lands, forested lands, and grass cover in Turkey and Bulgaria, and non-irrigated agricultural land (e.g., pasture) was found to be correlated with CCHF case counts in Turkey [Vescio et al., 2012, Estrada-Peña et al., 2007a, Estrada-Peña et al., 2007b].

The use of spatiotemporal modelling tools might help us better understand the characteristics of established outbreaks to develop different types of interventions to prevent and treat diseases. Predicting the emergence is not realistic because there are so many variables; nevertheless predicting the spatial and temporal trajectory is feasible and probably more effective [Holmes et al., 2018]. Such studies were done in the past for Ebola, Zika, H1N1 influenza, and severe acute respiratory syndrome viruses and the results of these studies helped decision makers to plan bed capacity [Washington et al., 2015], anticipate travel-related spread [Bogoch et al., 2015], and plan vaccine trials [Camacho et al., 2017].

World-wide CCHF retrospective risk maps were reported using the published cases [Messina et al., 2016], however a prospective risk analysis based on a comprehensive set of data including climatic, environmental, and husbandry parameters is still lacking. Turkey has the highest number of laboratory confirmed CCHF cases. Fourteen-year long monthly data of more than ten thousand cases could be quite insightful for understanding spatiotemporal dynamics of the disease spread. We already presented the improved performance of structured GP, against frequently-used machine learning algorithms used in ecological and epidemiological applications [Ak et al., 2018a]. By this study, we described the spatiotemporal dynamics of CCHF and extracted the important covariates for CCHFV infection using structured GP method on the surveillance data set of Turkey. We tested the generalization capability of our approach by predicting where and how many CCHF cases will be observed in each month in 2016 and 2017 prospectively.

4.1 Methods

The surveillance data consists of monthly case counts (i.e., observations) for each province. Our regression model takes the past case counts and covariate information as inputs and outputs a numeric value as the future case count.

Surveillance data:

The date (i.e., month and year) and location (i.e., province, county, and village) of the laboratory confirmed CCHF cases in Turkey between January 2004 and December 2015 were obtained from the Ministry of Health to train our predictive model. We were provided with the surveillance data between January 2016 and December 2017 after we made our predictions for these years. In our study, the case counts were aggregated in the province level, so the province centers were used as the case locations.

Agricultural, demographic, geographic, meteorological and temporal covariates:

We collected over 50 potentially related spatial and temporal covariates to be able to use as input in our model. These covariates are listed in Table 4.1. Detailed interpretations of the covariates are presented at `http://midas.ku.edu.tr/ProspectiveCCHF`.

Table 4.1: Spatial and temporal covariates

Covariate abbreviation	Covariate name	Covariate type
Spatial covariates		
Geographic coordinates		
Latit	Latitude	Static
Longi	Longitude	Static
Altit	Altitude	Static
Vegetation/agriculture		
Tparc	Total agricultural parcels	Static
Tsown	Total sown area	Static
Tgard	Total vegetables and flower gardens	Static
Tcrop	Total fruit orchards and other permanent crops	Static
Tpopl	Total poplar and willow land	Static
Tnuse	Total unused and undeveloped potentially productive land	Static

Continued on next page

Table 4.1 – continued from previous page

Covariate abbreviation	Covariate name	Covariate type
Tmead	Total permanent meadow	Static
Fallo	Fallow land	Static
Pastu	Pasture land	Static
Fores	Forest and woodland -heather and macquis are included -	Static
Nagri	Non-agricultural land - stony land, swamp, arid land etc. -	Static
Demography		
Nsett	Total number of settlements with fewer than 25,000 inhabitants	Static
Asett	Total area of settlements with fewer than 25,000 inhabitants	Static
Build	Land occupied by buildings - graveyard etc. -	Static
Nhous	Total number of households	Static
Nhagr	Number of households engaged in agricultural activity	Static
Nnhagr	Number of households not engaged in agricultural activity	Static
Temporal covariates		
Date		
Year	Year	Dynamic, annual
Month	Month	Dynamic, monthly
Seaso	Season	Dynamic, monthly
Meteorology		
Cldcv	Cloud cover	Dynamic, monthly
Ditmp	Diurnal temperature range	Dynamic, monthly
Gfrsf	Ground frost frequency	Dynamic, monthly
Tmpmx	Maximum temperature	Dynamic, monthly
Tmpme	Mean temperature	Dynamic, monthly
Tmpmn	Minimum temperature	Dynamic, monthly
Pevap	Potential evapotranspiration	Dynamic, monthly
Preci	Precipitation	Dynamic, monthly
Wetdy	Rainy days	Dynamic, monthly
Vapou	Vapor pressure	Dynamic, monthly
Vegetation / agriculture		
Cagri	Total utilized agricultural land	Dynamic, annual
Carcr	Total arable land and land under permanent crops	Dynamic, annual
Carab	Total arable land	Dynamic, annual
Csown	Sown area	Dynamic, annual
Cfallo	Fallow land	Dynamic, annual
Cpast	Land under permanent meadows and pasture	Dynamic, annual
Cgard	Area of vegetable gardens	Dynamic, annual
Ccrop	Total land under permanent crops	Dynamic, annual
Cfrui	Area of other fruit, beverage and spices crops	Dynamic, annual
Cvine	Area of vineyard	Dynamic, annual
Coliv	Area of olive trees	Dynamic, annual
Cfore	Forest area	Dynamic, annual

Continued on next page

Table 4.1 – continued from previous page

Covariate abbreviation	Covariate name	Covariate type
Livestock		
Ncatt	Number of cattle livestock	Dynamic, annual
Nshee	Number of sheep livestock	Dynamic, annual
Ngoat	Number of goat livestock	Dynamic, annual
Nbuff	Number of buffalo livestock	Dynamic, annual
Nscat	Number of slaughtered cattle	Dynamic, annual
Nsshe	Number of slaughtered sheeps	Dynamic, annual
Nsgoa	Number of slaughtered goats	Dynamic, annual
Nsbuff	Number of slaughtered buffaloes	Dynamic, annual

Latitudes, longitudes, and altitudes of province centers were taken from the website of General Directorate of Highways (`http://www.kgm.gov.tr`). Rest of the spatial covariates were obtained from the Census of Agriculture Agricultural Holdings (Households) of Turkey, which is obtained from the website of Turkish Statistical Institute (`http://www.turkstat.gov.tr`). Year and month information were extracted from the surveillance data given. 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.

Climate covariates were taken from the Climatic Research Unit database [Harris et al., 2014], and other temporal covariates were obtained from the website of Turkish Statistical Institute. The number of households was divided by the total population of each province and land related covariates were divided by the total area of each province to make these covariates comparable across different provinces.

Gaussian Processes

GP regression is a machine learning algorithm, which finds a relation between an output y (e.g., CCHF cases) and a set of inputs $\mathbf{X}$ (e.g., longitude, latitude, date, etc.). The main assumption of this model is that there is an unobserved or latent function

f that depends on $\mathbf{X}$, but for which we only have access the version with some noise, $\boldsymbol{y}$. This unobserved variable is a GP with the mean vector μ and covariance matrix $\boldsymbol{\Sigma}$, which depend on the inputs [Williams and Rasmussen, 2006]. In this study, we formulated a GP model with a Kronecker decomposition approach for spatiotemporal modeling, named structured Gaussian process (SGP), to learn covariance functions for both knowledge extraction and prediction. Our main hypothesis about the spatiotemporal processes is that response values depend on time and location. We need a kernel function (i.e., covariance function) that makes nearby observations in time and/or space produce similar values. Each spatial and temporal covariate is fed into a kernel function for SGP.

We define both spatial and temporal kernels as Hadamard multiplications of Gaussian kernels:

$$\begin{aligned} \mathbf{K}_s &= \mathbf{K}_{s,1} \circ \mathbf{K}_{s,2} \circ \cdots \circ \mathbf{K}_{s,P_s}, \\ \mathbf{K}_t &= \mathbf{K}_{t,1} \circ \mathbf{K}_{t,2} \circ \cdots \circ \mathbf{K}_{t,P_t}, \end{aligned}$$

where $\circ$ is the Hadamard product of input matrices, and P_s and P_t are the total numbers of chosen spatial and temporal kernels, respectively.

Learning step in GPs can be defined as the problem of finding proper parameters for the covariance functions. This construction provides us a model of the data and characteristics (i.e., kernel widths) that we can interpret. The adjustable parameters, namely, kernel width and variance parameters, can be learned from data using marginal likelihood or cross–validation. We want to infer the noise variance σ_y^2, spatial kernel widths of $\mathbf{K}_{s,m}$ for $m = 1, \cdots, P_s$, and temporal kernel widths of $\mathbf{K}_{t,n}$ for $n = 1, \cdots, P_t$ in order to make prediction and feature extraction simultaneously. The problem with GP prediction methods is that their computational complexity is $\mathcal{O}(n^3)$ due to the inversion of an $n \times n$ matrix, and their storage complexity is $\mathcal{O}(n^2)$ due to the storage of an $n \times n$ matrix. For large data sets, this is expensive in terms of both time and space complexity, and a number of methods have been developed for

fast computation. Matrix computations can be made more efficient using the special properties of Kronecker product [Saatçi, 2012, Bonilla et al., 2007, Andrade Pacheco, 2015, Finley et al., 2009a, Stegle et al., 2011, Gilboa et al., 2015, Riihimäki and Vehtari, 2014, Wilson et al., 2014]. We choose to learn kernel widths using a maximum likelihood approach because the required computations (integrals over the parameter space) are analytically tractable for standard GP. We apply Kronecker tricks to the equation of the likelihood, Equation (3.1) and the equations the partial derivatives of the marginal likelihood with respect to the hyper-parameters, Equation (6.8) [Williams and Rasmussen, 2006]. See Appendix, Section 6.2.2 for further details about the derivations.

Fifty-two spatial and temporal covariates were difficult to interpret, and some of them are highly correlated. That is why we excluded the highly correlated covariates (i.e., 10 temporal covariates are 98% correlated, thus we excluded the following covariates: Year (Year), maximum temperature (Tmpmx), mean temperature (Tmpme), minimum temperature (Tmpmn), total utilized agricultural land (Cagri), total arable land and land under permanent crops (Carcr), total arable land (Carab), total land under permanent crops (Ccrop), area of other fruit, beverage and spices crops (Cfrui), area of vineyard (Cvine) (See Table 4.1). We discarded land under permanent meadows and pasture (Cpast) covariate since it is constant and would have no effect in our model.

We get a better understanding about the underlying dynamics of the process to be modeled when data can be explained with fewer covariates, which may be hidden or latent factors that in combination play greater roles in the observed dynamics. In order to find these fewer but important covariates, we optimized each covariate's relative importance.

To be able to model case counts using Gaussian distribution, we first $\log_2$-scaled the CCHF surveillance data set, as it was done before in previous studies [Andrade Pacheco, 2015]. For the 2016 prediction, we used the years 2004–2015 as training sets (81 provinces $\times$ 144 months). We then used the trained model to predict case

counts of 81 provinces for 2016 (81 provinces × 12 months). For the 2017 prediction, we used the years 2004–2016 as training sets (81 provinces × 156 months). We then used the trained model to predict case counts of 81 provinces for 2017 (81 provinces × 12 months).

A study in Turkey found that areas with CCHF cases had lower mean temperatures in the late autumn and the winter [Estrada-Peña et al., 2010]. We used the fact that vector-borne disease dynamics are affected by previous year's weather conditions, animal population, etc. since vector abundance is also affected from these. In other words, covariates of this year will be used to make predictions for the case counts of next year. We trained our model using all spatial covariates, temporal covariates between 2003–2014 and case counts of years 2004–2015, then given all spatial covariates and temporal covariates of the year 2015 and, the learned parameters from our trained model we predicted the cases for 2016. The same approach was applied for 2017 predictions. We focused on prospective predictions of the years 2016 and 2017. Prediction for any given year can be done given the covariates of the previous year.

The Pearson's correlation coefficient (PCC) and normalized root mean squared error (NRMSE) were used to measure the prediction performance. Computational modelling was performed using the statistical software package R [R Core Team et al., 2013].

Source codes: The input covariates, nationwide CCHF surveillance data set and our computational results reported in this study can be publicly explored and downloaded at `http://midas.ku.edu.tr/ProspectiveCCHF/`.

4.2 Results

Spatial and temporal distribution of cases: In Turkey, 10,411 confirmed CCHF cases were reported between years 2004 and 2017, and mainly from April to October, yearly epidemic curves peaked around June and July (Figure 4.1 A). Most of these confirmed CCHF cases were reported in north and northeast regions of Anatolia (4.1 B). Detailed interpretations of the case counts are presented at `http://midas.ku.edu.tr/`

ProspectiveCCHF/.

4.2.1 Prospective Prediction of 2016 and 2017

We predicted the nationwide annual case count for 2016 as 438, whereas the observed case count was 432 (Figure 4.2). Similarly, we predicted the nationwide annual case count for 2017 as 341, whereas the observed case count was 343 (Figure 4.3). PCC and NRMSE values for 2016 prediction scenario is 0.83 and 0.58, respectively. For 2017 prediction, PCC is 0.87 and NRMSE is 0.52. Each month's prediction for all provinces on a map can be seen at http://midas.ku.edu.tr/ProspectiveCCHF/.

4.2.2 Covariate Importance

Latitude and number of settlements with fewer than 25,000 inhabitants covariates of provinces (i.e., spatial covariates) and monthly potential evapotranspiration (evaporation and transpiration) measurements (i.e., temporal covariate) were found to be the most explanatory covariates for 2016 prediction (Figure 4.4 A and Figure 4.4 B). In 2017 prediction, number of settlements with fewer than 25,000 inhabitants and longitude covariates of provinces (i.e., spatial covariates) and monthly potential evapotranspiration measurements (i.e., temporal covariate) were the most important covariates (Figure 4.4 C and Figure 4.4 D).

4.3 Discussion

Turkey has the highest number of laboratory confirmed CCHF cases, and we included all 10,441 CCHF cases into our computational analyses. We used a unified model including a rich collection of spatial and temporal data sources to determine the relative importance of each data source. We evaluated our approach by performing monthly predictions for each province in a prospective manner.

The latitude, longitude, and number of settlements with fewer than 25,000 inhabitants were found to be the most important spatial covariates for predicting CCHF

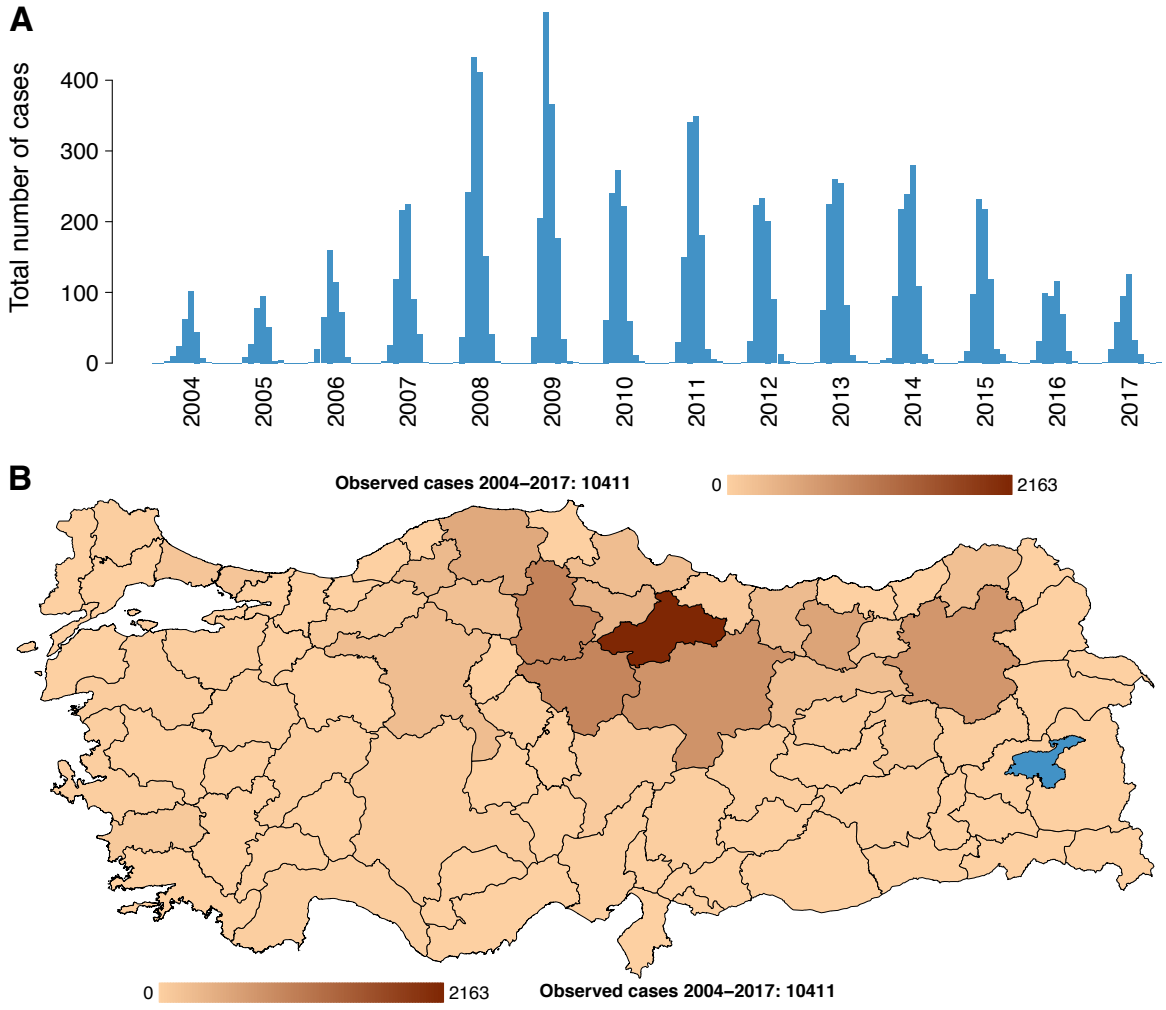

Figure 4.1: **Summary of Turkish nationwide CCHF surveillance data set.** (A) Monthly confirmed CCHF case counts between January 2004 and December 2017. (B) Total confirmed CCHF case counts for each province between years 2004 and 2017. Numbers in the legend of panel (B) correspond to the minimum and maximum numbers of observed cases in provinces between 2004 and 2017. Yearly case count maps can be seen at `http://midas.ku.edu.tr/ProspectiveCCHF/`.

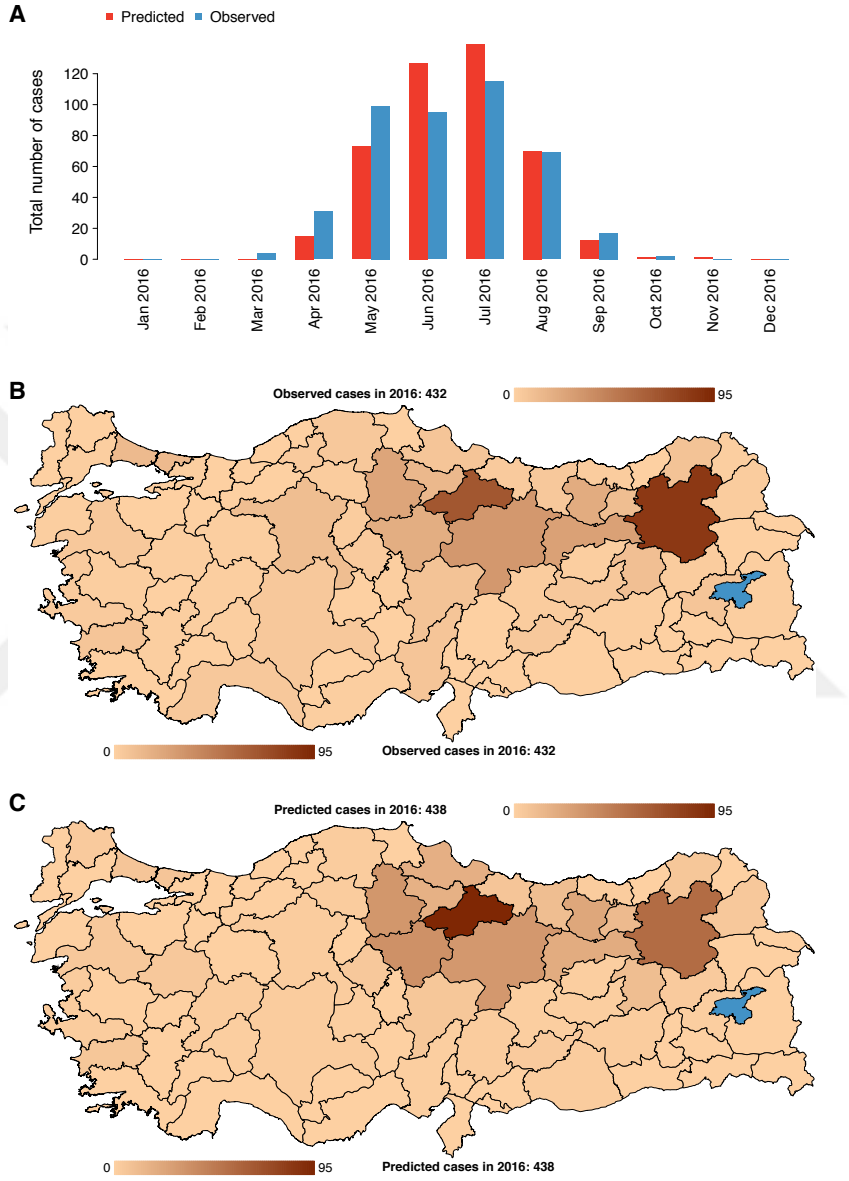

Figure 4.2: **Prediction results obtained by our SGP algorithm for 2016.** Observed cases are shown in blue and predicted cases are shown in red. (A) Monthly observed and predicted CCHF case counts for 2016. (B) Annual observed CCHF case counts for each province in 2016. (C) Annual predicted CCHF case counts for each province in 2016. Numbers in the legend of panels (B) and (C) correspond to the minimum and maximum numbers of observed and predicted cases in provinces for 2016. Monthly prediction maps can be seen at `http://midas.ku.edu.tr/ProspectiveCCHF/`.

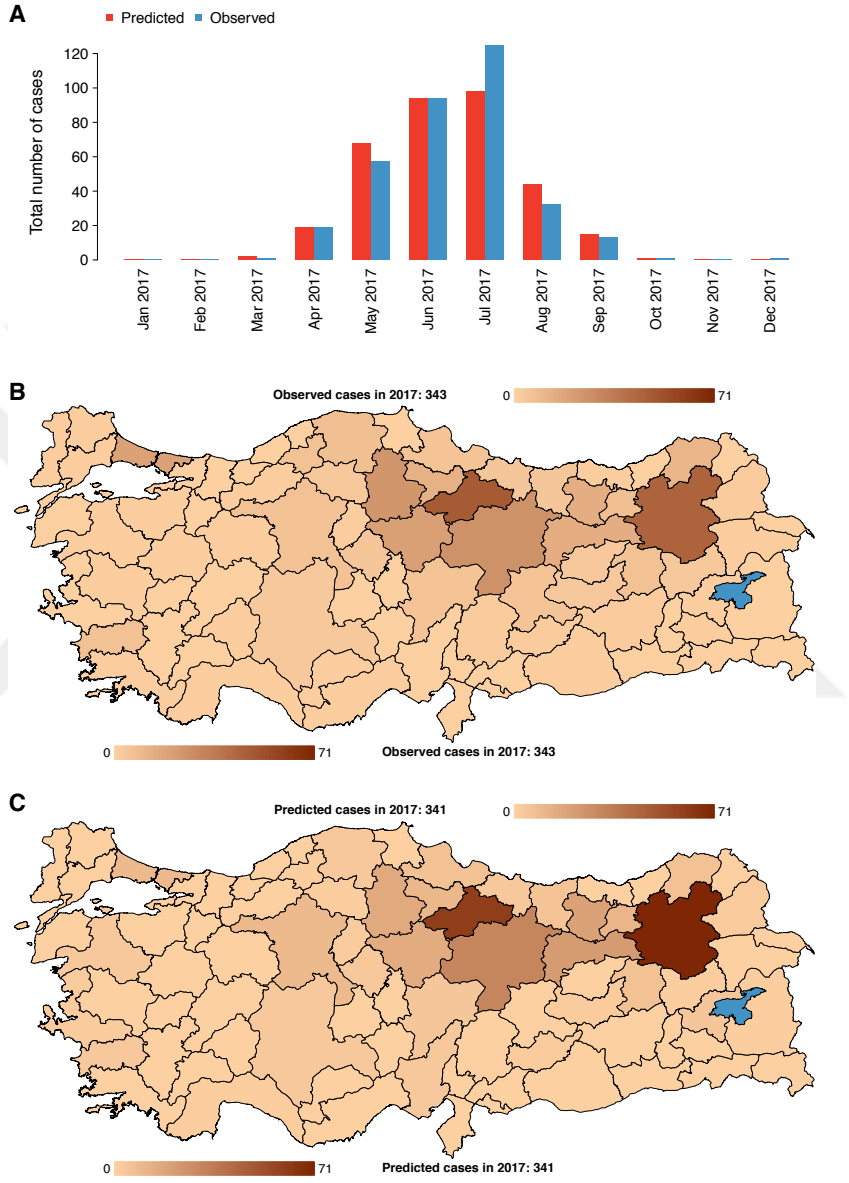

Figure 4.3: **Prediction results obtained by our SGP algorithm for 2017.**Observed cases are shown in blue and predicted cases are shown in red. (A) Monthly observed and predicted CCHF case counts for 2017. (B) Annual observed CCHF case counts for each province in 2017. (C) Annual predicted CCHF case counts for each province in 2017. Numbers in the legends of panels (B) and (C) correspond to the minimum and maximum numbers of observed and predicted cases in provinces for 2017. Monthly prediction maps can be seen at `http://midas.ku.edu.tr/ProspectiveCCHF/`.

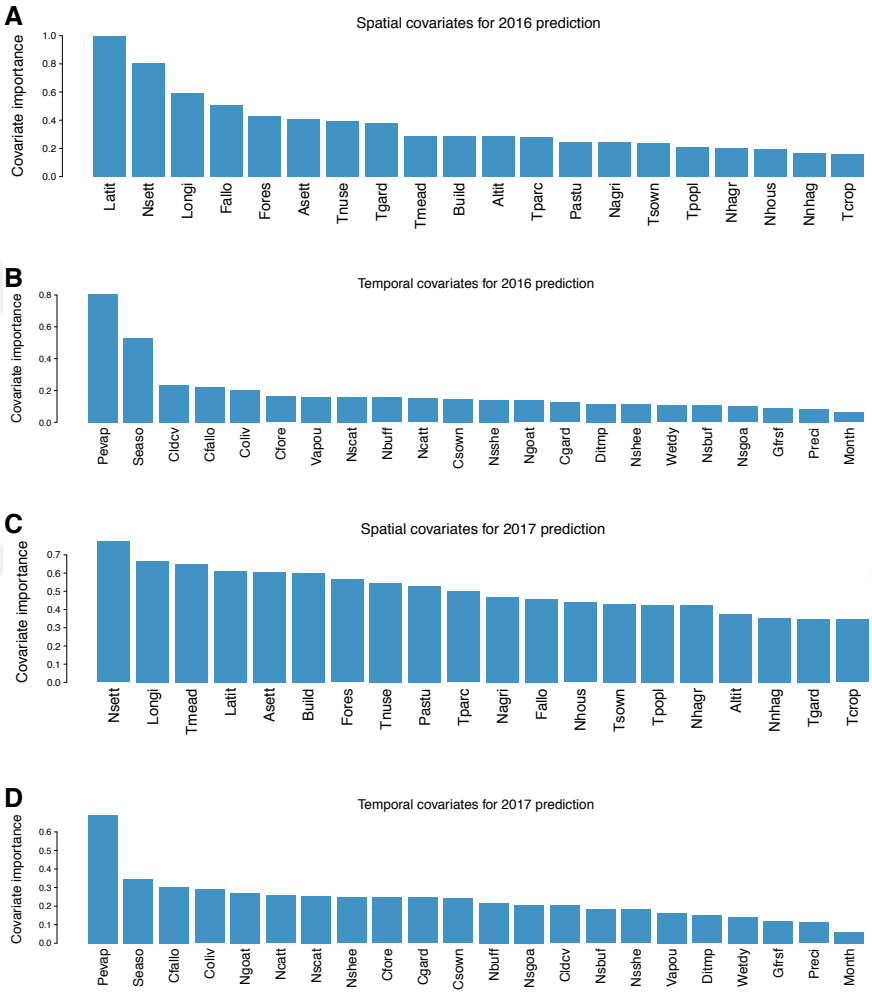

Figure 4.4: **Relative importance of spatial and temporal covariates assigned by our SGP algorithm.** See Table 4.1 for the covariates abbreviations' correspondent full name. (A) Spatial covariate importance for 2016 prediction. (B) Temporal covariate importance for 2016 prediction. (C) Spatial covariate importance for 2017 prediction. (D) Temporal covariate importance for 2017 prediction. Numbers shown in the figure are the outputs of our model, which give the relative importance of covariates.

case counts prospectively. Potential evapotranspiration and season were found to be the most informative temporal covariates for both 2016 and 2017 predictions.

Understanding how the disease spreads and identifying the related factors driving the disease dynamics are important to help elimination of the future cases. The roles of the covariates may change over years and our model helps us detect these changes because our model updates itself as the data come in.

Data were significantly focused locally, especially on the north eastern part of Anatolia, for example, 2,163 cases were from one province alone (i.e., Tokat, one of the rural provinces). This pattern was also reflected in relative importance of covariates. The importance of number of settlements with fewer than 25,000 inhabitants could be related to human population at risk living close to the habitat of ticks and animals since these settlements are situated usually in rural areas where people are engaged in agricultural activities. The number of settlements with fewer than 25,000 inhabitants is important for both years, but positions of latitude and longitude switched their rankings in terms of importance. This finding is in line with the increased number of CCHF cases in eastern parts in later years, which can be better captured by longitude rather than latitude.

The evapotranspiration is a climate variable and defined as the total water vapour produced in the water basin as a result of the growth of plants in the water basin. Potential evapotranspiration is evapotranspiration at the time when there is sufficient water available to provide to a surface completely covered with plants. This term refers to providing ideal amount of water to plants. It is also obvious that season covariate determines the temporal behaviour of CCHF or other seasonal infectious diseases in general. These two important temporal covariates confirm the role of the climate for the underlying mechanism of CCHF. Careful follow-up of these covariates may imply possible warnings in short term instead of waiting yearly predictions from our model. Higher temperature was also found as a main driver for the abundance of *H. marginatum*, previously [Ergönül, 2006, Messina et al., 2016, Estrada-Peña et al., 2010, Estrada-Peña et al., 2013] because high temperatures may fasten the cycle of

ticks and thus increase the host questing. It was also noted that areas with CCHF cases had lower mean temperatures in the late autumn and the winter [Estrada-Peña et al., 2010]. Another study proposed that the humans are more out in days with higher temperature, so that the chance of hosting a tick was increased because of the higher chance of contact with the animals [Rogers and Randolph, 2006].

In this chapter, we found that yearly change of the land of olive trees, fallow land, and forest land were more important than the animal population, Figure 4.4. Our findings were parallel with another report in which the land cover rather than climate and animal population was found to be the main driver for world-wide distribution of CCHF, and they commented that these factors might be more important in predicting in finer-scale prevalence patterns [Messina et al., 2016].

We used the annual data of husbandry from the Turkish Statistical Institute for the first time, and our model was able to reveal the importance of different animal groups, Figure 4.4. In our model for 2017 prediction, the number of goats, cattle, and sheep were found to be the most significant animals for CCHF dynamics and spread, respectively. Although these findings contradict with the observation of veterinarians on the field who claimed that bovine/cattle livestock were more important than goat livestock in the transmission cycle of the virus. This contradiction implies that there are some other underlying reasons such as the farmers, those who shepherd goat, might be dealing with their hands with or without protection. We must take into account the possible reasons why a covariate is chosen and take precaution against it respectively. The importance of covariates which may be related to human action indicates that awareness is lacking in some parts of the country about the presence of CCHF or precautions against CCHF. Our model identifies the directions we should pay close attention with high priority. For instance, in the areas with high goat, cattle or sheep density, agricultural workers and others working with animals should also be monitored and must be informed about CCHF. For further investigation, tick abundance studies in the field should be developed and improved.

Annual predictions for 2016 and 2017 are accurate but the prediction for each

province is not as much accurate (Figure 4.2 and Figure 4.3). Predicting the total number of the cases from overall seasonality is easier than capturing spatial dependencies because time series data are dependent if there is seasonality behavior of the data. One limitation of this study is that our model may not predict an outbreak if the reason of the outbreak is not related to the covariates that we use to train our model. However, when the first data of the outbreak arrive, model will update itself accordingly, but there might be some delay for accurate predictions. Another limitation is that even the surveillance data is ready, covariate data (e.g., livestock statistics) might be published much later or might be incomplete at the time of prediction. Then, the model would not be able to benefit from all the information may be related to the progress of disease's dynamics. Although these problems are valid for all data-driven models.

Our SGP algorithm was able to prospectively predict the numbers of CCHF cases using a nationwide surveillance data and a rich collection of explanatory covariates and, it was able to predict annual CCHF case counts very close to the observed case counts. Using these predictions for future infected cases, policy makers can make informed decisions on intervention or treatment strategies, for example, the amounts of vaccine purchases, the organizational details of public awareness campaigns and training programs for health care workers. Furthermore, we described the spatiotemporal dynamics of CCHF determining the relative importance of covariates. Our proof of concept study provided insights for understanding possible mechanisms of infectious diseases and found directions with high priority for practice and policy to combat against emerging infectious diseases. We tested our tool on a single disease, but the same framework can be extended towards other vector-borne infectious diseases and as well as other infectious diseases.

Chapter 5

CONCLUSION

In this thesis, we developed machine learning algorithms for spatiotemporal regression problems, which were able to make efficient and fast predictions as well as learn and understand the underlying dynamics of spatiotemporal data. Improving the predictions and making them fast, and adding more expressiveness to the model do not necessarily mean additional computational cost as we exploited the special structure of the proposed models for these purposes.

We constructed a computational framework using structured GPR for spatiotemporal modeling. Computational and storage complexities of GPs are costly. That is why we exploited special structure of our proposed structured GPR model for spatiotepmporal modeling to enable efficient and fast inference. We showed structured GPR is better than its counterparts in terms of efficiency and predictive performance. We were able to make accurate predictions given any time and any location pair, but that was not enough to answer the question why the predictions are as they are? To answer which information sources are more important, we integrated multiple kernel learning to SGP regression (i.e., SGP2MKL). In this way, we could explain the predictions and understood the dynamics of the disease to help public health policy-makers to take necessary precautions before the cases happen. We also applied SGP2MKL method to NASA's surface temperature data set. Then, for CCHF, we used every data source that we could benefit for the explanation of the disease dynamics, then by optimizing the kernel parameters we ranked the data sources with respect to their relative importance as well as improving the predictions. We also built a website where we present extensive data sets from many different sources as well as the observations and the predictions.

We hope that this thesis contributed to the area of spatiotemporal modeling with machine learning, and will lead to new questions and inspire both machine learning specialists and non-specialists for future research.

5.1 Future Work

We used Gaussian prior and Gaussian noise in our models to make inference with closed-form predictive equations. Inference with non-Gaussian priors and noise might be explored, but then efficient approximation methods will be needed for SGP or SGP2MKL.

The use of different kernels might also be explored. For example periodic kernels for seasonality effect on temporal features etc. Specific to infectious disease modeling, the CCHF data is very sparse and even sparser in finer spatial resolution. An approach especially for this kind of data can be developed for SGP and SGP2MKL using new neighborhood definitions or using different spatial similarity kernels for spatial kernels. If such an approach is developed, a study focused on the endemic region might be beneficial.

Collecting vector-related data is very difficult for all the spatial points in the data set, which is every province in the country. However, at least, for the province with the highest case counts, vector related data or other data sources for finer scales may be very helpful to see how the disease is spreading at a local scale.

The monitoring system of infectious cases can be connected to an early warning system. This strategy can be applied to any disease or any problem with spatiotemporal data sets.

Chapter 6

APPENDIX

In this chapter, we give technical details for the derivation of the equations described in the previous chapters.

6.1 Definitions and Identities

Kronecker product

Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{B}$ be a $p \times q$ matrix, then $\mathbf{A} \otimes \mathbf{B}$ is an $mp \times nq$ block matrix as follows

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}. \tag{6.1}$$

Marginalization and conditioning of Gaussians

Assume that random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ are normally distributed as follows:

$$\begin{aligned} \boldsymbol{a} &\sim \mathcal{N}(\boldsymbol{a}; \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}), \\ \boldsymbol{b} &\sim \mathcal{N}(\boldsymbol{b}; \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}). \end{aligned}$$

Then, we have the joint distribution of random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ as follows:

$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right), \tag{6.2}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma}_{aa} &= \mathrm{E}[(\boldsymbol{a}-\boldsymbol{\mu}_a)(\boldsymbol{a}-\boldsymbol{\mu}_a)^\top], \\
\boldsymbol{\Sigma}_{bb} &= \mathrm{E}[(\boldsymbol{b}-\boldsymbol{\mu}_b)(\boldsymbol{b}-\boldsymbol{\mu}_b)^\top], \\
\boldsymbol{\Sigma}_{ab} &= \mathrm{E}[(\boldsymbol{a}-\boldsymbol{\mu}_a)(\boldsymbol{b}-\boldsymbol{\mu}_b)^\top] = \boldsymbol{\Sigma}_{ba}^\top, \\
\boldsymbol{\Sigma}_{ba} &= \mathrm{E}[(\boldsymbol{b}-\boldsymbol{\mu}_b)(\boldsymbol{a}-\boldsymbol{\mu}_a)^\top] = \boldsymbol{\Sigma}_{ab}^\top.
\end{aligned}$$

The conditional distribution of the random variable $\boldsymbol{b}$ is also Gaussian as follows

$$\boldsymbol{b}|\boldsymbol{a} \sim \mathcal{N}\big(\boldsymbol{b}; \boldsymbol{\mu}_b + \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_a), \boldsymbol{\Sigma}_{bb} - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}\boldsymbol{\Sigma}_{ab}\big). \tag{6.3}$$

Matrix determinant lemma

Let $\mathbf{A}$ be an invertible square matrix, and $\boldsymbol{u}$ and $\boldsymbol{v}$ are column vectors. Then,

$$(\mathbf{A} + \boldsymbol{u}\boldsymbol{v}^\top) = (1 + \boldsymbol{v}^\top\mathbf{A}^{-1}\boldsymbol{u})\det(\mathbf{A}).$$

Generalization

Let $\mathbf{A}$ be an n-by-n invertible matrix, and $\mathbf{U}$ and $\mathbf{V}$ are n-by-m matrices. Then,

$$\det(\mathbf{A} + \mathbf{U}\mathbf{V}^\top) = \det(\mathbf{I}_m + \mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U})\det(\mathbf{A}).$$

If $\mathbf{W}$ is an m-by-m invertible matrix, then,

$$\det(\mathbf{A} + \mathbf{U}\mathbf{W}\mathbf{V}^\top) = \det(\mathbf{W}^{-1} + \mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U})\det(\mathbf{W})\det(\mathbf{A}). \tag{6.4}$$

Inverse of a diagonal matrix

Let $\mathbf{D}$ be an n-by-n diagonal matrix $\mathbf{D} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$, then, its inverse is given by

$$\mathbf{D}^{-1} = \begin{bmatrix} a_{11}^{-1} & 0 & \cdots & 0 \\ 0 & a_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{-1} \end{bmatrix}. \tag{6.5}$$

Also determinant of matrix $\mathbf{D}$ is the product of its diagonal elements:

$$\det(\mathbf{D}) = a_{11}a_{22}\cdots a_{nn}.$$

Kronecker vector-matrix multiplication rule

The vec$(\cdot)$ operator stacks the columns of a matrix into a vector. For example,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{vec}(\mathbf{A}) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix}.$$

One of the vec-operator property is as follows:

$$\text{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A})\,\text{vec}(\mathbf{X}). \tag{6.6}$$

Diagonal of a Kronecker product

Let $\mathbf{A}$ be a n-by-n matrix and $\mathbf{B}$ be a m-by-m matrix, then

$$\operatorname{diag}(\mathbf{A} \otimes \mathbf{B}) = \begin{bmatrix} a_{11} \\ a_{11} \\ \vdots \\ a_{11} \\ a_{22} \\ a_{22} \\ \vdots \\ a_{22} \\ \vdots \\ a_{nn} \\ a_{nn} \\ \vdots \\ a_{nn} \end{bmatrix} \circ \begin{bmatrix} b_{11} \\ b_{22} \\ \vdots \\ b_{mm} \\ b_{11} \\ b_{22} \\ \vdots \\ b_{mm} \\ \vdots \\ b_{11} \\ b_{22} \\ \vdots \\ b_{mm} \end{bmatrix}. \tag{6.7}$$

6.2 Derivations

6.2.1 Log Likelihood Derivations

Notice that the term $\boldsymbol{y}^\top(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}$ in the log likelihood, Equation (3.1) and in its gradient, Equation (3.2) is actually the transpose of the vector $\boldsymbol{\alpha} = (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}$.

$$\begin{aligned} \boldsymbol{\alpha}^\top &= ((\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y})^\top \\ &= \boldsymbol{y}^\top((\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top)^\top \\ &= \boldsymbol{y}^\top(\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top \\ &= \boldsymbol{y}^\top(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1} \end{aligned}$$

It is important that we compute the vector $\boldsymbol{\alpha} = (\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y}$ efficiently because it appears both in log likelihood and gradient equations (3.1) and (3.2).

Instead of calculating Kronecker multiplication and taking its inverse we will use Kronecker matrix-vector multiplication rule (see Equation(6.6)) and take only the inverse of a diagonal matrix (see Equation (6.5)).

$$\begin{aligned}
(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\boldsymbol{y} &= (\mathbf{U}_s \otimes \mathbf{U}_t)\underbrace{(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}}_{\mathbf{D}^{-1}}(\mathbf{U}_s{}^\top \otimes \mathbf{U}_t{}^\top)\boldsymbol{y} \\
&= (\mathbf{U}_s \otimes \mathbf{U}_t)\mathbf{D}^{-1}\operatorname{vec}(\mathbf{U}_t{}^\top\mathbf{Y}\mathbf{U}_s) \\
&= (\mathbf{U}_s \otimes \mathbf{U}_t)\underbrace{\left(\frac{1}{\operatorname{diag}(\mathbf{D})} \circ \operatorname{vec}(\mathbf{U}_t{}^\top\mathbf{Y}\mathbf{U}_s)\right)}_{\boldsymbol{x}} \\
&= \operatorname{vec}(\mathbf{U}_t\mathbf{X}\mathbf{U}_s{}^\top),
\end{aligned}$$

where $\boldsymbol{y} = \operatorname{vec}(\mathbf{Y})$, $\boldsymbol{x} = \operatorname{vec}(\mathbf{X})$, and $\mathbf{D} = \mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I}$. The second difficulty is calculating the determinant of $(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})$ in the log likelihood (3.1). We will use a variation of the generalization of the matrix determinant lemma (see Equation (6.4)). We set $\mathbf{A} := \sigma_y^2\mathbf{I}$, $\mathbf{U} := \mathbf{U}_s \otimes \mathbf{U}_t$, $\mathbf{W} := \mathbf{D}_s \otimes \mathbf{D}_t$ and $\mathbf{V} := \mathbf{U}$. Then, we can write $\det(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})$ as follows:

$$\begin{aligned}
\det(\sigma_y^2\mathbf{I} &+ \mathbf{K}_s \otimes \mathbf{K}_t) \\
&= \det(\sigma_y^2\mathbf{I} + (\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t)(\mathbf{U}_s^\top \otimes \mathbf{U}_t^\top)) \\
&= \det((\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + (\mathbf{U}_s{}^\top \otimes \mathbf{U}_t{}^\top)(\sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t))\det(\mathbf{D}_s \otimes \mathbf{D}_t)\det(\sigma_y^2\mathbf{I}) \\
&= \det((\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + (\sigma_y^{-2})(\mathbf{U}_s{}^\top \otimes \mathbf{U}_t{}^\top)(\mathbf{U}_s \otimes \mathbf{U}_t))\det(\mathbf{D}_s \otimes \mathbf{D}_t)\det(\sigma_y^2\mathbf{I}) \\
&= \det((\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + (\sigma_y^{-2})(\mathbf{U}_s{}^\top\mathbf{U}_s \otimes \mathbf{U}_t{}^\top\mathbf{U}_t))\det(\mathbf{D}_s \otimes \mathbf{D}_t)\det(\sigma_y^2\mathbf{I}) \\
&= \det((\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + (\sigma_y^{-2})\mathbf{I})\det(\mathbf{D}_s \otimes \mathbf{D}_t)\det(\sigma_y^2\mathbf{I}) \\
&= \det((\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + (\sigma_y^{-2})\mathbf{I})(\mathbf{D}_s \otimes \mathbf{D}_t)(\sigma_y^2\mathbf{I})) \\
&= \prod((\operatorname{diag}(\mathbf{D}_s \otimes \mathbf{D}_t)^{-1} + \operatorname{diag}(\sigma_y^{-2}\mathbf{I})) \circ (\operatorname{diag}(\mathbf{D}_s \otimes \mathbf{D}_t)\sigma_y^2)).
\end{aligned}$$

We also use the following property of logarithm function: $\log(\prod \boldsymbol{x}) = \sum \log(\boldsymbol{x})$ due to computational issues (i.e., multiplying many small numbers tends to zero and log is not defined at zero).

6.2.2 Derivatives with Respect to Kernel Parameters

We define spatial and temporal kernels as a function of Gaussian kernels (i.e., multiplication, geometric mean, and arithmetic mean). Gaussian kernels define similarity functions on spatial and temporal covariates. The Gaussian kernel function $\mathbf{k}_{\mathcal{G}}(\cdot,\cdot)$ between two data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be defined as

$$\mathbf{k}_{\mathcal{G}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\mathbf{D}^2}{2s^2}\right),$$

where s is the kernel width parameter, and $\mathbf{D}=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$.

Before mentioning the computational difficulties for gradient in Equation (3.2), we must find the derivatives of our structured kernel $\left(\text{i.e., } \frac{\partial(\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})}{\partial\boldsymbol{\theta}_j}\right)$.

Recall derivatives of Kronecker and Hadamard multiplications:

$$\begin{aligned}\partial(\mathbf{K}_1\otimes\mathbf{K}_2) &= \partial(\mathbf{K}_1)\otimes\mathbf{K}_2+\mathbf{K}_1\otimes\partial(\mathbf{K}_2),\\ \partial(\mathbf{K}_1\circ\mathbf{K}_2) &= \partial(\mathbf{K}_1)\circ\mathbf{K}_2+\mathbf{K}_1\circ\partial(\mathbf{K}_2).\end{aligned}$$

Let s_m and t_n denote the kernel width parameters of spatial and temporal kernels respectively and suppose that $\boldsymbol{\theta}=(s_m,t_n,\sigma_y)$. Then, we have:

$$\begin{cases}\dfrac{\partial(\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})}{\partial s_m}=\dfrac{\partial\mathbf{K}_s}{\partial s_m}\otimes\mathbf{K}_t\\ \dfrac{\partial(\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})}{\partial t_n}=\mathbf{K}_s\otimes\dfrac{\partial\mathbf{K}_t}{\partial t_n}\\ \dfrac{\partial(\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})}{\partial\sigma_y}=2\sigma_y\mathbf{I}\end{cases}\tag{6.8}$$

since $\dfrac{\partial\mathbf{K}_s}{\partial t_n}$, $\dfrac{\partial\mathbf{K}_t}{\partial s_m}$, $\dfrac{\partial(\sigma_{\mathbf{y}}^{\mathbf{2}}\mathbf{I})}{\partial s_m}$ and $\dfrac{\partial(\sigma_{\mathbf{y}}^{\mathbf{2}}\mathbf{I})}{\partial t_n}$, they all are zero.

The derivatives of each spatial and temporal Gaussian kernel will be needed in Equations (6.8) in the next section, when calculating $\dfrac{\partial\mathbf{K}_s}{\partial s_m}$ and $\dfrac{\partial\mathbf{K}_t}{\partial t_n}$, that we can write as follows:

$$\frac{\partial\mathbf{k}_{\mathcal{G}}}{\partial s}=\frac{\mathbf{D}^2}{s^3}\circ\exp\left(-\frac{\mathbf{D}^2}{2s^2}\right)=\frac{\mathbf{D}^2}{s^3}\circ\mathbf{k}_{\mathcal{G}}.$$

In Chapter 4, we defined spatial kernel $\mathbf{K}_s$ and temporal kernel $\mathbf{K}_t$ as multiplication of Gaussian kernels:

$$\begin{aligned}\mathbf{K}_s &= \mathbf{K}_{s,1} \circ \mathbf{K}_{s,2} \circ \cdots \circ \mathbf{K}_{s,P_s},\\ \mathbf{K}_t &= \mathbf{K}_{t,1} \circ \mathbf{K}_{t,2} \circ \cdots \circ \mathbf{K}_{t,P_t}.\end{aligned}$$

In Equation (6.8), we need the derivatives of each kernel with respect to the parameter of each Gaussian kernels. By using derivatives of Hadamard multiplication rule, we can calculate them as follows:

$$\begin{aligned}\frac{\partial \mathbf{K}_s}{\partial s_m} &= \left(\frac{\mathbf{D}^2}{s_m^3} \circ \mathbf{K}_{s,m}\right) \circ (\mathbf{K}_{s,1} \circ \cdots \mathbf{K}_{s,m-1} \circ \mathbf{K}_{s,m+1} \cdots \circ \mathbf{K}_{s,P_s}) = \frac{\mathbf{D}^2}{s_m^3} \circ \mathbf{K}_s,\\ \frac{\partial \mathbf{K}_t}{\partial t_n} &= \left(\frac{\mathbf{D}^2}{t_n^3} \circ \mathbf{K}_{t,n}\right) \circ (\mathbf{K}_1 \circ \cdots \mathbf{K}_{t,n-1} \circ \mathbf{K}_{t,n+1} \cdots \circ \mathbf{K}_{t,P_t}) = \frac{\mathbf{D}^2}{t_n^3} \circ \mathbf{K}_t.\end{aligned}$$

We replace these derivatives in Equations (6.8), then in gradient equation (3.2). We obtain three general gradient equation (for each kernel parameter in the Kronecker multiplication and for noise variance parameter).

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial s_m} &= \frac{1}{2}\boldsymbol{\alpha}^\top \left(\left(\frac{\mathbf{D}^2}{s_m^3} \circ \mathbf{K}_s\right) \otimes \mathbf{K}_t\right)\boldsymbol{\alpha}\\ &\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\left(\left(\frac{\mathbf{D}^2}{s_m^3} \circ \mathbf{K}_s\right) \otimes \mathbf{K}_t\right)\right)\\ \frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial t_n} &= \frac{1}{2}\boldsymbol{\alpha}^\top \left(\mathbf{K}_s \otimes \left(\frac{\mathbf{D}^2}{t_n^3} \circ \mathbf{K}_t\right)\right)\boldsymbol{\alpha}\\ &\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}\left(\mathbf{K}_s \otimes \left(\frac{\mathbf{D}^2}{t_n^3} \circ \mathbf{K}_t\right)\right)\right)\\ \frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial \sigma_y} &= \frac{1}{2}\boldsymbol{\alpha}^\top(2\sigma_y\mathbf{I})\boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2\mathbf{I})^{-1}(2\sigma_y\mathbf{I})\right)\end{aligned}$$

In these equations, we have two issues to deal with. First, we apply the Kronecker vector-matrix multiplication rule to the Kronecker product and $\boldsymbol{\alpha}$ in the first part of the gradient equation, where $\boldsymbol{\alpha} := \operatorname{vec}(\mathbf{A})$, then it is just a vector multiplication with $\boldsymbol{\alpha}^\top$. Second, we use some trace and Kronecker product rule in order to simplify inside the trace function. Here, we give the procedure for the derivative with respect to the

spatial parameter. Same derivation can be applied to the derivative with respect to the temporal parameters.

$$
\begin{aligned}
&\frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial s_m} \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\left(\mathbf{K}_t\mathbf{A}\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)^\top\right) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\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\left(\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)\otimes\mathbf{K}_t\right)\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\left(\mathbf{K}_t\mathbf{A}\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)^\top\right) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{D}_s\otimes\mathbf{D}_t+\sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s\otimes\mathbf{U}_t)^\top\left(\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)\otimes\mathbf{K}_t\right)(\mathbf{U}_s\otimes\mathbf{U}_t)\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\left(\mathbf{K}_t\mathbf{A}\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)^\top\right) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{D}_s\otimes\mathbf{D}_t+\sigma_y^2\mathbf{I})^{-1}\left(\left(\mathbf{U}_s{}^\top\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)\mathbf{U}_s\right)\otimes\left(\mathbf{U}_t{}^\top\mathbf{K}_t\mathbf{U}_t\right)\right)\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\left(\mathbf{K}_t\mathbf{A}\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)^\top\right) \\
&\quad - \frac{1}{2}\operatorname{tr}\left(\underbrace{(\mathbf{D}_s\otimes\mathbf{D}_t+\sigma^2\mathbf{I})^{-1}}_{\mathbf{D}^{-1}}\underbrace{\left(\left(\mathbf{U}_s{}^\top\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)\mathbf{U}_s\right)\otimes\mathbf{D}_t\right)}_{\mathbf{B}}\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\left(\mathbf{K}_t\mathbf{A}\left(\frac{\mathbf{D}^2}{s_m^3}\circ\mathbf{K}_s\right)^\top\right) - \frac{1}{2}\sum(\operatorname{diag}(\mathbf{D}^{-1})\circ\operatorname{diag}(\mathbf{B}))
\end{aligned}
$$

In order to calculate the trace, we need the diagonal of the matrix multiplication in the trace function. Since first matrix is a diagonal matrix, we can perform an element-wise multiplication of diagonals of two matrices and, then, sum them. We also use an efficient way to obtain the diagonal of a Kronecker multiplication in the second matrix $\mathbf{B}$(see Equation (6.7)).

We apply the same simplifications in the next sections.

6.2.3 Derivatives with Respect to Kernel Weights

In Chapter 3.1.1, we define spatial kernel $\mathbf{K}_s$ and temporal kernel $\mathbf{K}_t$ as linear combination of Gaussian kernels.

In order to learn the weight of each kernel we need the derivatives of each kernel with respect to spatial and temporal kernel weights $\eta_{s,m}$ and $\eta_{t,n}$ in Equations (3.3)-(3.5):

$$\begin{aligned}\frac{\partial \mathbf{K}_s}{\partial \eta_{s,m}} &= \mathbf{K}_{s,m},\\ \frac{\partial \mathbf{K}_t}{\partial \eta_{t,n}} &= \mathbf{K}_{t,n}.\end{aligned}$$

We replace these derivatives in Equations (3.3)-(3.5), then in gradient equation (3.2). We obtain three general gradient equations (for spatial kernels weights, temporal weights and for σ_y parameter).

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial \boldsymbol{\eta}_{s,m}} &= \frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{K}_{s,m}\otimes\mathbf{K}_t)\boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})^{-1}(\mathbf{K}_{s,m}\otimes\mathbf{K}_t)\right)\\ \frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial \boldsymbol{\eta}_{t,n}} &= \frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{K}_{s}\otimes\mathbf{K}_{t,n})\boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})^{-1}(\mathbf{K}_{s}\otimes\mathbf{K}_{t,n})\right)\\ \frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial \sigma_y} &= \frac{1}{2}\boldsymbol{\alpha}^\top(2\sigma_y\mathbf{I})\boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left((\mathbf{K}_s\otimes\mathbf{K}_t+\sigma_y^2\mathbf{I})^{-1}(2\sigma_y\mathbf{I})\right)\end{aligned}$$

In these equations, we have two problems to deal with. First, we apply the Kronecker vector-matrix multiplication rule to the Kronecker product and $\boldsymbol{\alpha}$ in the first part of the gradient equation where $\boldsymbol{\alpha} := \operatorname{vec}(\mathbf{A})$, then it is just a vector multiplication with $\boldsymbol{\alpha}^\top$. Second, we use some trace and Kronecker product rule in order to simplify inside the trace function. Here, we show the procedure for the derivative with respect to a spatial parameter. Same derivation can also be applied to the derivative

with respect to a temporal parameter.

$$
\begin{aligned}
\frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta})}{\partial \boldsymbol{\eta}_{s,m}} &= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\big(\mathbf{K}_t\mathbf{A}\mathbf{K}_{s,m}^\top\big) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top(\mathbf{K}_{s,m} \otimes \mathbf{K}_t)\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\big(\mathbf{K}_t\mathbf{A}\mathbf{K}_{s,m}^\top\big) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top(\mathbf{K}_{s,m} \otimes \mathbf{K}_t)(\mathbf{U}_s \otimes \mathbf{U}_t)\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\big(\mathbf{K}_t\mathbf{A}\mathbf{K}_{s,m}^\top\big) \\
&\quad - \frac{1}{2}\operatorname{tr}\left((\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}((\mathbf{U}_s{}^\top\mathbf{K}_{s,m}\mathbf{U}_s) \otimes (\mathbf{U}_t{}^\top\mathbf{K}_t\mathbf{U}_t))\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\big(\mathbf{K}_t\mathbf{A}\mathbf{K}_{s,m}^\top\big) \\
&\quad - \frac{1}{2}\operatorname{tr}\Big(\underbrace{(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2\mathbf{I})^{-1}}_{\mathbf{D}^{-1}}\underbrace{((\mathbf{U}_s{}^\top\mathbf{K}_{s,m}\mathbf{U}_s) \otimes \mathbf{D}_t)}_{\mathbf{B}}\Big) \\
&= \frac{1}{2}\boldsymbol{\alpha}^\top \operatorname{vec}\big(\mathbf{K}_t\mathbf{A}\mathbf{K}_{s,m}^\top\big) - \frac{1}{2}\sum(\operatorname{diag}(\mathbf{D}^{-1}) \circ \operatorname{diag}(\mathbf{B}))
\end{aligned}
$$

In order to calculate the trace, we need the diagonal of the matrix multiplication in the trace function. Since first matrix is a diagonal matrix, we can perform an element-wise multiplication of diagonals of two matrices and, then, sum them. We also use an efficient way to obtain the diagonal of a Kronecker multiplication in the second matrix $\mathbf{B}$(see Equation (6.6)).

Finally, we can simplify the calculations for the derivative with respect to σ_y parameter as follows:

$$
\begin{aligned}
\frac{\partial \log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta})}{\partial \sigma_y} &= \frac{1}{2}\boldsymbol{\alpha}^\top (2\sigma_y \mathbf{I})\boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left(2\sigma_y(\mathbf{K}_s \otimes \mathbf{K}_t + \sigma_y^2 \mathbf{I})^{-1}\right) \\
&= \sigma_y \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left(2\sigma_y((\mathbf{U}_s \otimes \mathbf{U}_t)(\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2 \mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top\right) \\
&= \sigma_y \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left(2\sigma_y((\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2 \mathbf{I})^{-1}(\mathbf{U}_s \otimes \mathbf{U}_t)^\top(\mathbf{U}_s \otimes \mathbf{U}_t)\right) \\
&= \sigma_y \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \frac{1}{2}\operatorname{tr}\left(2\sigma_y((\mathbf{D}_s \otimes \mathbf{D}_t + \sigma_y^2 \mathbf{I})^{-1}(\mathbf{U}_s{}^\top \mathbf{U}_s \otimes \mathbf{U}_t{}^\top \mathbf{U}_t)\right) \\
&= \sigma_y \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \frac{1}{2}\sum(2\sigma_y \operatorname{diag}(\mathbf{D}^{-1})).
\end{aligned}
$$

Chapter 7

SUPPLEMENTARY FIGURES

In this chapter, we give supplementary figures.

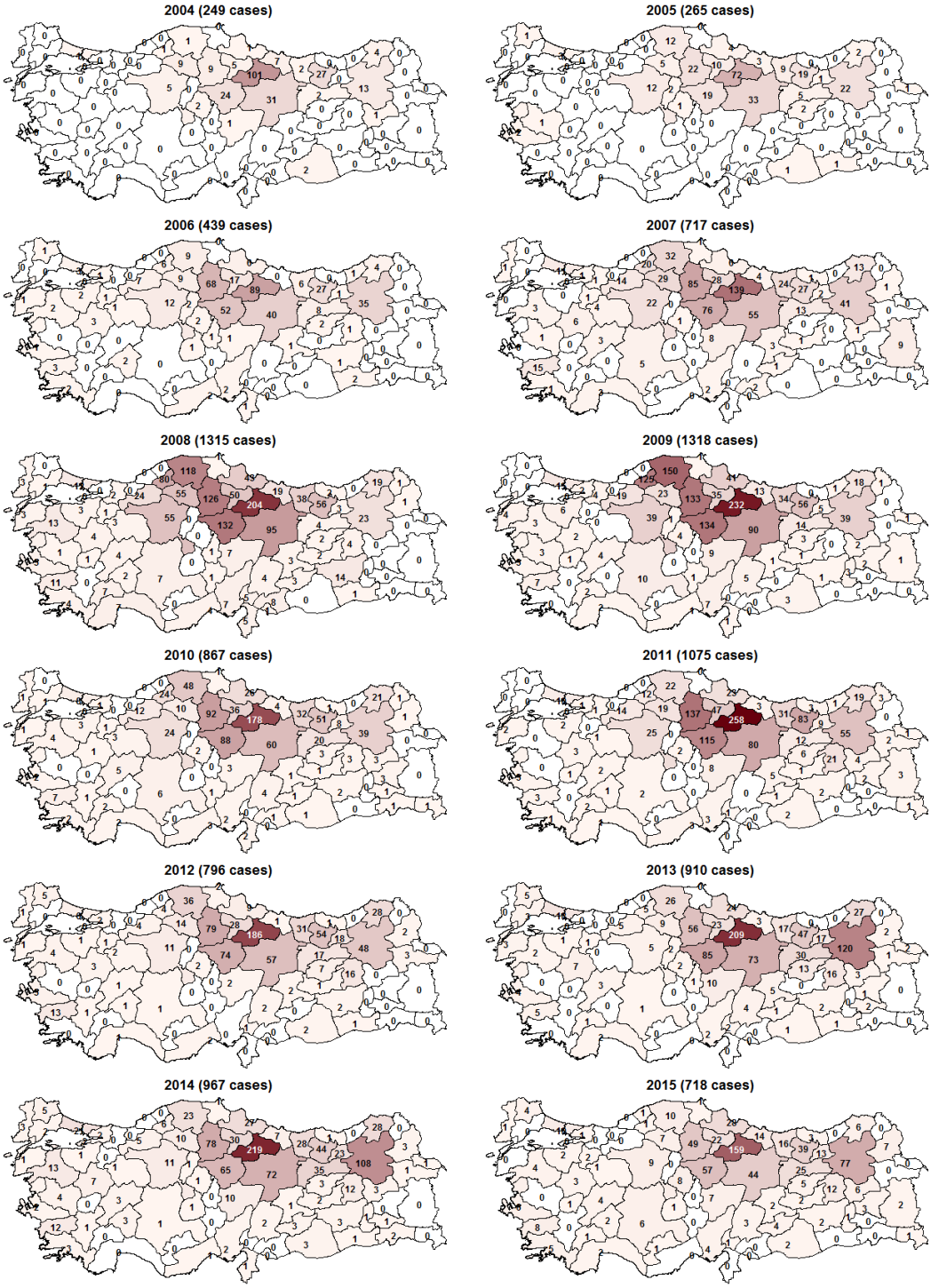

Figure 7.1: **Yearly CCHF case counts between years 2004 and 2015 for 81 provinces of Turkey.**

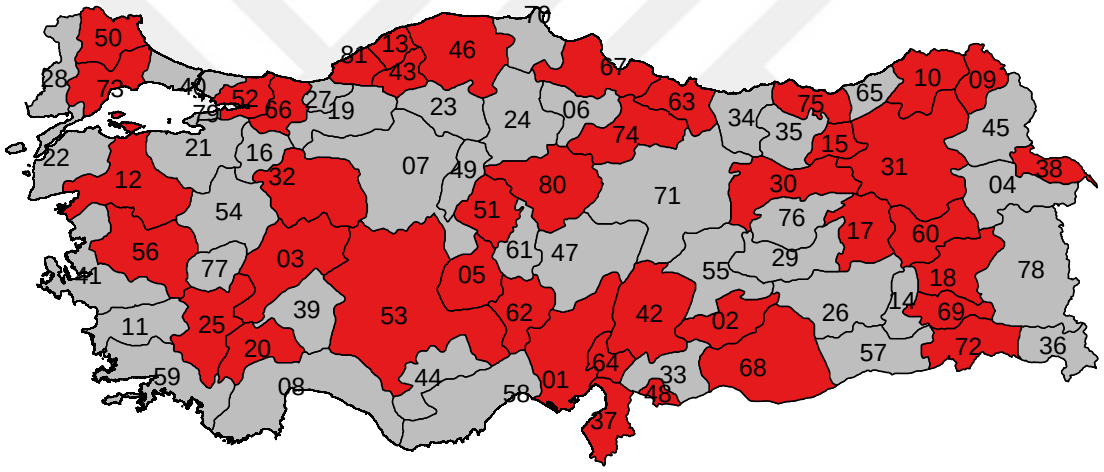

Figure 7.2: **Training and test set split of 81 provinces for spatial and spatiotemporal modeling scenarios.** Red-colored 41 provinces were used as the training set, whereas gray-colored 40 provinces were used as the test test. Province IDs were shown on the province centers. This map was generated using the Turkish administrative map downloaded from `https://www.gadm.org` and the R package maps version 3.3.0 at `https://cran.r-project.org/web/packages/maps`.

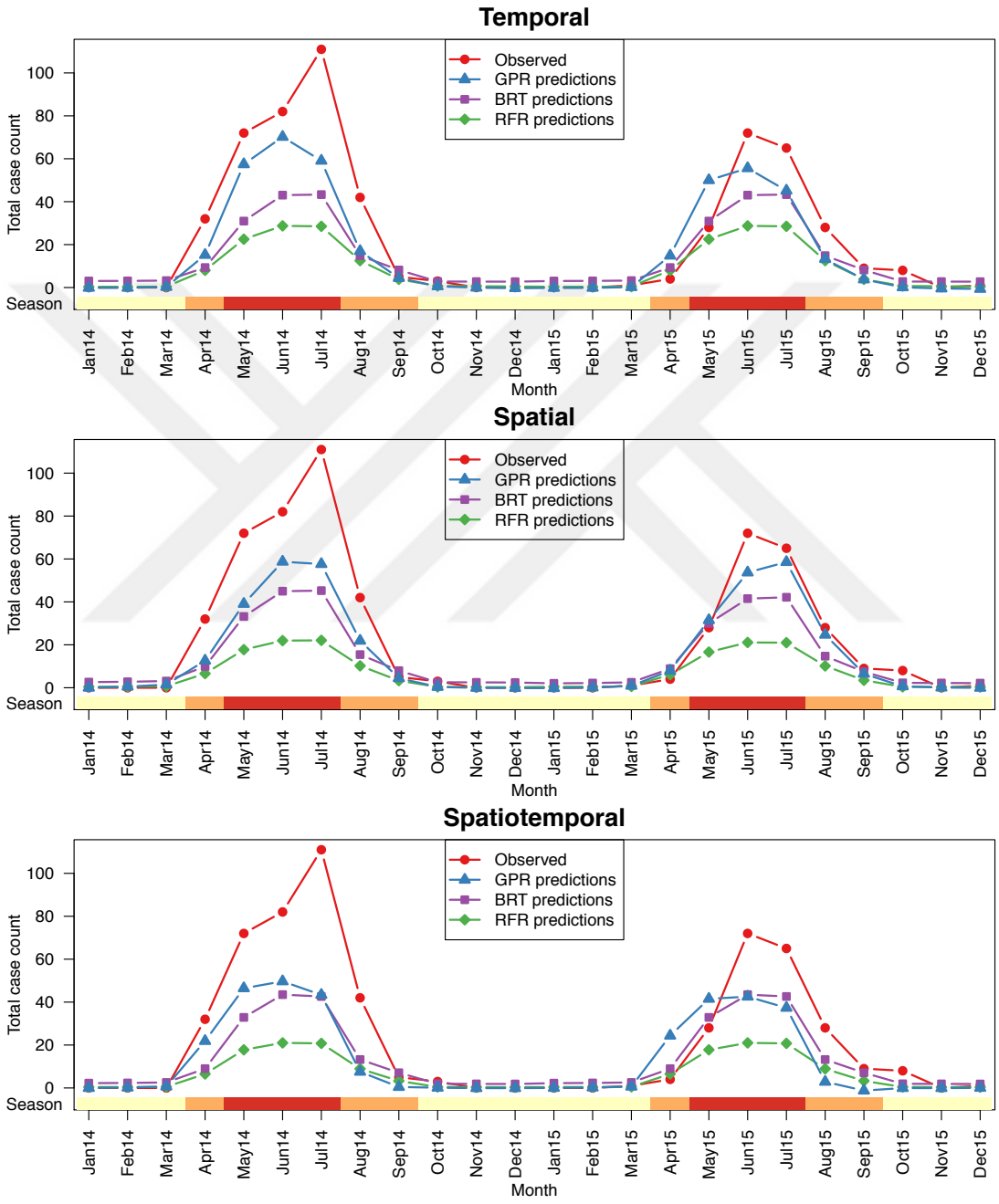

Figure 7.3: **The total observed and predicted case counts by each algorithm for years 2014 and 2015 over the 10 provinces with the highest case counts among 40 common test provinces of all scenarios.** The time periods were annotated by their seasonal group information at the top (yellow: cold; orange: warm; red: hot).

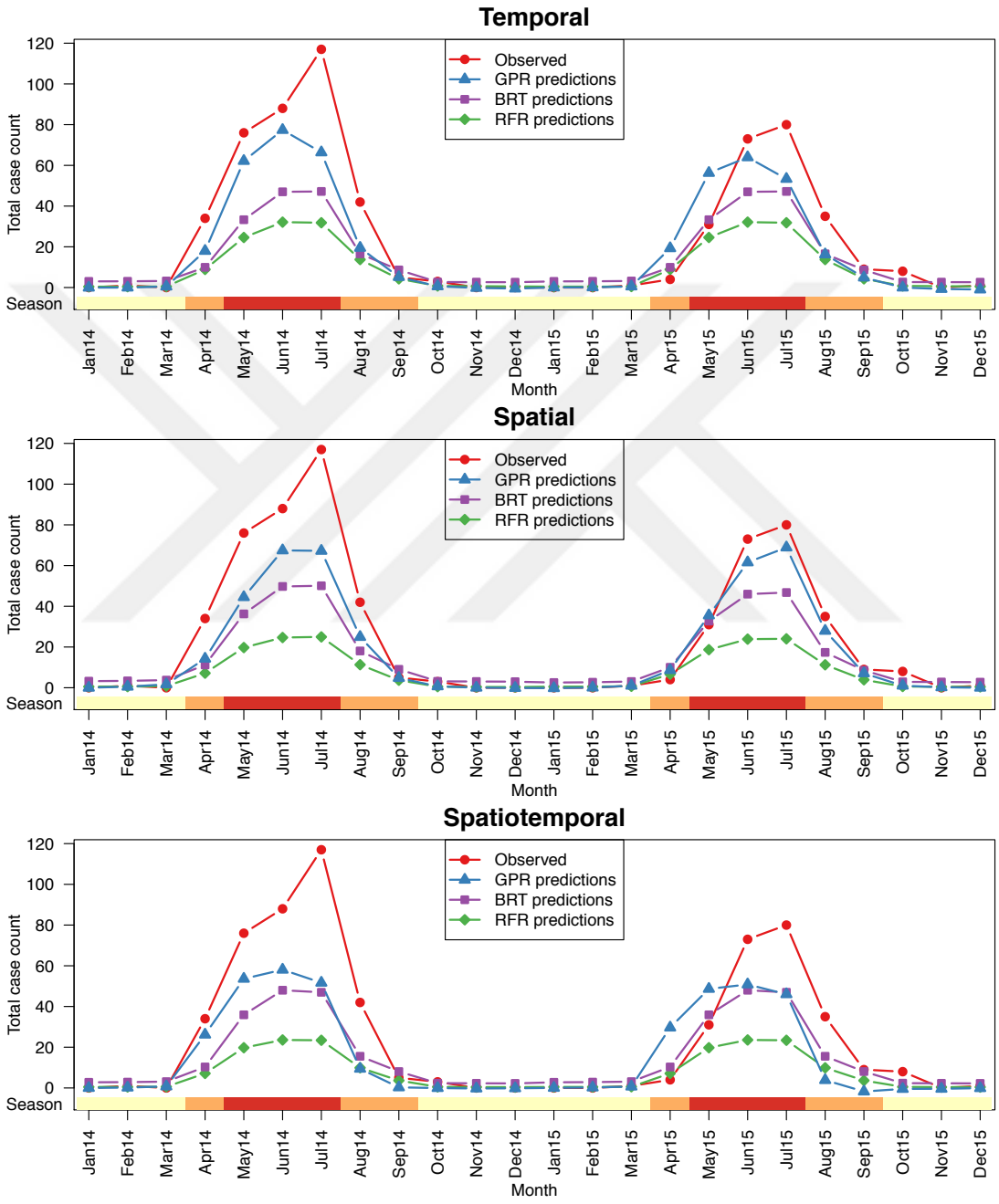

Figure 7.4: **The total observed and predicted case counts by each algorithm for years 2014 and 2015 over the 15 provinces with the highest case counts among 40 common test provinces of all scenarios.** The time periods were annotated by their seasonal group information at the top (yellow: cold; orange: warm; red: hot).

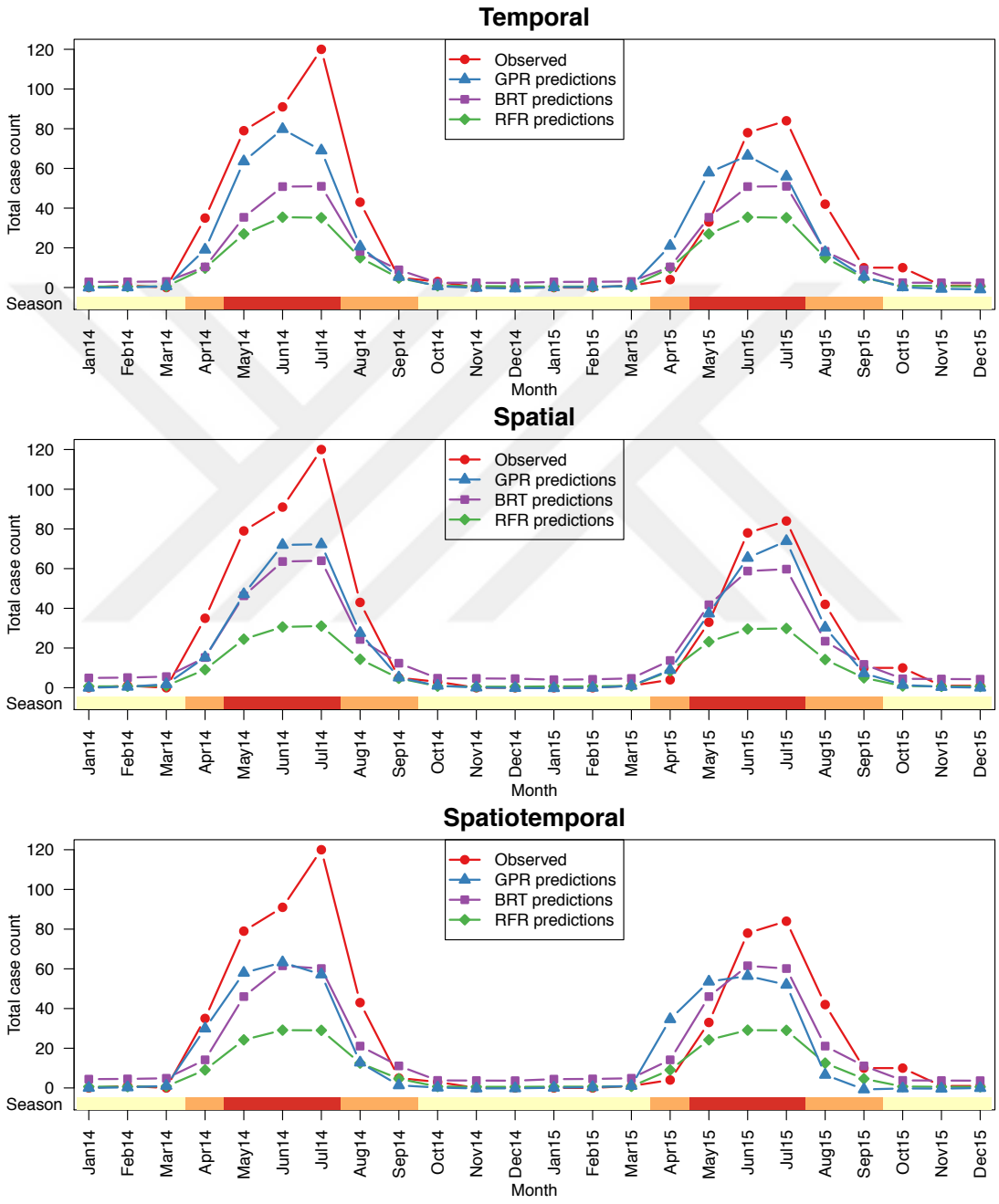

Figure 7.5: **The total observed and predicted case counts by each algorithm for years 2014 and 2015 over the 20 provinces with the highest case counts among 40 common test provinces of all scenarios.** The time periods were annotated by their seasonal group information at the top (yellow: cold; orange: warm; red: hot).

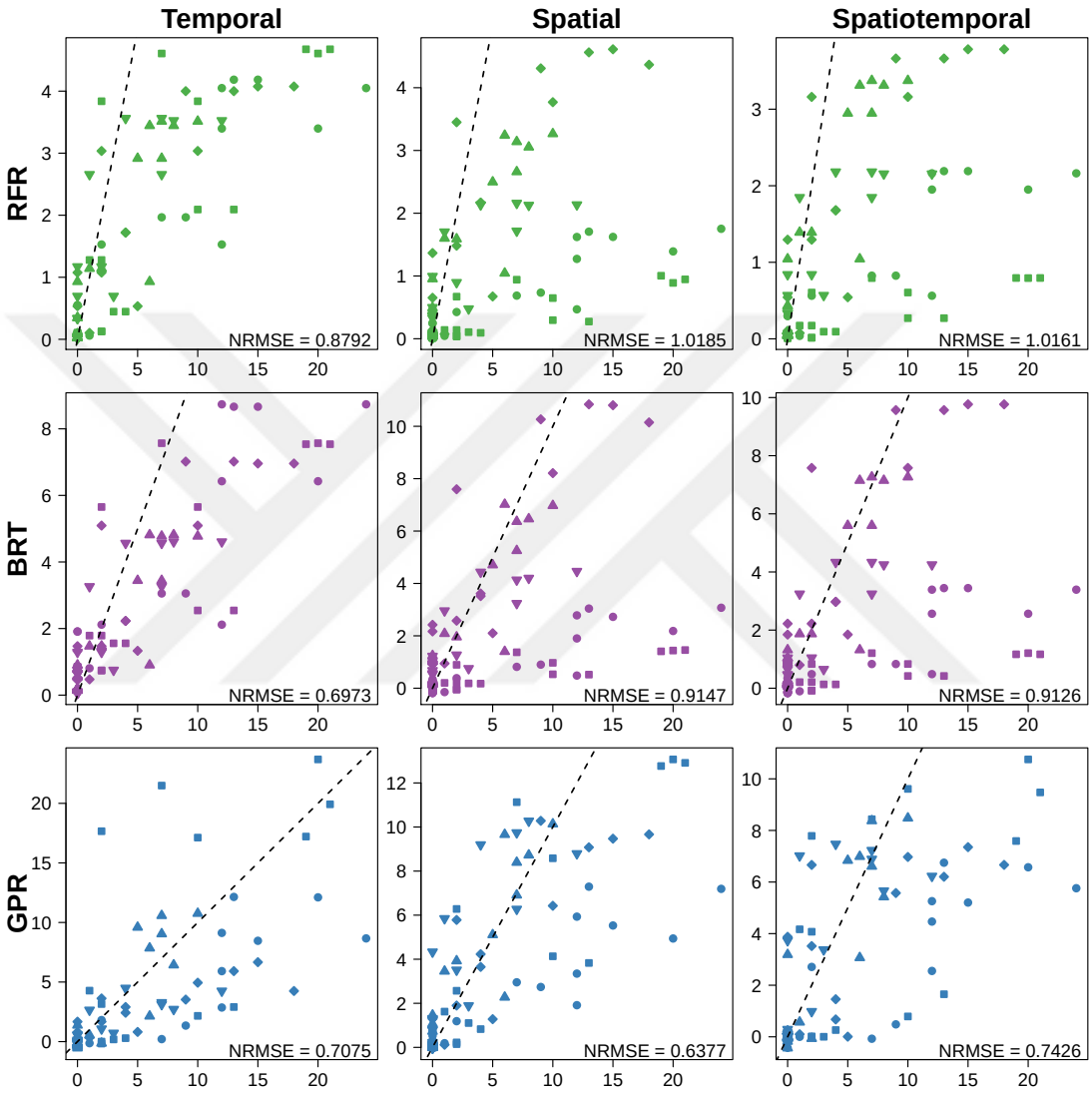

Figure 7.6: **The observed (x-axis) and predicted case counts (y-axis) by each algorithm in time periods of years 2014 and 2015 for the five provinces with the highest case counts among 40 common test provinces of all scenarios.** Each province was represented with a distinct marker. We also reported NRMSE values for each algorithm and scenario pair at the bottom-right corner. We also drew a dashed unit slope line to show whether the algorithms captured the range of observed CCHF case counts. Note that BRT and GPR algorithms obtained comparable results for temporal scenario, whereas GPR algorithm achieved remarkably better prediction performances than RFR and BRT algorithms under other two scenarios.

BIBLIOGRAPHY

[Airola and Pahikkala, 2018] Airola, A. and Pahikkala, T. (2018). Fast Kronecker product kernel methods via generalized vec trick. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3374–3387.

[Ak et al., 2018a] Ak, Ç., Ergönül, Ö., Şencan, İ., Torunoğlu, M. A., and Gönen, M. (2018a). Spatiotemporal prediction of infectious diseases using structured Gaussian processes with application to Crimean–Congo hemorrhagic fever. *PLoS Neglected Tropical Diseases*, 12(8):e0006737.

[Ak et al., 2018b] Ak, Ç., Ergönül, Ö., and Gönen, M. (2018b). Structured gaussian processes with twin multiple kernel learning. *Proceedings of Machine Learning Research*, 95:1–16.

[Ak et al., 2019] Ak, Ç., Ergönül, Ö., and Gönen, M. (2019). A prospective prediction tool for understanding Crimean–Congo haemorrhagic fever dynamics in Turkey. *Clinical Microbiology and Infection*.

[Andrade Pacheco, 2015] Andrade Pacheco, R. (2015). *Gaussian processes for spatiotemporal modelling*. PhD thesis, University of Sheffield.

[Andrade-Pacheco et al., 2014] Andrade-Pacheco, R., Mubangizi, M., Quinn, J., and Lawrence, N. D. (2014). Consistent mapping of government malaria records across a changing territory delimitation. *Malaria Journal*, 13(Suppl 1):P5.

[Ansari et al., 2014] Ansari, H., Shahbaz, B., Izadi, S., Zeinali, M., Tabatabaee, S. M., Mahmoodi, M., Holakouie Naieni, K., and Mansournia, M. A. (2014). Crimean–Congo hemorrhagic fever and its relationship with climate factors in

southeast Iran: a 13-year experience. *Journal of Infection in Developing Countries*, 8(6):749–757.

[Bhatt et al., 2017] Bhatt, S., Cameron, E., Flaxman, S. R., Weiss, D. J., Smith, D. L., and Gething, P. W. (2017). Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalization. *Journal of the Royal Society, Interface*, 14(134):pii: 20170520.

[Bhatt et al., 2013] Bhatt, S., Gething, P. W., Brady, O. J., Messina, J. P., Farlow, A. W., Moyes, C. L., Drake, J. M., Brownstein, J. S., Hoen, A. G., Sankoh, O., et al. (2013). The global distribution and burden of dengue. *Nature*, 496(7446):504.

[Bogoch et al., 2015] Bogoch, I. I., Creatore, M. I., Cetron, M. S., Brownstein, J. S., Pesik, N., Miniota, J., Tam, T., Hu, W., Nicolucci, A., Ahmed, S., Yoon, J. W., Berry, I., Hay, S. I., Anema, A., Tatem, A. J., MacFadden, D., German, M., and Khan, K. (2015). Assessment of the potential for international dissemination of Ebola virus via commercial air travel during the 2014 west African outbreak. *Lancet*, 385(9962):29–35.

[Bonilla et al., 2007] Bonilla, E. V., Ming, K., Chai, A., and Williams, C. K. I. (2007). Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Camacho et al., 2017] Camacho, A., Eggo, R. M., Goeyvaerts, N., Vandebosch, A., Mogg, R., Funk, S., Kucharski, A. J., Watson, C. H., Vangeneugden, T., and Edmunds, W. J. (2017). Real-time dynamic modelling for the design of a cluster-randomized phase 3 Ebola vaccine trial in Sierra Leone. *Vaccine*, 35(4):544–551.

[Cappelle et al., 2010] Cappelle, J., Girard, O., Fofana, B., Gaidet, N., and Gilbert, M. (2010). Ecological modeling of the spatial distribution of wild waterbirds to

identify the main areas where avian influenza viruses are circulating in the Inner Niger Delta, Mali. *EcoHealth*, 7(3):283–293.

[Chen et al., 2013] Chen, N., Qian, Z., Nabney, I. T., and Meng, X. (2013). Short-term wind power forecasting using Gaussian processes. In *23rd International Joint Conference on Artificial Intelligence*, pages 2790–2796.

[Ducheyne et al., 2015] Ducheyne, E., Charlier, J., Vercruysse, J., Rinaldi, L., Biggeri, A., Demeler, J., Brandt, C., de Waal, T., Selemetas, N., Höglund, J., et al. (2015). Modelling the spatial distribution of Fasciola hepatica in dairy cattle in Europe. *Geospatial Health*, 9(2):261–270.

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

[Ergönül, 2012] Ergönül, Ö. (2012). Crimean–Congo hemorrhagic fever virus: new outbreaks, new discoveries. *Current Opinion in Virology*, 2(2):215–220.

[Ergönül et al., 2005] Ergönül, Ö., Akgunduz, S., Kocaman, I., Vatansever, Z., and Korten, V. (2005). Changes in temperature and the Crimean–Congo haemorrhagic fever outbreak in Turkey. *Clinical Microbiology & Infection Supplement*, 11.

[Ergönül and Whitehouse, 2007] Ergönül, Ö. and Whitehouse, C. (2007). Crimean–Congo hemorrhagic fever, a global perspective. *Springer*, 10:1402061056.

[Estrada-Peña et al., 2013] Estrada-Peña, A., Ruiz-Fons, F., Acevedo, P., Gortazar, C., and de la Fuente, J. (2013). Factors driving the circulation and possible expansion of Crimean–Congo haemorrhagic fever virus in the western Palearctic. *Journal of Applied Microbiology*, 114(1):278–286.

[Estrada-Peña et al., 2007a] Estrada-Peña, A., Vatansever, Z., Gargili, A., and Buzgan, T. (2007a). An early warning system for Crimean-Congo haemorrhagic fever

seasonality in Turkey based on remote sensing technology. *Geospatial Health*, 2(1):127–135.

[Estrada-Peña et al., 2010] Estrada-Peña, A., Vatansever, Z., Gargili, A., and Ergönul, O. (2010). The trend towards habitat fragmentation is the key factor driving the spread of Crimean-Congo haemorrhagic fever. *Epidemiology and Infection*, 138(8):1194–1203.

[Estrada-Peña et al., 2007b] Estrada-Peña, A., Zatansever, Z., Gargili, A., Aktas, M., Uzun, R., Ergonul, O., and Jongejan, F. (2007b). Modeling the spatial distribution of Crimean–Congo hemorrhagic fever outbreaks in Turkey. *Vector-Borne and Zoonotic Diseases*, 7(4):667–678.

[Finley et al., 2009a] Finley, A. O., Banerjee, S., Waldmann, P., and Ericsson, T. (2009a). Hierarchical spatial modeling of additive and dominance genetic variance for large spatial trial datasets. *Biometrics*, 65:441–451.

[Finley et al., 2009b] Finley, A. O., Banerjee, S., Waldmann, P., and Ericsson, T. (2009b). Hierarchical spatial modeling of additive and dominance genetic variance for large spatial trial datasets. *Biometrics*, 65(2):441–451.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

[Friedman, 2002] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

[Gilboa et al., 2015] Gilboa, E., Saatçi, Y., and Cunningham, J. P. (2015). Scaling multidimensional inference for structured Gaussian pocesses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):424–436.

[Gönen et al., 2011] Gönen, M., Alpaydın, E., and Bach, F. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268.

[Harris et al., 2014] Harris, I., Jones, P. D., Osborn, T. J., and Lister, D. H. (2014). Updated high-resolution grids of monthly climatic observations - the CRU TS3.10 Dataset. *International Journal of Climatology*, 34:623–642.

[Harris et al., 2012] Harris, M., Reza, J., et al. (2012). *Global report for research on infectious diseases of poverty*. World Health Organization.

[Hay et al., 2013] Hay, S. I., George, D. B., Moyes, C. L., and Brownstein, J. S. (2013). Big data opportunities for global infectious disease surveillance. *PLoS Medicine*, 10(4):e1001413.

[Holmes et al., 2018] Holmes, E. C., Rambaut, A., and Andersen, K. G. (2018). Pandemics: spend on surveillance, not prediction. *Nature*, 558(7709):180–182.

[Ince et al., 2014] Ince, Y., Yasa, C., Metin, M., Sonmez, M., Meram, E., Benkli, B., and Ergonul, O. (2014). Crimean–Congo hemorrhagic fever infections reported by ProMED. *International Journal of Infectious Diseases*, 26:44–46.

[Jones et al., 2008] Jones, K. E., Patel, N. G., Levy, M. A., Storeygard, A., Balk, D., Gittleman, J. L., and Daszak, P. (2008). Global trends in emerging infectious diseases. *Nature*, 451(7181):990–993.

[Kane et al., 2014] Kane, M. J., Price, N., Scotch, M., and Rabinowitz, P. (2014). Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics*, 15(1):276.

[Kilpatrick and Randolph, 2012] Kilpatrick, A. M. and Randolph, S. E. (2012). Drivers, dynamics, and control of emerging vector-borne zoonotic diseases. *Lancet*, 380(9857):1946–1955.

[Le et al., 2013] Le, Q., Sarlós, T., and Smola, A. (2013). Fastfood-computing hilbert space expansions in loglinear time. In *International Conference on Machine Learning*, pages 244–252.

[Liaw et al., 2002] Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomForest. *R news*, 2(3):18–22.

[Luttinen and Ilin, 2012] Luttinen, J. and Ilin, A. (2012). Efficient Gaussian process inference for short-scale spatio-temporal modeling. In *Artificial Intelligence and Statistics*, pages 741–750.

[Messina et al., 2016] Messina, J. P., Kraemer, M. U., Brady, O. J., Pigott, D. M., Shearer, F. M., Weiss, D. J., Golding, N., Ruktanonchai, C. W., Gething, P. W., Cohn, E., et al. (2016). Mapping lobal environmental suitability for Zika virus. *Elife*, 5:e15272.

[Mostafavi et al., 2013] Mostafavi, E., Haghdoost, A., Khakifirouz, S., and Chinikar, S. (2013). Spatial analysis of Crimean–Congo hemorrhagic fever in Iran. *The American Journal of Tropical Medicine and Hygiene*, 89(6):1135–1141.

[Neal, 2012] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

[Nguyen et al., 2017] Nguyen, L., Hu, G., and Spanos, C. J. (2017). Spatio-temporal environmental monitoring for smart buildings. In *Proc. 13th IEEE Int. Conf. Control Automation (ICCA)*, pages 277–282.

[R Core Team et al., 2013] R Core Team et al. (2013). R: A language and environment for statistical computing.

[Randolph and Ergönül, 2008] Randolph, S. and Ergönül, Ö. (2008). Crimean–Congo hemorrhagic fever: Exceptional epidemic of viral hemorrhagic fever in Turkey. *Future Virology*, 3(4):303–306.

[Ridgeway et al., 2006] Ridgeway, G. et al. (2006). gbm: Generalized boosted regression models. *R package version*, 1(3):55.

[Riihimäki and Vehtari, 2014] Riihimäki, J. and Vehtari, A. (2014). Laplace approximation for logistic Gaussian process density estimation and regression. *Bayesian Analysis*, 9(2):425–447.

[Rogers and Randolph, 2006] Rogers, D. J. and Randolph, S. E. (2006). Climate change and vector-borne diseases. *Advances in Parasitology*, 62:345–381.

[Saatçi, 2012] Saatçi, Y. (2012). *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge.

[Särkkä and Hartikainen, 2012] Särkkä, S. and Hartikainen, J. (2012). Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In *International Conference on Artificial Intelligence and Statistics*, pages 993–1001.

[Senanayake et al., 2016] Senanayake, R., O'Callaghan, S., and Ramos, F. (2016). Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression. In *13th AAAI Conference on Artificial Intelligence*, pages 3901–3907.

[Stegle et al., 2011] Stegle, O., Lippert, C., Mooij, J. M., Lawrence, N. D., and Borgwardt, K. (2011). Efficient inference in matrix-variate Gaussian models with\iid observation noise. In *Advances in Neural Information Processing Systems*, pages 630–638.

[Tobler, 1970] Tobler, W. R. (1970). A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46(sup1):234–240.

[Vanhatalo et al., 2010] Vanhatalo, J., Pietiläinen, V., and Vehtari, A. (2010). Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine*, 29(15):1580–1607.

[Varadhan, 2017] Varadhan, R. (2017). Alabama: Constrained nonlinear optimization, R package version 2015.3-1, 2015. *URL https://CRAN. R-project. org/package= alabama. Accessed on June*.

[Vescio et al., 2012] Vescio, F. M., Busani, L., Mughini-Gras, L., Khoury, C., Avellis, L., Taseva, E., Rezza, G., and Christova, I. (2012). Environmental correlates of Crimean–Congo haemorrhagic fever incidence in Bulgaria. *BMC Public Health*, 12:1116.

[Wang and Chaib-Draa, 2013] Wang, Y. and Chaib-Draa, B. (2013). A knn based Kalman filter Gaussian process regression. In *23rd International Joint Conference on Artificial Intelligence*, pages 1771–1777.

[Washington et al., 2015] Washington, M. L., Meltzer, M. L., for Disease Control, C., and (CDC), P. (2015). Effectiveness of Ebola treatment units and community care centers - Liberia, September 23-October 31, 2014. *Morbidity and Mortality Weekly Report (MMWR)*, 64(3):67–69.

[Williams and Rasmussen, 2006] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT Press Cambridge, MA.

[Wilson et al., 2014] Wilson, A. G., Gilboa, E., Nehorai, A., and Cunningham, J. P. (2014). Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, pages 3626–3634.

[World Health Organization, 2014] World Health Organization (2014). World malaria report 2014. Geneva.