Machine learning in space and time

Spatiotemporal learning and inference with Gaussian processes and kernel methods

Seth R. Flaxman

August 2015

School of Computer Science
Machine Learning Department

School of Public Policy and Management
H. John Heinz III College

Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

Thesis Committee:

Daniel Neill, co-chair
Alex Smola, co-chair
Cosma Shalizi
Andrew Gelman (Columbia University)

Acknowledgements

This dissertation is dedicated to Jaclyn: thank you for setting us on a journey both literal (Lausanne, Geneva, Jerusalem, New York, Oxford) and intellectual, and always expecting something humanistic (or at least, social scientific) from a computer scientist.

Thanks to my committee, Daniel Neill, Alex Smola, Cosma Shalizi, and Andrew Gelman for your high expectations and insights, and always asking the important questions. My deep and abiding gratitude to Daniel for the initial suggestions that led to the questions that consume my research, for always keeping my methodological explorations and research agenda grounded, and for being so generous and punctual with his careful and honest critiques of my work. A heartfelt thanks to Alex for throwing dozens (if not hundreds) of ideas at me, and allowing me to do the same to him. Alex's faith in me and my abilities was unflagging, constantly motivating me to explore new research directions. Some day I'll write it at all up.

Thanks to my collaborators, Andrew Gelman, Sharad Goel, William Herlands, Karim Kassam, Charles Loeffler, Daniel Neill, Hannes Nickisch, Justin Rao, Alex Smola, Aki Vehtari, Yu-Xiang Wang, and Andrew Wilson.

Thanks to all the CMU professors who were always generous with their time and advice, especially Al Blumstein, Jon Caulkins, Dave Choi, Alex Chouldechova, Steve Fienberg, Clark Glymour, Geoff Gordon, Wil Gorr, Karim Kassam, Brian Kovak, David Krackhardt, Tom Mitchell, Jared Murray, Daniel Nagin, Rema Padman, Barnabás Póczos, Noah Smith, Cosma Shalizi, Larry Wasserman, Mark Schervish, Peter Spirtes, Roni Rosenfeld, Rebecca Steorts, Andrew Thomas, and Ryan Tibshirani. Thanks to Tom Mitchell for creating MLD and Ramayya Krishnan, Marty Gaynor, and Rahul Telang for steering Heinz. An enormous thanks to the two indefatigable people who keep the two programs afloat, Diane Stidle and Gretchen Hunter. Thanks to Robert Sampson at Harvard and thanks to the Columbia professors who hosted me and exposed me to new perspectives while I completed this work: Dave Blei, John Cunningham, and Andrew Gelman.

Thank you to my peers. First to my fellow Event and Pattern Detection laboratory members, especially those who paved the way, Edward McFowland, Sriram Somanchi, and Skyler Speakman. Second to my stats study group, Brian Kent, Erich Kummerfeld, Sriram Somanchi, Amy Wesolowski, and Ruben and my machine learning study group of one, Jesse Dunietz.

Thanks to machine learners / statisticians Allison Chaney, Ben Eckart, Wei Dai, Georg Goerg, Adona Iosif, Alp Kucukelbir, Yifei Ma, Susanna Makela, Avner May, James McInerney, Shira Mitchell, Louis Mittel, Brendan O'Connor, Aaditya Ramdas, Rajesh Ranganath, Benjamin Reddy, Wei Wang, Yu-Xiang Wang, and Leila Wehbe for being enthusiastic about public policy, and to public policyniks Maria de Arteaga, Lindsay Bostwick, Matt Crespi, Maria Cuellar, Ciprian Domnisoru, Matthew Eisenberg, Thomas Goldring, Sonam Samat, Nathan VanHoudnos, Quan Wang, Hilary Wolfendale, Yiye Zhang, Zhe Zhang, and Melanie Zilora for being enthusiastic about machine learning. Thanks to Alex Bick, Ofer Eckstein, Ezra Furman, Miriam Hinman, Emily Kaplan, Hilary Finucane, Sam Garfinkel, Noah Hertz-Bunzl, Ben Milder, Jenna Mellor, Anna Ratoff, Yakir Reshef, Zach Rudin, Jill Swencionis, Jean Yang, Tom Wooten, and Luba Zeifman for always being friends, and often caring about my research. *A sheynem un hartsikn dank* to the CMU klezmer band and the Yiddish club, especially to the Davids, and to my housemates in the Bronx who were very understanding while Jaclyn and I both completed our PhDs.

Thanks to the teachers, professors, researchers, supervisors, co-workers, and mentors who inspired me to start this PhD: John Benson and Mark Vondracek at ETHS, David Kelly at Hampshire, Ran Libeskind-Hadas at Harvey Mudd College, Marshall Ganz, Harry Lewis, Michael Mitzenmacher, Michael Rabin, and Salil Vadhan at Harvard, Alex Samorodnitsky at Hebrew University, Jeffrey Huang, Bernard Moret, Amin Shokrollahi, and Xavier Comtesse at EPFL, and Majid Ezzati, Mariel Finucane, Abraham Flaxman, Maya Mascarenhas, Colin Mathers, Chris Paciorek, and Gretchen Stevens on the Global Burden of Disease project.

And finally, thanks to my family, to my in-laws, and Ruben for all of their support in myriad ways. To Deborah, whose imminent arrival spurred on the completion of this work. To my father and brothers for nurturing my deep interest in computers and mathematics from a young age, to my mother for sparking my curiosity about social science and the experimental method, to my brother Abie for opening doors to me in computer science and public health, and to both of my brothers for setting a high bar to follow.

Abstract

In this thesis, I present novel statistical machine learning methods for answering public policy-motivated questions about spatiotemporal datasets. Gaussian processes provide a coherent Bayesian framework for analyzing spatiotemporal data, while kernel methods have deep roots in spatial statistics and have more recently given rise to a variety of fresh perspectives on classical statistical questions. Both have been quite successful and popular in machine learning and beyond, yet run-time and storage complexity have been a limiting factor in their widespread adoption. I present new approaches combining Gaussian processes and recent advances in kernel methods, with a focus on scalable Bayesian inference to answer scientifically relevant questions. These questions, drawn from the domain of public policy, include: how to define valid measures of association between variables observed in space and time, how to create accurate small area spatiotemporal forecasts which adequately reflect the uncertainty in these forecasts, how to make causal inference in the presence of spatiotemporal structure, and how to draw conclusions about individuals from aggregate-level data.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The purpose of this thesis is to develop new statistical machine learning methods for spatiotemporal data. My goals are learning and scalable inference focused on drawing scientifically relevant conclusions from spatiotemporal observational datasets.

Tom Mitchell's by now classic definition of machine learning says, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell, 1997). For me, *statistical* machine learning with a focus on valid scientific conclusions draws on a view of the Bayesian statistical approach which emphasizes an iterative approach to model checking, refinement, and continuous model expansion (Gelman and Shalizi, 2013). Drawing inspiration also from Breiman's classic "two paradigms" dichotomy between statistics and machine learning (2001), I give the following provisional definition:

Statistical machine learning is a set of methods (probability models, inference techniques, posterior checks) M, which when used appropriately by scientists, enables learning from experience E with respect to scientific tasks T and performance measure P, if the performance of the methods M on tasks T as measured by P enables the practitioners to derive new, statistically characterized, scientific knowledge about experience E and to refine M.

Or, put more succinctly, models exist to tell us where they fail, and thus beget new models[1].

In this thesis I explore these issues in the context of observational spatiotemporal datasets and the attendant literature on time series and spatial statistics. Unlike classical statistics (whether Bayesian or frequentist), with spatiotemporal data the basic assumption of independent and identically distributed (iid) observations is immediately violated.

[1] I attribute this idea to Andrew Gelman, who traces it to Lakatos (1978). Lakatos writes, "Now, Newton's theory of gravitation, Einstein's relativity theory, quantum mechanics, Marxism, Freudianism, are all research programmes ... Each of them, at any stage of its development, has unsolved problems and undigested anomalies. All theories, in this sense, are born refuted and die refuted."

This thesis focuses on space-time processes indexed by spatial locations $\boldsymbol{s} \in D \subset \mathcal{R}^d$ and temporal labels $t \in T \subset \mathcal{R}$ (see, e.g. Cressie (1993)):

$$\{Z(\boldsymbol{s},t) : \boldsymbol{s} \in D, t \in T\} \tag{1.1}$$

A space-time process is a stochastic process, that is, a family of random variables defined on a probability space $(\Omega, \mathcal{F}, P)$. Following Brockwell and Davis (2013), notice that for a fixed space-time location $(\boldsymbol{s},t)$, $Z(\boldsymbol{s},t)$ is a random variable, with all the usual statistical properties of random variables. Formally we have that for fixed $(\boldsymbol{s},t)$, $Z(\boldsymbol{s},t)(\cdot)$ is a function on $\omega \in \Omega$. If we are able to repeatedly draw samples at $(\boldsymbol{s},t)$ then we have returned to a classical iid statistical framework. But as we will see below, we often only have a single observation at $(\boldsymbol{s},t)$, and worse, it is the fundamental nature of spatiotemporal data that observations at nearby locations in space and time are similar, thus violating the classical statistical assumption of independence. This is the first major difficulty of spatiotemporal statistics.

If we instead fix ω, then $Z(\cdot,\cdot)(\omega)$ is a function over space and time which we call the realizations or sample paths of the process $\{Z(\boldsymbol{s},t), s \in D, t \in T\}$. Just as a single observation is a draw from a probability distribution, a set of spatiotemporal observations, indexed by space and time, are a realization of a spatiotemporal process. But herein lies the second major difficulty of spatiotemporal statistics: while in classical statistics we would obtain multiple, identically distributed observations and use these to perform inference, in spatiotemporal statistics a single set of spatiotemporal observations is the only realization we have of the process. Our sample size is only $n = 1$!

Spatiotemporal datasets immediately challenge the hope that big data will obviate the work of careful statistical modeling. Even in settings where datasets are seemingly exhaustive—with complete coverage of all the users in a social network or hospital, for example—we see that asking scientific questions like, “what will happen tomorrow?” (forecasting), “do these patterns hold beyond this setting?” (generalizability), and “what will happen if we change something about the system?” (causal inference) shows that there is much that we have not and cannot observe. Modeling these questions, estimating our answers to these questions, and characterizing the uncertainty in these estimates is the work of statistics, and statistical machine learning has much to offer in the service of these goals. In this thesis, I focus on deriving valid scientific conclusions from spatiotemporal datasets. The scientific conclusions I have in mind include:

- valid measures of association between variables observed in space and time
- accurate small area spatiotemporal forecasting with valid uncertainty intervals

- causal inference, either exploiting or adjusting for spatiotemporal structure
- ecological inference, that is, drawing conclusions about individuals from aggregate-level data

In each case, the presence of spatiotemporal structure in the data both motivates and complicates the scientific question. For example, measures of association are biased by the presence of temporal confounding, but time structure allows for accurate spatiotemporal forecasting. Ecological inference relies on variation at the level of the ecological unit, but the spatial structure is usually ignored. Causal inference with observational data is a particularly interesting case because it is hard for the same basic reason that spatiotemporal statistics is hard, namely that the iid assumption does not hold, meaning that the data cannot be analyzed as a random sample.

In addition to drawing on two manuscripts which are in preparation (Flaxman et al., 2013, 2015a), this thesis contains work from three publications:

- Flaxman, Neill, and Smola, "Gaussian processes for independence tests with non-iid data in causal inference," ACM Transactions on Intelligent Systems and Technology (TIST), 2015b.
- Flaxman, Wilson, Neill, Nickisch, and Smola, "Fast Kronecker inference in Gaussian Processes with non-Gaussian likelihoods," International Conference on Machine Learning (ICML), 2015d.
- Flaxman, Wang, and Smola, "Ecological inference with distribution regression," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2015c.

My thesis is structured as follows. Chapter 2 provides a background on kernel methods and a catalog of useful kernels for spatiotemporal data, illustrated with Gaussian processes. Chapter 3 takes a detailed look at the problems with classical space/time interaction tests (Knox, Mantel) which are revealed by my new kernelized version of these tests. Chapter 4 describes Gaussian processes and argues for their use as a general purpose regression method for dealing with spatiotemporal data, with a specific focus on scalable Bayesian inference, highlighting my recent work which exploits structure in the kernel (covariance function) of the GP (Flaxman et al., 2015b). Chapter 5 presents my new ecological inference through distribution regression method, combining explicit feature space expansions of kernels for scalability with GP logistic regression for inference (Flaxman et al., 2015c). Chapter 6 presents my new conditional independence test (Flaxman et al., 2015b), especially useful for algorithmic causal inference. Chapter 7 revisits the GP models of Chapter 4 to highlight the potential for a fully Bayesian approach to inference and hyperparameter learning through a hierarchical specification (Flaxman et al., 2015a). Links to source code repositories, demos, and tutorials can be found online at www.sethrf.com.

Chapter 2

Kernel methods for spatiotemporal data

In this chapter I introduce kernel methods, with a focus on their use for learning with spatiotemporal data. Tobler's first law of geography, "Everything is related to everything else, but near things are more related than distant things" (1970) provides the intuition for the use of kernels in spatiotemporal learning. If we are to take Tobler's law as motivating our modeling assumptions, then we need a mathematical model quantifying the extent to which things are related to one another over space and time. Kernels provide that model. The exponential kernel, for example, has the following form:

$$k(t_1,t_2) = e^{-|t_1-t_2|} \tag{2.1}$$

Noting that this kernel is stationary (i.e. $k(t_1+h,t_2+h) = k(t_1,t_2)$ for any shift h) we visualize $k(u)$ where $u = t_1 - t_2$ in Figure 2.1. We can understand k as a function characterizing Tobler's law, where we imagine that t_1 and t_2 are spatial (or temporal) locations and $k(\cdot,\cdot)$ quantifies the strength of the relationship between these two locations. As shown in Figure 2.1, everything is indeed related to everything else, and moreover, near things are more related than distant things.

2.1 Reproducing kernel Hilbert space

We have the following definition (Schölkopf and Smola, 2002; Wahba, 1990):

Definition 1 *Given a set $\mathcal{T}$, a positive semidefinite kernel k is a symmetric bivariate function with the property that for any real $a_1,\ldots,a_n$ and $t_1,\ldots,t_n \in \mathcal{T}$:*

$$\sum_{i,j=1}^{n} a_i a_j k(t_i,t_j) \geq 0 \tag{2.2}$$

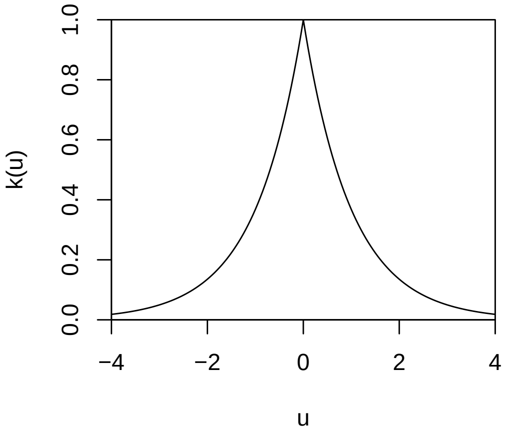


Fig. 2.1 Plot of the exponential kernel $k(t_1,t_2) = k(u) = e^{-u}$.

Why is it so important that k be positive semidefinite? Following Wahba (1990), consider the case that $\mathcal{T} = (1,2,\ldots,N)$. Construct the $N \times N$ Gram matrix K where $K_{ij} = k(i,j)$ for $i,j \in \mathcal{T}$. We can now rewrite Eq. (2.2) in matrix form as:

$$\boldsymbol{a}^\top K \boldsymbol{a} \geq 0, \ \ \forall \boldsymbol{a} \in \mathcal{R}^n \tag{2.3}$$

We recognize this constraint as saying that K is positive semidefinite, and as a result it is a valid covariance matrix, meaning that the powerful machinery of classical statistics is now available to us. (If we treat K as the covariance matrix for a Gaussian distribution, we have Gaussian processes, as explained in subsequent chapters.) Let us consider the eigendecomposition of K, writing $K = Q\Lambda Q^\top$ with Q orthogonal and Λ diagonal. Since K is positive semidefinite, the entries of Λ are non-negative, so we can write: $K = Q\Lambda^{1/2}\Lambda^{1/2}Q^\top = Q\Lambda^{1/2}(Q\Lambda^{1/2})^\top = \Phi\Phi^\top$ for $\Phi := Q\Lambda^{1/2}$. Denoting the ith column of Φ as Φ_i we have:

$$k(i,j) = \Phi_i^\top \Phi_j = \langle \Phi_i, \Phi_j \rangle \tag{2.4}$$

Notice that Φ_i represents an element $i \in \mathcal{T}$ by a vector of features (a so-called "feature space representation"), such that inner products between vectors Φ_i and Φ_j exactly correspond to our kernel k.

Returning to the case of general sets $\mathcal{T}$, we can generalize this feature space representation by defining a Hilbert space $\mathcal{H}$ using k where members of the Hilbert space are functions

(Schölkopf and Smola, 2002):

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(t_i, \cdot) \tag{2.5}$$

for $m \in \mathcal{N}$, $\alpha_i \in \mathcal{R}$ and $t_i \in \mathcal{T}$. If $g(\cdot) = \sum_j \beta_j k(t_j, \cdot) \in \mathcal{H}$ we define dot products as:

$$\langle f, g \rangle := \sum_i \sum_j \alpha_i \beta_j k(x_i, x_j) \tag{2.6}$$

Eq. (2.6) immediately implies the following properties:

$$\langle k(\cdot, t), f \rangle = f(x) \tag{2.7}$$

$$\langle k(t_i, \cdot), k(t_j, \cdot) \rangle = k(t_i, t_j) \tag{2.8}$$

The second property explains the term “reproducing” and we have the following definition (Schölkopf and Smola, 2002):

Definition 2 *For a set $\mathcal{T}$ and a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{T} \to \mathcal{R}$, $\mathcal{H}$ is a reproducing kernel Hilbert space if there exists a kernel k with the reproducing property of Eq.* (2.8) *and with the property that $\mathcal{H} = \overline{span\{k(t, \cdot) | t \in \mathcal{T}\}}$ where $\overline{A}$ denotes the completion of the set A.*

Just as we eigendecomposed K above, Mercer’s theorem tells us that if $\int \int k^2(s,t) ds dt < \infty$ then we have:

$$k(s,t) = \sum_i \lambda_i \Phi_i(s) \Phi_i(t) \tag{2.9}$$

This gives us a feature space representation:

$$\Phi : t \to (\sqrt{\lambda_i} \Phi_i(t))_i \tag{2.10}$$

because we have:

$$\langle \Phi(s), \Phi(t) \rangle = \langle (\sqrt{\lambda_i} \Phi_i(s))_i, (\sqrt{\lambda_i} \Phi_i(t)) \rangle \tag{2.11}$$

$$= \sum_i \lambda_i \Phi_i(s) \Phi_i(t) \tag{2.12}$$

$$= k(s,t) \tag{2.13}$$

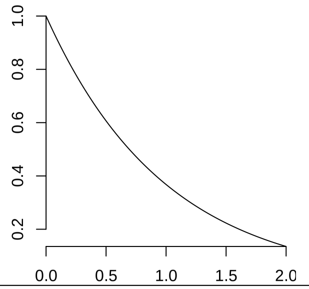
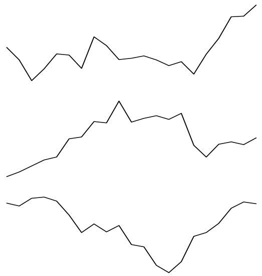
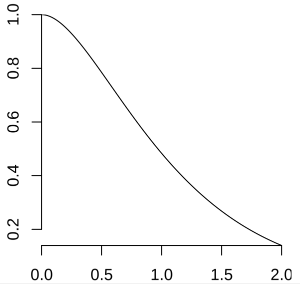
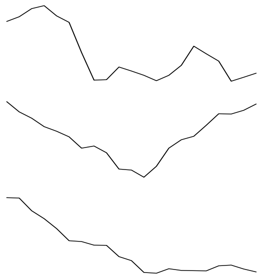
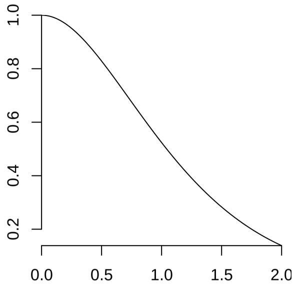
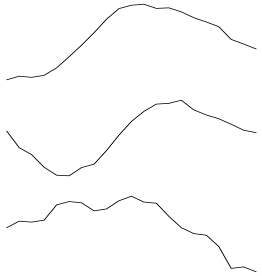
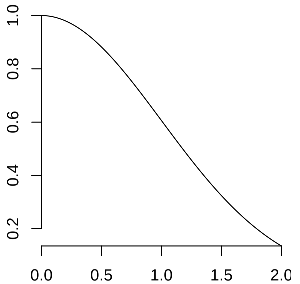
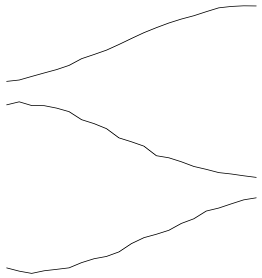
This feature space representation is called the Mercer map, and it is in general not the same as the RKHS mapping $\phi(t) = k(t, \cdot)$. But since both correspond to the same kernel $k(\cdot, \cdot)$ and this is usually what we ultimately care about, we identify reproducing kernel Hilbert spaces with the same kernel. We will return to the Mercer map representation in the next chapter on Gaussian processes.

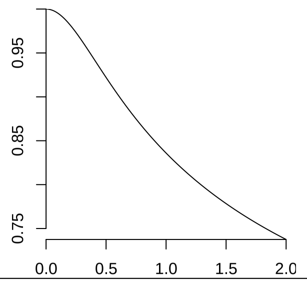
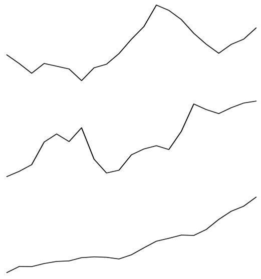
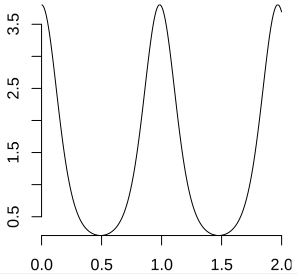
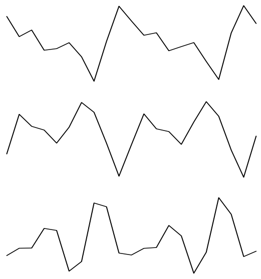
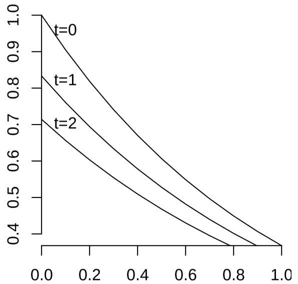
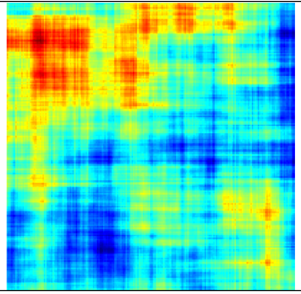
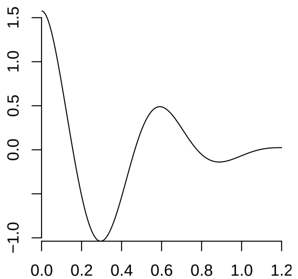
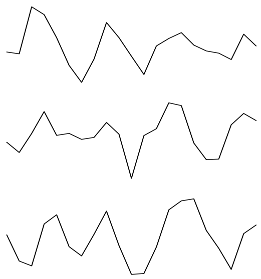
Below, I catalog useful kernels for spatiotemporal data, and give an overview of kernel embeddings of probability distributions.

2.2 Kernels for spatiotemporal data

In this section I give an overview of some of the kernel choices which I have found to be useful for spatiotemporal modeling. For an overview of useful kernels in machine learning more generally, see Rasmussen and Williams (2006); Schölkopf and Smola (2002); Souza (2010). For a very useful catalog explaining how to build complex kernels out of simpler ones (e.g. addition and multiplication) see Genton (2002) and for work on automating this process see Duvenaud et al. (2013).

For each kernel, I have plotted the covariance function with default hyperparameters for illustration (e.g. signal variance 1 and length-scale 1), along with sample paths drawn from a GP with covariance function given by the kernel. I have omitted examples of non-stationary kernels, but see Paciorek and Schervish (2006) and references therein.

Kernel	Covariance plot	Sample paths	Description
Exponential (also known as Laplace, Matérn-1/2) $k(d) = \sigma^2 \exp\left(-\frac{d}{\ell}\right)$			Gives rough sample paths. Covariance for an Ornstein-Uhlenbeck process, the continuous time generalization of an autoregressive process
Matérn-$\frac{3}{2}$ $k(d) = \sigma^2 \left(1 + \sqrt{3}d\ell\right) \exp\left(-\frac{\sqrt{3}d}{\ell}\right)$			Gives smoother sample paths than exponential.
Matérn-$\frac{5}{2}$ $k(d) = \sigma^2 \left(1 + \sqrt{5}\lvert d\rvert\ell + \frac{5\lvert d\rvert^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\lvert d\rvert}{\ell}\right)$			Gives smoother sample paths than Matérn-3/2.
Squared exponential (also known as RBF) $k(d) = \sigma^2 \exp\left(-\frac{d^2}{\lambda^2}\right)$			Very smooth. Equivalent to Matern-ν for $\nu \to \infty$.

Kernel	Covariance plot	Sample paths	Description
Rational Quadratic (plotted with $\alpha = .1$) $k(d) = \sigma^2 \left(1 + \frac{d^2}{\alpha \ell^2}\right)^{-\alpha}$			Additive mixture of SE kernels with different lengthscales.
Periodic $k(d) = \sigma^2 \exp\left(-2\sin^2(\frac{d\pi}{p\ell^2})\right)$			Period is given by p.
Gneiting $k(d,t) = \frac{\sigma^2}{a\lvert t\rvert+1} \exp\left(-\frac{d}{(a\lvert t\rvert+1)^{\beta/2}}\right)$ [see Gneiting (2002) for other possibilities]			$\beta \in [0,1]$ controls the degree of space-time interaction. $\beta = 0$ implies fully separable, $\beta = 1$ implies nonseparable.
Spectral Mixture (Wilson and Adams, 2013b) $k(d) = \sum_{q=1}^{Q} w_q \exp(-2\pi^2 d^2 v_q)\cos(2\pi d \mu_q)$			Gaussian mixture model in the spectral domain. Can reproduce any stationary covariance and learn different periodicities.

2.3 Separable kernels for spatiotemporal data

In the case of spatiotemporal data where space locations are indexed by s (e.g. (x,y) state-plane coordinates or latitude/longitude) and time labels are indexed by t, a natural way to build a space/time kernel is to multiply a spatial and temporal kernel (additive models are considered in Chapter 7):

$$k((s,t),(s',t')) = k_s(s,s')k_t(t,t') \tag{2.14}$$

This specification is referred to as a “separable” kernel in the literature. It has computational benefits for scaling up Gaussian process models, based on Kronecker algebra, which we exploit in Chapters 4 and 7. The results that we need are below.

Note first that to calculate the Gram matrix K for Eq. (2.14), we calculate the smaller Gram matrices K_s and K_t and then use the Kronecker product: $K = K_s \otimes K_t$. This gives a hint of the efficiency gains we can exploit. For Kronecker matrix vector multiplication, we use the identity Steeb and Hardy (2011):

$$(B^\top \otimes A)v = \mathrm{vec}(AVB) \tag{2.15}$$

where $v = \mathrm{vec}(V)$ and the vec operator turns a matrix into a vector by stacking columns vertically. Since a full $n \times n$ matrix is never formed, this approach is very efficient in terms of space and time complexity, relying only on operations with the smaller matrices K_i and the matrix V which only has n entries. Matrix-vector multiplication $(\bigotimes K_d)\, v$ reduces to D matrix-matrix multiplications VK_j where V is a matrix with n entries total, reshaped to be $n^{\frac{D-1}{D}} \times n^{\frac{1}{D}}$. This matrix-matrix multiplication is $\mathcal{O}(n^{\frac{D-1}{D}} n^{\frac{1}{D}} n^{\frac{1}{D}}) = \mathcal{O}(n^{\frac{D+1}{D}})$ so the total run-time is $\mathcal{O}(Dn^{\frac{D+1}{D}})$.

Another result we use is that given the eigendecompositions of $K_d = Q_d \Lambda_d Q_d^T$, we have:

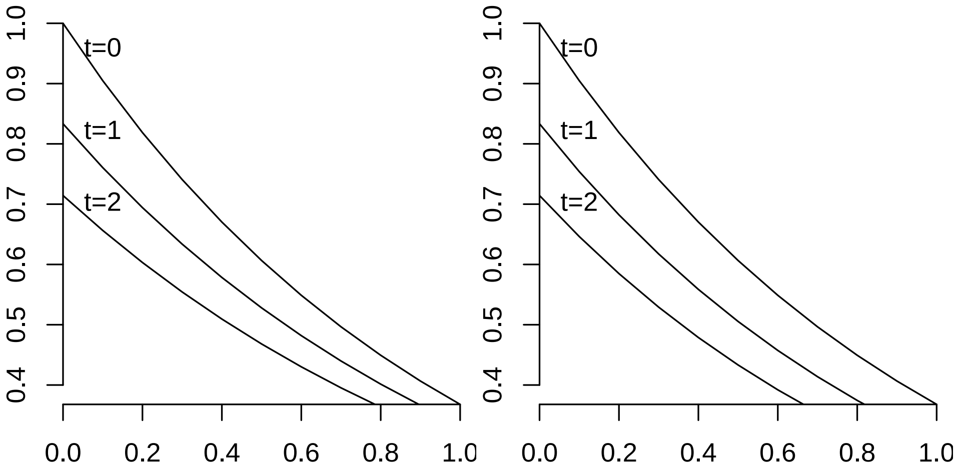
$$K = (\bigotimes Q_d)(\bigotimes \Lambda_d)(\bigotimes Q_d^T) \tag{2.16}$$

A final, useful result concerns determinants:

$$\det(K_1 \otimes K_2) = \det(K_1)^m \det(K_2)^n \tag{2.17}$$

where K_1 is $n \times n$ and K_2 is $m \times m$.

Despite these attractive computational features, it is worth asking whether the assumption of separability applies in practice. This question, and the search for nonseparable kernels, has attracted attention in the literature (e.g. Cressie and Huang (1999); Gneiting (2002); Gneiting et al. (2007); Mitchell et al. (2005)). Consider the two kernels visualized in Figure 2.2, a nonseparable version of one of Gneiting’s covariances on the left ($\beta = 1$) and the corresponding separable version ($\beta = 0$) on the right. Separability implies, e.g. that the temporal autocorrelation structure for two locations a distance 1 unit apart is exactly proportional to the temporal autocorrelation structure for two locations a distance 5 units apart. This assumption is likely to be violated with real data, but whether or not this matters in practice is a separate question. In the case of GPs, the kernel parameterizes the prior function class, but the observations, especially if they are abundant, can outweight the prior. Thus, the assumption of separability is not usually an overly restrictive assumption, unless space/time interaction is explicitly what is being studied, as in Chapter 3.



(a) Nonseparable Gneiting covariance ($\beta = 1$) (b) Separable Gneiting covariance ($\beta = 0$)

Fig. 2.2 Gneiting (2002) proposed classes of non-separable covariance functions. Plotted here is: $k(d,t) = \frac{\sigma^2}{a|t|+1} \exp\left(-\frac{d}{(a|t|+1)^{\beta/2}}\right)$ with the nonseparable $\beta = 1$ case on left and the separable $\beta = 0$ case on the right. While the difference can be quite subtle visually, notice that in the separable case on right, the lines are exactly proportional to each other.

2.4 Kernel embeddings of probability distributions

In this section, I overview work that has emerged in the machine learning literature in the last two decades on applying kernel methods to classical statistical problems. Early examples include kernel PCA (Schölkopf et al., 1997), kernel ICA (Bach and Jordan, 2003), and kernel dimensionality reduction (Fukumizu et al., 2004). Going beyond merely applying the "kernel trick" to create non-linear versions of existing methods, these approaches have statistical guarantees and a mathematical foundation based on RKHS theory, especially the use of covariance operators. Below, I explain the use of covariance operators in one of the most popular parts of this large literature, kernel methods for embedding probability distributions, illustrated by testing for statistical independence.

Let us start by defining the kernel mean embedding operator. The intuition is that we wish to extend the feature space representation of kernels introduced earlier from embedding single elements to embedding probability distributions. Given a random variable $x \sim P$ with feature map $\phi(x)$ which projects x into an RKHS $\mathcal{H}$, we want to have a formal way of defining $\mathbf{E}_x[\phi(x)]$. The mean embedding operator is that formalism.

Define $\mu_P \in \mathcal{H}$ to have the following property:

$$\mathbf{E}_x[f] = \langle f, \mu_P \rangle_{\mathcal{H}} \tag{2.18}$$

Following the presentation in Gretton et al. (2012), we have the following lemma for the existence of μ_P:

Lemma 3 *If $k(\cdot,\cdot)$ is measurable and $\mathbf{E}_x\left[\sqrt{k(x,x)}\right] < \infty$ then $\mu_P \in \mathcal{H}$. Moreover, $\mu_P = \mathbf{E}_x\left[\phi(x)\right]$.*

Proof We will rely on the Riesz representation theorem [see references in Gretton et al. (2012)] to prove the existence of μ_P. First we define the linear operator $T_p f = \mathbf{E}_x f$ and see that it is bounded for all f because:

$$|T_p f| = |\mathbf{E}_x f(x)| \leq E_x|f(x)| = \mathbf{E}_x|\langle f, \phi(x)\rangle_{\mathcal{H}}| \leq \mathbf{E}_x\left[\sqrt{k(x,x)}\|f\|_{\mathcal{H}}\right] < \infty \tag{2.19}$$

Since T_p is a bounded linear operator, there must be an element in $\mathcal{H}$ which we denote μ_P with the property:

$$T_p f = \langle f, \mu_P \rangle_{\mathcal{H}} \tag{2.20}$$

This completes the proof of the first claim. To see why $\mu_P = \mathbf{E}_x[\phi(x)]$ we consider:

$$\mu_P(t) = \langle \mu_P, \phi(t) \rangle = \mathbf{E}_x\left[\phi(t)(x)\right] \tag{2.21}$$

where the first step is by the Riesz representation theorem (recalling that $k(t,\cdot) = \phi(t)$) and the second step uses the result we just established. Finally, we treat t as an arbitrary input and use $\phi(t)(x) = \phi(x)(t)$ to obtain:

$$\mu_P = \mathbf{E}_x \phi(x) \tag{2.22}$$

■

Mean embeddings are a very powerful tool as they give us a nonparametric representation of a distribution as a (possibly infinite dimensional) vector, which we can use in any number of statistical methods. In Chapter 5 we will use the finite dimensional representation introduced in the next section to approximate kernel mean embeddings for use in a regression setting.

For kernels which are universal and / or characteristic (Sriperumbudur et al., 2010), such as the RBF kernel, the mean embedding operator is injective, so if we have samples from two distributions $x \sim P$ and $y \sim Q$ then $\mu_P = \mu_Q \iff P = Q$. This is the basis for the Maximum Mean Discrepancy (MMD) test statistic (Gretton et al., 2012), which measures the difference $||\mu_P - \mu_Q||$.

2.5 Testing for independence with kernels

We now turn to a related test statistic, the Hilbert-Schmidt Independence Criterion. Given observations from a joint distribution $P(X,Y)$, the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2008, 2005) is a statistical test for the null hypothesis of independence: $X \perp\!\!\!\perp Y$. HSIC uses mean embeddings to compare the joint distribution to the product of the marginals (which is an equivalent statement to $X \perp\!\!\!\perp Y$). Let A represent the joint distribution $p_{X,Y}$ and let B represent the product of the marginal distributions $p_X p_Y$. Then we wish to test whether $\mu_A = \mu_B$. Usually a separable kernel is used to define the Hilbert space $\mathcal{H}_{X,Y} = \mathcal{H}_X \otimes \mathcal{H}_Y$, i.e. given kernels k for $\mathcal{H}_X$ and ℓ for $\mathcal{H}_Y$, the kernel u for the product space is given by:

$$u((x,y),(x',y')) = k(x,x')\ell(y,y') \tag{2.23}$$

Thus using mean embeddings, we have $\mu_A = \mu_{PQ}$ and $\mu_B = \mu_P \otimes \mu_Q$.

We consider the following test statistic, which is equivalent to the square of MMD:

$$\text{HSIC} = ||\mu_{PQ} - \mu_P \otimes \mu_Q||^2 = \left|\left|\mathbf{E}_{x,y}\left[k(x,\cdot)\ell(y,\cdot)\right] - \mathbf{E}_x\left[k(x,\cdot)\mathbf{E}_y \ell(y,\cdot)\right]\right|\right|^2 \tag{2.24}$$

By analogy to the mean embedding operator, and using similar arguments, we can define the covariance operator Σ_{PQ}:

$$\langle f, \Sigma_{PQ} g\rangle = \text{Cov}(f(x), g(y)) = E[f(x)g(y)] - E[f(x)]E[g(y)] = \mu_{PQ} - \mu_P \otimes \mu_Q \tag{2.25}$$

Thus HSIC can equivalently be written as $||\Sigma_{PQ}||^2$.

An estimator can be immediately derived:

$$\widehat{\text{HSIC}} = \frac{1}{n^2}\sum_{i,j} k(x_i,x_j)\ell(y_i,y_j) - \frac{2}{n^3}\sum_{i,j,q} k(x_i,x_j),\ell(y_i,y_q) + \frac{1}{n^4}\sum_{i,j,q,r} k(x_i,x_j)\ell(y_q,y_r) \tag{2.26}$$

This estimator can be written compactly in terms of Gram matrices K and L:

$$\widehat{\text{HSIC}} = \frac{1}{n^2}\text{tr}(KHLH)$$

where $K_{ij} = k(x_i, x_j)$, $L_{ij} = \ell(y_i, y_j)$, and $H = I - \frac{1}{n}11^T$ is a centering matrix.

The distribution of HSIC under the null can be obtained by randomization testing: given pairs (x_i, y_i) we shuffle the y's and recompute $\widehat{\text{HSIC}}$. Gretton et al. (2008) gives an asymptotic result based on the Gamma distribution, and Zhang et al. (2011) gives a test based on the eigenvalues of the kernel matrices.

2.6 Randomized feature expansions

While feature expansions for most popular kernels (and all the ones considered in Section 2.2) are infinite dimensional, it is often possible to implement kernelized algorithms based on the covariance (Gram) matrix containing $k(x_i, x_j)$ for every pair of observations x_i and x_j. HSIC, for example, relies only on matrices K and L, and as we will see in later chapters, the necessary calculations for Gaussian processes are in terms of the covariance matrices. Nevertheless in the large data setting, calculating, storing, and manipulating $n \times n$ matrices can be computationally prohibitive.

In recent years, a number of approximate feature expansions have been proposed which find a d-dimensional approximation $\hat{\phi}(x) \in \mathbb{R}^d$ of $\phi(x)$ for every x such that:

$$k(x,y) = \langle \phi(x), \phi(y) \rangle \approx \langle \hat{\phi}(x), \hat{\phi}(y) \rangle \tag{2.27}$$

Rahimi and Recht (2007) proposed random Fourier features, an approximation for stationary kernels based on randomly sampling from the Fourier transform of the kernel function, relying on Bochner's theorem (also used in Section 4.8) which says that this spectral representation fully represents the kernel. Le et al. (2013) proposed FastFood, a more computationally efficient version of random Fourier features that we use in Chapter 5. Given an explicit d-dimensional feature expansion, most kernelized algorithms can be made much more efficient. Bach (2015) gives a comprehensive theoretical analysis of random features, making interesting connections with quadrature.

Chapter 3

Kernel space/time interaction tests

3.1 Introduction

Knox (1964), Mantel (1967), and Diggle et al. (1995) all developed important and widely used tests for space-time interaction with spatiotemporal point processes. While each of these statistical tests has different features, a fundamental limitation of each is the requirement that the user pre-specify a range of critical spatial and temporal distances of interest, i.e. a priori knowledge must be used to decide what distances are considered “close” versus “far”, in space and time. One of the motivating goals of this chapter is to relax this assumption while not sacrificing statistical power. I take a new look at the assumptions underlying these tests, showing how each can be understood as testing a particular null hypothesis, namely that the probability distributions over interpoint distances and interpoint time intervals are independent.

In this framework, I focus on the development of a set of new space-time interaction tests, based on the Hilbert Schmidt Independence Criterion (HSIC) (reviewed in Chapter 2) a kernel-based test statistic for testing for independence between probability distributions. While HSIC was originally proposed for independent, identically distributed (iid) data, I show how it can be used with spatiotemporal point patterns. This new perspective allows us to fix a problem with Mantel’s test and it also shows important failings, which have not been previously noted in the literature, with previously proposed extensions of the classical tests to the case of bivariate point patterns.

I assess the power of my new test experimentally in a simulated dataset, where it compares favorably to existing methods for testing for space-time interaction, without requiring precise specification of various parameters. Space-time interaction tests are most widely used in the epidemiological literature focusing on the question of, e.g. the etiology of childhood leukemia (Alexander et al., 1998). I demonstrate their use in the case of urban event data, asking whether various types of crime and disorder have space/time clustering.

Our Kernel Space-Time interaction test (KST) is a novel contribution which also gives a new perspective on the classical Mantel test, provides an alternative to classical tests for space-time interaction, and shows how kernel-embedding techniques can be used with spatiotemporal point processes.

3.2 Classical Tests for Space-Time Interaction

Let $\mathcal{P} = \{(s_i, t_i), i = 1, \ldots, n\}$ be a realization of a spatiotemporal point process with two spatial dimensions ($s_i \in \mathcal{R}^2$) and a time dimension. We can think of $s_i \in A$ for a spatial region A and $t_i \in T$ for a time window T. An illustration is shown in Figure 3.1.

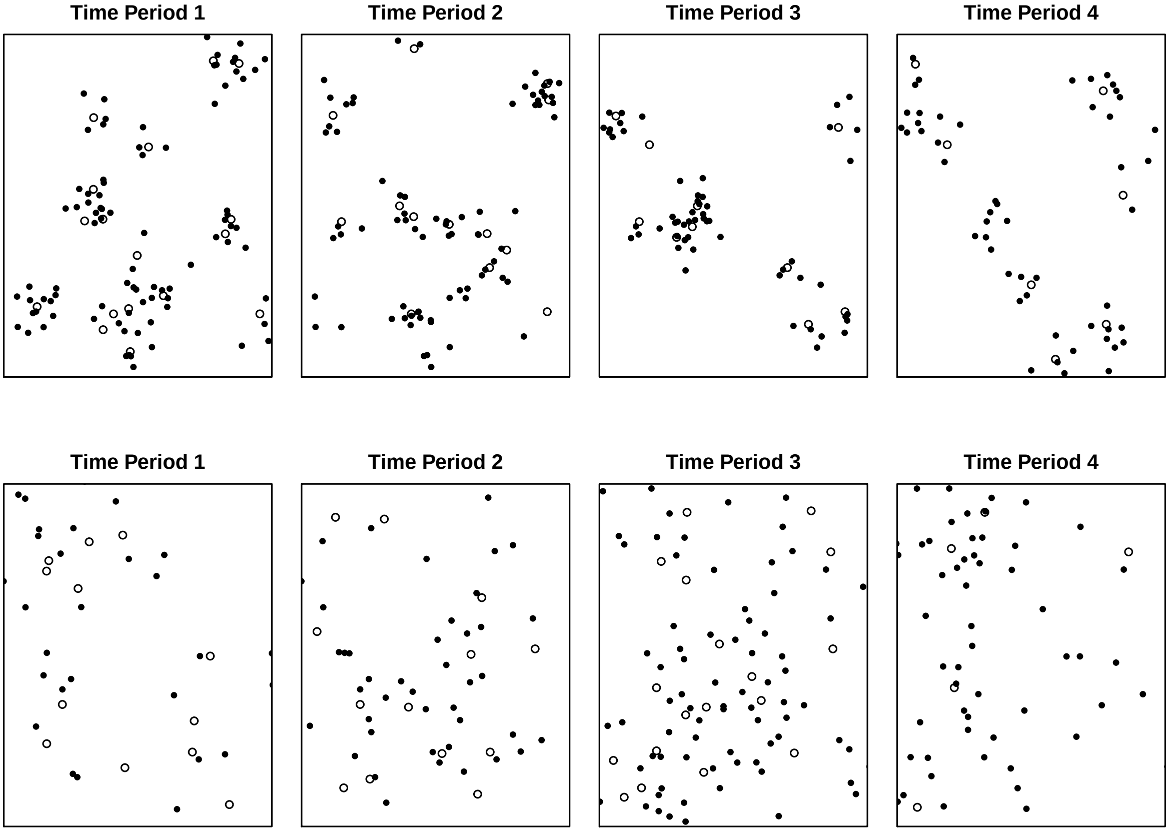


Fig. 3.1 Two different "infectious" Poisson cluster processes with parents shown as open circles and children shown as filled circles. Children are displaced from parents in space and time by iid draws from $N(0, \sigma)$. The top row displays the case for $\sigma = 0.05$, the bottom row for $\sigma = 0.2$. Visual inspection reveals space-time interaction in the first row while the second row is more ambiguous. Tests for space-time interaction correctly reject the null hypothesis (of no space-time interaction) in both cases, with $p \leq 0.01$.

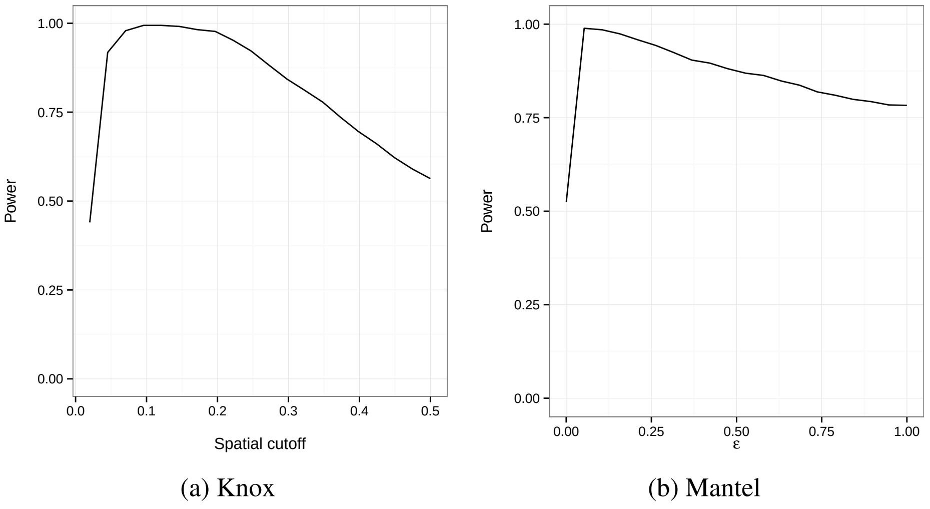
The unifying framework for understanding each test stated below is that of testing for statistical independence, which was introduced in Section 2.5. For each test, we assess (using more or less powerful tests) whether distributions related to the spatial and temporal locations of the points are independent.

I start by stating the **Knox test** (Knox, 1964). Given $\mathcal{P}$, we create a two-by-two contingency table as follows: pick a threshold distance for "near in space" s_0 and a threshold time interval for "near in time" t_0. Now, consider every pair of distinct points $s, s' \in \mathcal{P}$. Let $d_s(p, p')$ measure the Euclidean distance between p and p': $\sqrt{(x-x')^2+(y-y')^2}$ and $d_t(p, p')$ measure the time interval: $|t-t'|$. Then, we can fill in the table by asking for each pair of points whether $d_s(p, p') \leq s_0$ and whether $d_t(p, p') \leq t_0$:

	near in space	far in space
near in time	X	n_1
far in time	n_2	$N-(X+n_1+n_2)$

If there are $N = n(n-1)/2$ pairs of points, the test statistic is given by the difference between the number of pairs that we observe to be near in both time and space, X, and the number of pairs that we would expect to be near in both time and space if time and space are independent: $N\frac{X+n_1}{N}\frac{X+n_2}{N}$. Together this is: $X - \frac{1}{N}(X+n_1)(X+n_2)$. Since the null hypothesis is that space and time are independent, we can empirically find the distribution of X under the null by randomly permuting the time labels and recomputing the test statistic. Notice that $X+n_1$, the number of points that are close in time, is unchanged if the time labels are permuted. The same is true of $X+n_2$. This simplifies our calculations, and we need only consider the distribution of the test statistic X under the null. Various asymptotic approximations to the null distribution are discussed in Kulldorff and Hjalmars (1999).

The Knox test is very straightforward, but it clearly has limitations. Correctly specifying the spatial and temporal ranges is not always easy, and considering a range of values leads to problems of multiple hypothesis testing. As a toy example, Figure 3.2(a) shows how the power of the Knox test depends critically on the choice of cutoffs. We generated synthetic data from a point process with space-time clustering, using the setup discussed in Section 3.5.1, and varied the spatial cutoff for the Knox test from 0 to 0.5. When the spatial cutoff is equal to about 0.1, the test correctly rejects the null in almost every case (α is fixed at 0.05). But for smaller and larger values of s_0, the power decreases. For further intuition, consider the illustrative dataset in Table 3.1. If the cutoff is set such that close in space is defined as ≤ 1 hour then every cell in the table will equal 10 and the Knox test will fail to reject the null hypothesis of no space-time interaction. But if the cutoff is set such that close in space is defined as ≤ 2 hours, the Knox test will reject the hypothesis of independence ($p = 0.01$).



(a) Knox (b) Mantel

Fig. 3.2 We generated synthetic data from a cluster point process with child points displaced from their parents a distance $\sim N(0, \sigma = .05)$ in space and time. For $\alpha = 0.05$, the Knox test correctly rejects the null when the spatial cutoff is well chosen, but as the cutoff decreases or increases, the power decreases. The temporal cutoff is fixed at 0.1 in every case. This demonstrates that the power of the Knox test depends on correctly specifying cutoffs for "close in space" and "close in time." Similarly, the Mantel test's power depends on correctly specifying a transformation from distance to "similarity." In Mantel's original formulation, distances x were transformed as $f(x) = \frac{1}{x+\varepsilon}$ for some ε. On the right, the Mantel test correctly rejects the null almost all the time when ε is well chosen, but as ε increases or decreases the power decreases. The same transformation was used for space and time.

Another concern is that the Knox test is based solely on distances between points, ignoring any other relevant features, like location in space and time. When Knox proposed his test, he was quite explicit, stating that *all* of the information required for a test of space-interaction is found in the interpoint time and space distances (Knox, 1964). But his claim ignores the possibility of other types of inhomogeneities, as was pointed out at the time (Bartlett, 1964).

Next, we describe the **Mantel test** (Mantel, 1967). Given $\mathcal{P}$, we create an $n \times n$ spatial distance matrix D_S with entries given by $d_s(p_i, p_j)$ for row i and column j and an $n \times n$ temporal distance matrix D_T with entries given by $d_t(p_i, p_j)$. As with the Knox test, we wish to ask whether space and time, now represented by two matrices, are independent. We string out the entries above the diagonal of each matrix as a vector with $n(n-1)/2$ entries, and calculate the Pearson correlation between these vectors. Notice, however, that the usual significance test for Pearson's correlation is not valid, because the observations are not independent. To derive the null distribution, we again turn to randomization testing, this time applying a given permutation to the rows and columns of one of the matrices, so as to preserve the dependence

	close in space	far in space
1 hour	10	10
2 hours	9	1
3 hours	1	9

Table 3.1 In this simple example, the detection of space-time interaction is sensitive to the choice of temporal cutoff. When close in time is defined as ≤ 1 hour, the contingency table reduces to a table with each cell equal to 10, consistent with independence. But when the temporal cutoff is ≤ 2 hours, the null hypothesis of independence is rejected (p = 0.01) for the resulting table.

structure among the entries. We are typically concerned about shorter time and spatial distances, but the Mantel test could be significant due to (spurious) longer range features. Mantel 1967 proposed the reciprocal transformation for both spatial and temporal distances x, forming the matrices of $f_s(d_s(p_i, p_j))$ and $f_t(d_t(p_i, p_j))$ where $f_s(x) = \frac{1}{x+\varepsilon_s}$ and $f_t(x) = \frac{1}{x+\varepsilon_t}$. The Mantel test is essentially a linear test of dependence, so we expect it to have the same shortcomings as Pearson correlation, i.e. zero correlation implies no linear relationship, but it does not imply independence.

Diggle et al. (1995) proposes a test with a similar flavor to the Knox test, but rather than a single threshold value, it requires the specification of a range of values. First, we define Ripley's K function (also called the reduced second moment measure) for a single spatial point process as the following:

$$K(s) = \frac{1}{\lambda_S} \mathbf{E}\left[\#\text{ of events occurring within a distance } s \text{ of an arbitrary event}\right] \tag{3.1}$$

where λ_S is the intensity of the point process. An estimate of λ_S is given by $\widehat{\lambda}_S = N/A$ for N points in a spatial region with area A.

Given spatial point locations $p \in \mathcal{R}^2$ in a region with area A, the simplest way of estimating $\widehat{K}(s)$ is by averaging:

$$\widehat{K}(s) = \frac{1}{\lambda_S} \sum_{i=1}^{n} \frac{1}{n-1} \sum_{i \neq j} I(d_s(p_i, p_j) \leq s) \tag{3.2}$$

$$= \frac{A}{n(n-1)} \sum_{i} \sum_{i \neq j} I(d_s(p_i, p_j) \leq s) \tag{3.3}$$

This estimator assumes a known constant first order intensity λ_S. Ripley (1976) discusses approaches to estimating both K and λ_S. This test also ignores the issue of edge corrections: at the boundary of the spatial or temporal region, "missing" observations bias the estimate. This becomes an issue for small n or for s large compared to A. Corrections are given in (Ripley, 1982). The new test that we propose does not have this shortcoming.

Ripley's K function has natural extensions to the purely temporal $K(t)$ and space-time $K(s,t)$ cases, with similar estimators to the above. I remark that $\widehat{\lambda}_{ST}\widehat{K}(s_0,t_0)$ is equal to the entry in the upper-left hand corner of the contingency table used in Knox's test, and similarly $\widehat{\lambda}_S\widehat{K}(s_0)$ and $\widehat{\lambda}_T\widehat{K}(t_0)$ are equal to the top row and left column, respectively.

Diggle et al. define residual space-time interaction at spatial scale s and time scale t as:

$$D(s,t) = K(s,t) - K(s)K(t)$$

Using this function, Diggle et al. define a test statistic calculated over a grid of pre-specified spatial distances $s_1,\ldots,s_k$ and time intervals $t_1,\ldots,t_l$:

$$R = \sum_{s_i}\sum_{t_j} D(s_i,t_j)$$

Under the null hypothesis of no space-time interaction, the expectation of R should be constant (but not necessarily zero as claimed by Diggle et al. (1995), see Møller and Ghorbani (2012)). The intuition is the same as for the previous tests: $K(s,t)$ tells us how many points we expect to see within a distance s and time t of an arbitrary point. As in the previous tests, permutation testing by shuffling the time labels is used to obtain the null distribution of R. The Diggle et al. test is meant to address the issue of multiple hypothesis testing that arises when the Mantel or Knox test are applied repeatedly. However, it may lose power due to the fact that it is measuring a statistic of interest over multiple thresholds: this statistic may be positive or negative at different thresholds, and thus may cancel out, or it may be zero at many thresholds and thus go undetected.

This completes our presentation of classical space-time interaction tests. Note that we have not provided an exhaustive review. Other tests for point processes include Jacquez's nearest neighbor based method (Jacquez, 1996). Various improvements to the Knox test have been proposed in (Baker, 1996; Kulldorff, 1997). There is also a parallel literature in geostatistics and Gaussian processes on tests for the separability (defined in Section 2.2) of space-time covariance functions (Fuentes, 2006; Gneiting et al., 2007).

Notice the commonalities among the tests: each is a hypothesis test with the same null hypothesis, that the interpoint spatial and temporal distributions are independent. To see this, note that the contingency table in the Knox test is used to ask whether binary indicator variables for pairs of points (near in space, near in time) are independent. The Mantel test uses Pearson correlation to test whether the interpoint space and interpoint time distributions are independent.

Diggle et al.'s test asks whether there is a difference between the joint K function which counts the number of points that are near in space and near in time and the product of the marginal K functions counting the number of points that are near in space and the number of points that are near in time.

3.3 Kernel Space-Time Interaction Tests

As an intermediate step towards using kernel embeddings to test for space-time interaction, and because it sheds light on the classical version of the Mantel test, we define a kernelized version of the Mantel test. The Mantel test was described in Section 3.2. I briefly restate it in a more general form, following Legendre and Legendre (2012). The Mantel test measures the correlation between a pair of dissimilarity (distance) matrices. Given a set of objects P, and two different ways of measuring the dissimilarity between these objects, the null hypothesis is that the two different types of measurements are independent. Given, e.g. two $n \times n$ matrices of distances K and L where $k(i,j)$ gives the Euclidean distance between objects i and j and $\ell(i,j)$ gives some other dissimilarity measure, the Mantel test statistic is $\sum_{i \neq j} k(i,j)\ell(i,j)$. Interestingly, this is the first term in the estimator for HSIC, as shown in Equation (2.26). While the Mantel test is usually presented in terms of distance matrices, it is valid for similarity matrices as well. I propose considering a kernelized version of the Mantel test. Given objects $P = (p_1, \dots, p_n)$ and two kernels k and ℓ, we construct the Gram matrices K and L and ask, as in the Mantel test, whether the two kernels are measuring independent properties of the objects of P.

Once we have Gram matrices, we proceed exactly as with the Mantel test, defining the test statistic $T = \sum_{i \neq j} k(i,j)\ell(i,j)$, and obtaining significance levels by randomization testing. I call this test the "kernelized Mantel test." To my knowledge, it has not been explicitly considered in the literature, but in fact, the reciprocal transformation considered by Mantel (Mantel, 1967) ($f(s) = \frac{1}{s+\varepsilon}$) is an example of a Mercer kernel: $k(x,x') = \frac{1}{\|x-x'\|^2+\varepsilon}$ (Micchelli, 1986) (as cited in (Souza, 2010)).

With this approach as background, we are ready to define a new test for space-time interaction based on kernel embeddings. In the spirit of the classical tests described in Section 3.2, the most straightforward approach to using HSIC would be to define new distributions $P = \{d_s(i,j) : i \neq j\}$ for the Euclidean distances between pairs of points and $Q = \{d_t(i,j) : i \neq j\}$ for the interpoint time intervals, and apply HSIC as a black box to test whether the distributions P and Q are independent. However, this is not an attractive option computationally, as it leads to $O(n^4)$ computations because HSIC considers pairs of observations, and in this case observations are themselves pairs of points.

I now consider an alternative, more computationally efficient approach which we term the Kernel Space-Time (KST) test. To motivate this test, let us more closely inspect my kernelized Mantel test, and in particular how it differs from the HSIC test statistic. Recall the HSIC estimator of Section 2.5:

$$\widehat{\text{HSIC}} = \frac{1}{n^2}\sum_{i,j} k(x_i,x_j)\ell(y_i,y_j) - \frac{2}{n^3}\sum_{i,j,q} k(x_i,x_j),\ell(y_i,y_q) + \frac{1}{n^4}\sum_{i,j,q,r} k(x_i,x_j)\ell(y_q,y_r)$$

The last term of the HSIC estimator, which estimates $||\mu_p \mu_q||^2$, is unchanged by randomization testing, so the key difference is the cross-term, $\langle \mu_p \mu_q, \mu_{pq} \rangle$. Recall our definition of the covariance operator in Eq. (2.25). An equivalent definition is:

$$\Sigma_{PQ} = E_{xy}[(\phi(x) - \mu_p) \otimes (\psi(y) - \mu_q)]$$

where ϕ is the feature embedding for $\mathcal{H}_{\mathcal{K}}$ and ψ is the feature embedding for $\mathcal{H}_{\mathcal{L}}$. Thus, we see that the cross-term in $||\Sigma_{PQ}||^2_{HS}$ arises because the feature vectors $\phi(x)$ and $\psi(y)$ are centered before being multiplied together (by analogy, we can write: $\text{Cov}(P,Q) = E[(P - E[P])(Q - E[Q])]$). Returning to the Mantel test, this is the critical difference—the Mantel test measures dependence by calculating the inner product between two matrices treated as vectors, where these vectors are centered by subtracting the mean of their entries, that is, subtracting the mean of the empirical distribution over pairwise distances.[1]. But this is *not* equivalent to the centering done by HSIC: $\tilde{\phi}(x) = \phi(x) - \mu_p$ centers the feature embedding so that it has mean 0. From this, the centered Gram matrices $\tilde{K} = HKH$ and $\tilde{L} = HLH$ where $H = I - \frac{1}{n}11^T$ are calculated, and then the covariance is measured as $\frac{1}{n^2}\,\text{tr}\,\tilde{K}\tilde{L}$.

This suggests a simple fix for the Mantel test, which can even be applied to the classic version. Given similarity, dissimilarity, or Gram matrices K and L, calculate $\tilde{K}$ and $\tilde{L}$ and then apply the Mantel test: $\sum_{i,j} \tilde{K}_{ij}\tilde{L}_{ij}$. Since this is proportional to $\frac{1}{n^2}\,\text{tr}(\tilde{K}\tilde{L})$, my final "Kernelized Space-Time" (KST) test takes the same form as HSIC[2]

[1] In fact, some formulations of the Mantel test actually calculate Pearson's correlation: given matrices K and L, string out the upper-triangle of K and L as vectors $\vec{K}$ and $\vec{L}$ and calculate:

$$\frac{\langle \vec{K} - \mu_{\vec{K}}, \vec{L} - \mu_{\vec{L}} \rangle}{||\sigma_{\vec{K}}|| \cdot ||\sigma_{\vec{L}}||}$$

Note that using Pearson correlation instead of covariance does not change the significance level derived from randomization testing. Nor does it change the fact that the centering occurs by subtracting the mean of the dot products.

[2] Note that this same fix for the Mantel test has been independently proposed in the distance correlation literature. For a full discussion, see Omelka and Hudecová (2013); Szekely et al. (2014).

I now show an alternative and simple perspective on my new KST test. Given a probability distribution over points in space $A = \{(s,t)\}$, with $s \in \mathcal{R}^2$ and $t \in \mathcal{R}$ and kernels k (for RKHS $\mathcal{H}_\mathcal{K}$) and ℓ (for RKHS $\mathcal{H}_\mathcal{L}$), let $k(a,\cdot) = k(s,\cdot) = \phi(s)$ and $\ell(a,\cdot) = \ell(t,\cdot) = \psi(t)$, so that k embeds the spatial coordinates of A with feature map $\phi(s)$, ignoring the temporal coordinates, and ℓ embeds the temporal coordinates of A with feature map $\psi(t)$, ignoring the spatial coordinates. The null hypothesis we wish to test is that for a random $a \sim A$:

$$H_0 : k(a,\cdot) \perp\!\!\!\perp \ell(a,\cdot)$$

or equivalently:

$$H_0 : \phi(s) \perp\!\!\!\perp \psi(t)$$

which exactly captures the hypothesis that space and time are independent. Hypothetically, we could apply feature embeddings to this, considering embeddings of $\phi(a)$ and $\psi(a)$ into a different feature space and using HSIC, but since we are already in feature space, and assuming we have chosen characteristic or universal kernels (see Section 2.4), we might consider simply checking whether:

$$\mathbf{E}_a[\phi(s)\psi(t)] = \mathbf{E}_a\phi(s)\mathbf{E}_a\psi(t) \tag{3.4}$$

The test statistic derived from this expression is exactly the KST test statistic I proposed above. By the derivation of HSIC, Eq. (3.4) holds if and only if the underlying distributions which we've embedded are independent. In this case, those underlying distributions are simply the distribution of points in space and the distribution of points in time.

To recap, given the space-time coordinates of a set of points, we wish to test whether there is space-time interaction. Using kernels, we represent these points through their similarity to every other point, i.e. we represent these points using a kernel $k(a,\cdot) = \phi(s)$—a measure of the spatial distance between point s and any other point and by $\ell(a,\cdot) = \psi(t)$—a measure of the time interval between point a and every other point. Given these representations, we proceed just as in the classical tests, asking whether the distribution over spatial distances $\phi(s)$) is independent of the distribution over time intervals $\psi(t)$. Note that if we want to stay as close as possible to classical tests for space-time interaction, we could insist that k and ℓ be stationary so that $k(s,s') = k(\|s-s'\|)$ and $\ell(t,t') = \ell(|t-t'|)$, but this additional assumption is not necessary in our framework.

3.4 Extending the Classical Tests to Bivariate Space-Time Interaction

In this section, we present the standard approach to extending classical space/time interaction tests to the bivariate case. These tests only work under certain restrictive assumptions, a fact that has not been previously established in the literature. Our new kernel perspective helps clarify these points.

Given $\mathcal{P}^1 = \{(s_i^1, t_i^1), i = 1, \ldots, n_1\}$ and $\mathcal{P}^2 = \{(s_i^2, t_i^2), i = 1, \ldots, n_2\}$, we wish to know whether there is significant space-time interaction between $\mathcal{P}^1$ and $\mathcal{P}^2$. The null hypothesis is that there is no space-time interaction between the two processes. Notice that we are not interested in whether there is purely spatial dependence between $\mathcal{P}^1$ and $\mathcal{P}^2$: any two processes associated with, for example, an underlying population density will be spatially correlated. Similarly, we are not interested in purely temporal dependence between the two processes, e.g. due to seasonal trends. Instead, we wish to test whether seeing points of type 1 at a certain location in space and time makes it more or less likely that we will see points of type 2 nearby in space and time, once we have controlled for separable spatial and temporal correlations between $\mathcal{P}^1$ and $\mathcal{P}^2$. Considering open circles as type 1 and closed circles as type 2, Fig. 3.1 is an example of space-time interaction.

The Mantel, Knox, and Diggle et al. tests each focus on pairs of points. For the bivariate extension for each, we consider all $n_1 \cdot n_2$ cross-pairs of points. For the Knox test, we create the same contingency table, where each entry counts the number of cross-pairs that are near in time and near in space, the number of cross-pairs that are near in time and far in space, etc. For randomization testing, the standard approach in the literature is to permute the time labels of only one of the point processes. For the Mantel test, we create an $n_1 \times n_2$ spatial cross-distance matrix and an $n_1 \times n_2$ temporal cross-distance matrix, and the test statistic is the same. The bivariate version of the Mantel test was explored in (Klauber, 1971). The Diggle et al. extension is straightforward as well (Lynch and Moorcroft, 2008).

However, there is an underappreciated problem with these tests. They are only valid in the case that one or both of the point processes does not have within-type space/time interaction. The reason for this requirement is that if there is within-type space/time interaction in one of the processes but but no cross-type space/time interaction, then we may incorrectly reject the null, and incorrectly conclude that $\mathcal{P}^1$ and $\mathcal{P}^2$ are not independent. I demonstrate this by counterexample below. The basic reason is that we *cannot* in general simulate from the correct null hypothesis. When we try to do so by permuting the time labels, we have to be very careful. If we permute the time labels of a process with within-type interaction in order to test for cross-type interaction, the effect is that we destroy the within-type interaction, so the observed data will not have come from the null distribution that we simulate, and we will incorrectly reject the null hypothesis. Since we can choose which type to permute the time labels of, these tests remain valid if at least one type does not have within-type interaction, and we permute the time labels of this type only.

The kernel perspective helps clarify these problems. Let $\mathcal{P}^1$ and $\mathcal{P}^2$ be two point patterns for which we want to test whether there is cross-type space-time interaction. Consider kernel mean embeddings for each type, μ_P^1 and μ_P^2. We wish to ask whether $\mathcal{P}^1 \perp\!\!\!\perp \mathcal{P}^2$ and as before we might hope to that kernel mean embeddings will allow us to test whether $P(P^1,P^2) = P(P^1)P(P^2)$. The problem is that if we consider the mean embedding corresponding to the LHS probability distribution we have μ_{P^1,P^2} while if we consider the mean embedding for the RHS probability distribution we have $\mu_{P^1} \otimes \mu_{P^2}$. Theoretically, we can ask whether these are different. But in practice, we have no way of estimating them because we do not have access to paired samples of points of type 1 and type 2. When we calculate the bivariate Knox test, we consider all pairs of points of type 1 and 2, just as we do when we calculate μ_{P^1,P^2}. But then when we consider estimating $\mu_{P^1} \otimes \mu_{P^2}$ we simply estimate μ_{P^1} and μ_{P^2} separately, and then consider the tensor product space, which contains every pair of points, again! In the Knox test, we simulate from the null by permuting the time labels of e.g. type 2. But if we consider $p^1 = (s^1, t^1)$ and $p^2 = (s^2, t^2)$ so that we have embeddings μ_{s^1,t^1} and μ_{s^2,t^2}, then permuting t^2 turns μ_{s^2,t^2} into $\mu_{s^2}\mu_{t^2}$ (just as it did in the univariate case). The result is that we are testing:

$$\mu_{s_1,t_1,s_2,t_2} = \mu_{s_1,t_1}\mu_{s_2}\mu_{t_2} \tag{3.5}$$

Or equivalently:

$$\mu_{s_2,t_2} = \mu_{s_2}\mu_{t_2} \tag{3.6}$$

But this is not what we want to test at all! If there was actually no within-type interaction, then we would still not have a valid kernelized test, because we'd have:

$$\mu_{s_1,t_1,s_2,t_2} = \mu_{s_1}\mu_{t_1}\mu_{s_2}\mu_{t_2} \tag{3.7}$$

And permuting t_2 would have no effect.

In Figure 3.3 we show a multitype point process with no cross-type interaction. Type 1 (open circles) is a homogeneous Poisson process while type 2 is a Poisson cluster process following the setup in Section 3.5.1. But the Knox test gives a significant p-value because of type 2's within-type interaction.

Interestingly, at least one instance of this last example will be correctly handled by the classical tests. Let P^1 be a homogeneous Poisson process and P^2 be a copy of P^1 with a

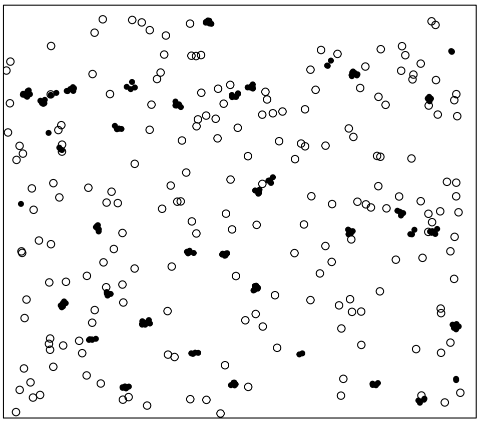


Fig. 3.3 A multitype point process in which points of type 1 (open circles) are drawn from a homogeneous Poisson process while points of type 2 (closed circles) are drawn from a Poisson cluster process. The Knox and Mantel test incorrectly reject the null hypothesis of cross-type independence due to type 2's within-type interaction.

slight amount of random jitter added to each point in space and time. Then the classical tests, which calculate distances in space and time between points of different types will have many entries that for which the distance in space and the distance in time are both very close to zero. But permuting t_2 will eliminate these entries. The fact that the classical tests throw away information, focusing only on the distances between the points (a second order measure), can actually help, suggesting that one solution might be a return to the inefficient test proposed at the beginning of Section 3.3 which relied on kernel embeddings for the interpoint distances.

I close with three remarks. First, the cases in which the classical bivariate extensions fail are those for which there is within type interaction, and these can be readily diagnosed using the univariate test. Second, a permutation-based approach might still be feasible, but the permutation must be more structured so as to preserve within-type interaction. The same issue arises in time series, where the block bootstrap (Kunsch, 1989) has been proposed. In spatial statistics a much less well researched version of the block bootstrap resamples spatial rectangles (Finkenstadt and Held, 2006). I thus propose permuting spatiotemporal cubes of one of the two types of points as a way of testing whether $\mu_{P1} = \mu_{P2}$ with the test statistic $\|\mu_{P_1} \mu_{P_2}\|$. Third, if we are willing to consider aggregating the point pattern to some kind of grid—which can be very helpful computationally as discussed in Chapter 4—then we can fit a Gaussian process model to the spatiotemporal count process and either use a multi-output GP with an appropriate parameter modeling cross-type interaction or use the methods for testing for independence with spatiotemporally-observed data that I develop in Chapter 6 to test whether the two processes are associated.

3.5 Experimental Evaluation

Below, I describe experimental evaluations of my new space-time test. First, using synthetic data, I compare the performance of our test as compared to the classical tests. Second, I show the applicability of our methods to urban event data: using publicly available data on calls for service and crime incidents, I ask whether and which types of citizen complaints exhibit space/time interaction.

3.5.1 Synthetic Data

My power analysis, inspired by the one in (Diggle et al., 1995) uses the following setup for a Poisson cluster process: parent locations (x,y,t) are sampled on the unit cube. The number of children for each parent is drawn iid $\sim$ Poisson(5). The location of each child is generated as a random displacement from the parent's location, in space and time, where each coordinate's offset is independently sampled from $N(0,\sigma)$. This induces space-time interaction, and as σ increases, the signal of this interaction becomes swamped by noise. Figure 3.1 shows two examples, one with $\sigma = 0.025$ and the other with $\sigma = 0.2$. I consider $0 < \sigma \leq .4$.

I will consider the same set of tuning parameters $\Delta = \{0.05, 0.10, \ldots, 0.25\}$ for each test. For the Knox test, the spatial cutoff varies over Δ while the temporal cutoff is fixed at 0.1. For the Mantel test, I use the transformation considered earlier: $\frac{1}{x+\varepsilon}$ for $\varepsilon \in \Delta$. For the Diggle et al. test, I follow (Diggle et al., 1995) and use a grid of side length varying over Δ for the points at which the K function is evaluated. The grid always has the same coarseness 0.01. For KST, the bandwidth σ of the RBF kernel varies over Δ. For each method, each value of Δ, and each value of σ, I draw 500 random point patterns and obtain p-values using randomization testing. The power is shown in Figure 3.4 as the fraction of simulations which correctly rejected the null hypothesis of independence between space and time at $\alpha = 5\%$. The four methods are compared in Figure 4.6. For each method, the relevant parameter that was chosen was the parameter with the highest power for $\sigma = 0.15$. When σ is small, all methods have equally high power, but as σ increases, the power decreases at different rates. The KST method I proposed has the highest power for $\sigma > 0.1$.

3.5.2 Real Data

I obtained geocoded, date-stamped locations of crimes and calls for 311 from Chicago's open data portal[3] covering 2010-2012. There were 37 different types in the dataset, including calls to 311 about rats and graffiti, and crime reports about arson, homicide, and burglary. For each type of crime or call type I calculated the KST test statistic using RBF kernels with a bandwidth of 7 days and 1/4 mile. I implemented KST using the same approach as the reference implementation of HSIC [4], with the Gamma approximation to derive the significance level. This implementation does not scale to the size of our dataset, so when necessary I used independent random thinning of our dataset to retain $n = 10,000$ observations. The 311 call types (complaints) I considered were: vacant / abandoned building, alley light out, garbage cart, street lights out, rodent, pot hole, abandoned vehicle, sanitation, tree debris, and tree trim. All were significant at $p < 0.001$. The crime types I considered were: burglary, theft, public peace, battery, assault, narcotics, criminal trespass, criminal damage, auto theft, other offense, deceptive practice, weapons violation, liquor, robbery, offense with children, sexual assault, homicide, stalking, arson, interference, kidnapping, prostitution, sex offense, gambling, and intimidation. The only types that were *not significant* were battery, assault, other offense, sexual assault, interference, and intimidation.

Recall that significant means that we reject the null hypothesis that the spatiotemporal distribution can be explained by the product of underlying spatial or temporal distributions. Thus for almost all of the types, the implication is either that there is actually an "infectious" component to the process or something else is driving the process (e.g. individuals as perpetrators or targets) and whatever this is, it has an infectious component. The fact that most types of calls to 311 are infectious could be explained very simply by considering, e.g. potholes—multiple citizens may report the same call, but our test has no way of knowing this and it would thus appear as space/time interaction, or a natural process (weather) may be the cause, so if this is sufficiently localized in space and time then it will lead to spatiotemporal clustering. The fact that most types of crime have an infectious component agrees with work in the criminology literature on crime as a contagious processes (Loftin, 1986). Thus it is very interesting to note that battery, assault, and sexual assault were not significant under our test, meaning non-contagious. While further research which would look at underlying causes and include covariates is of course necessary, these preliminary results suggest that there may be something interesting to uncover in the differences between assaults on the one hand, and most other crime types on the other hand.

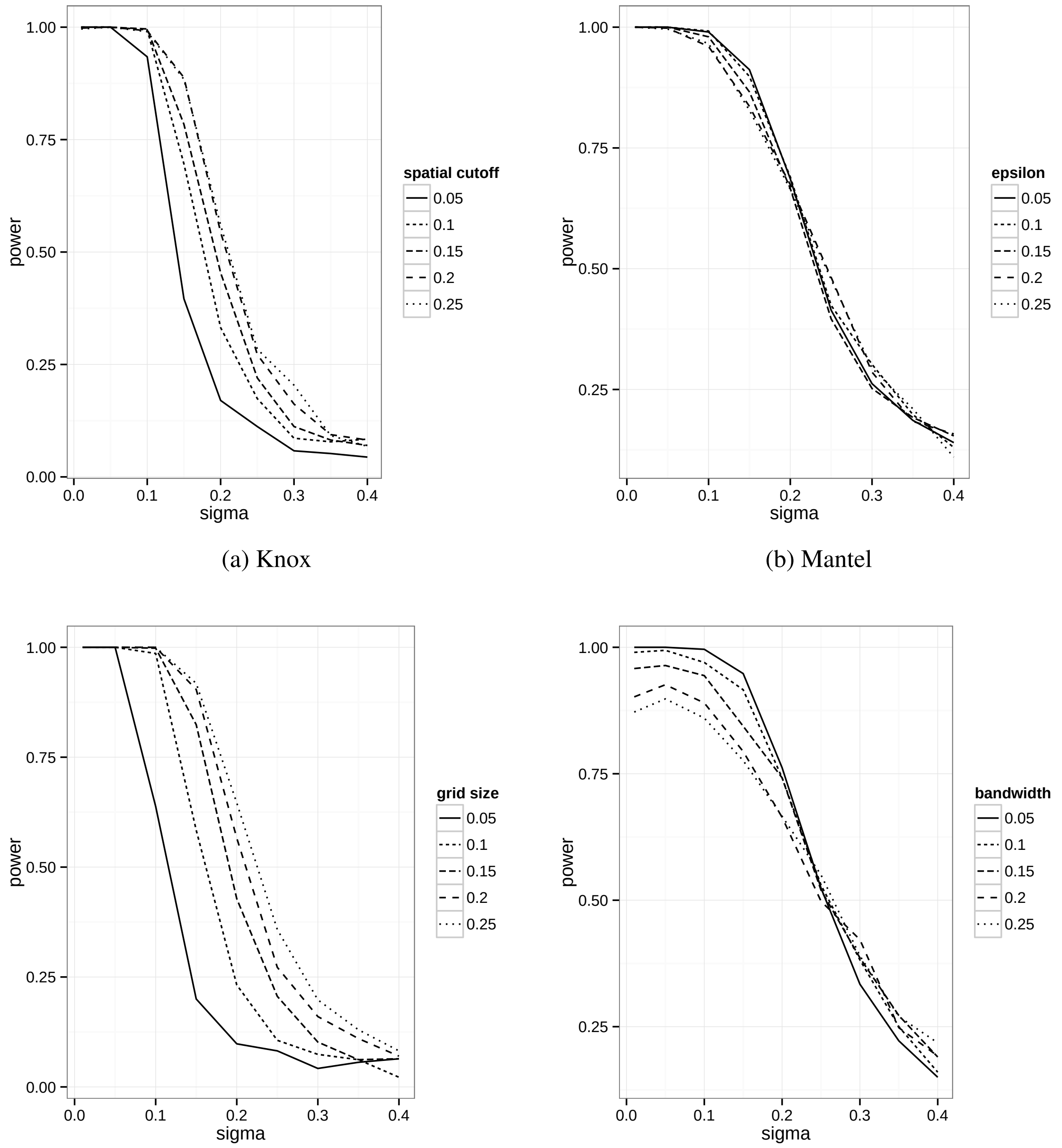
[3] data.cityofchicago.org

[4] http://www.gatsby.ucl.ac.uk/~gretton/indepTestFiles/indep.htm

3.6 Conclusion

In this chapter, I proposed a new method for address a long-standing question asked in various disciplines including epidemiology, criminology, and environmental science, namely, given a set of spatiotemporal locations of a particular type of event, can we say that they exhibit space/time interaction? I highlighted the deficiences of the classical tests addressing this question and proposed my new Kernel Space-Time interaction test. Along the way, I proposed a principled fix to the often criticized Mantel test. My test is easy to implement either from scratch or using existing code for the Hilbert-Schmidt Independence Criterion. My new kernel perspective on space/time interaction illuminates further issues that lie ahead in this literature as researchers ask important questions, like, do two point patterns exhibit cross-type space/time interaction? Future work in this area could focus on kernel choice and kernel learning and on scalable methods for very large datasets.

The framework of hypothesis testing is in many ways a limited one, as demonstrated by my experiments on real data. Going beyond a single p-value, I would like to be able to characterize the extent of the space/time interaction and ask whether its strength varies in space and / or time. In subsequent chapters, the Bayesian modeling framework that I emphasize will provide alternative methods for answering these types of questions. Through Gaussian process-based modeling of spatiotemporal data, I will set aside the framework of null hypothesis significance testing to enter the rich world of Bayesian modeling.



(a) Knox

(b) Mantel

(c) Diggle

(d) KST

Fig. 3.4 We compare the Knox, Mantel, Diggle et al., and newly-proposed KST tests on synthetic data. On the y-axis, we show the power as the fraction of simulations in which the test correctly rejected the null hypothesis of independence between space and time for $\alpha = 5\%$. In the simulations, a cluster point process is generated where children points are offset from a parent a distance $\sim N(0,\sigma)$ in each dimension. As σ grows, the problem becomes harder and each method's power decreases. For each method, we vary a tuning parameter: for Knox we vary the definition of "near" in space, for Mantel we vary the ε in the reciprocal transformation, for Diggle et al. we vary the grid size, and for KST we vary the bandwidth of the RBF kernel.

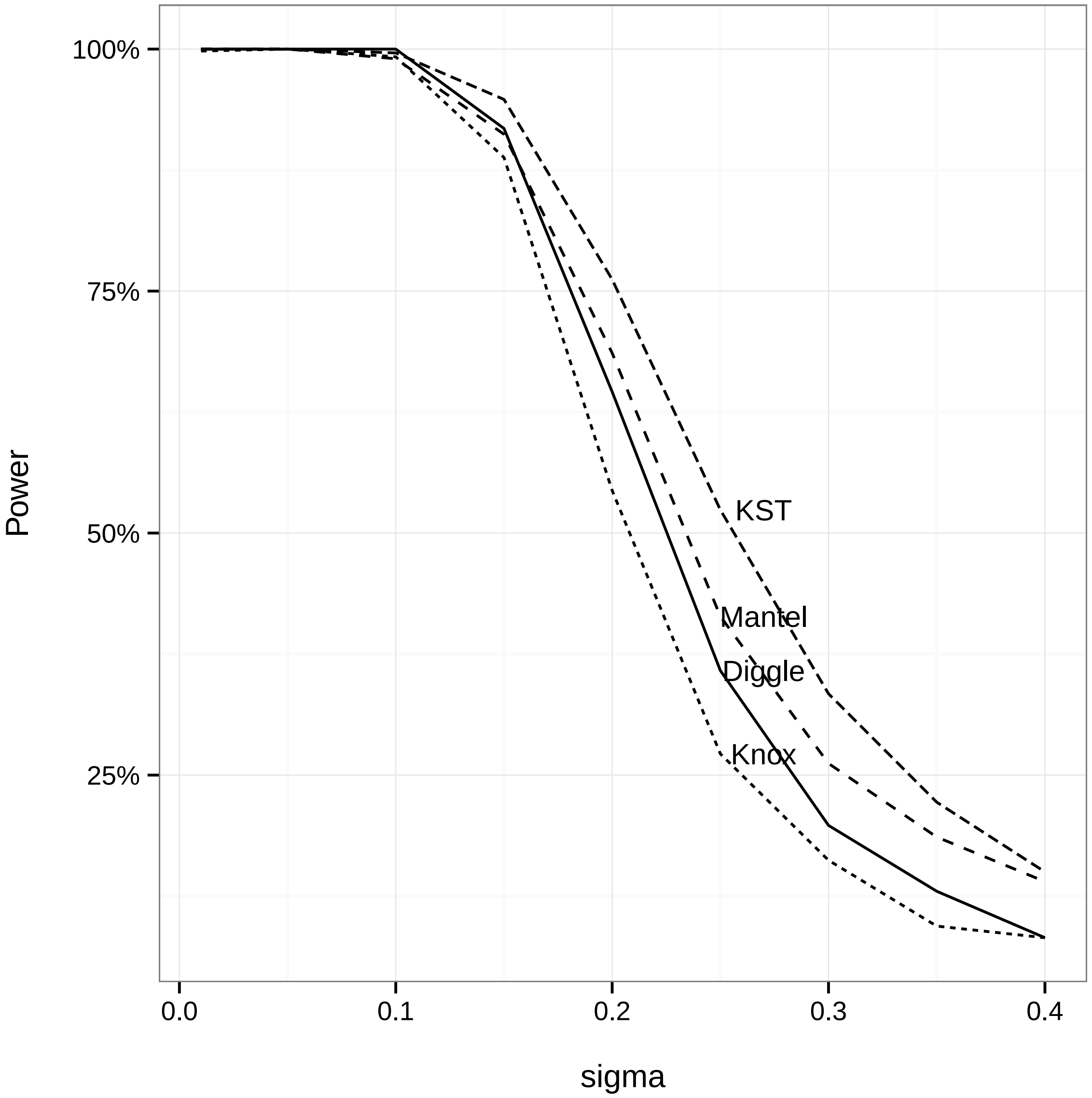


Fig. 3.5 The four methods from Figure 3.4 are shown here for comparison. For each method, the relevant parameter that was chosen was the parameter with the highest power for $\sigma = 0.15$. When σ is small, all methods perform equally well, but as σ increases, we see differences. The method I propose has the highest power for $\sigma \geq 0.1$.

Chapter 4

Gaussian process models for spatiotemporal learning and inference

> Two neighboring samples are certainly not independent... The misunderstanding of this fact and the rough transposition of classical statistics has sometimes led to surprising misjudgments. Around the fifties, in mining exploration, it was advised to draw lots to locate each drilling (i.e., to locate them exactly anywhere). Miners of course went on still using traditional regular grid pattern sampling, and geostatistics could later prove they were right Matheron (1963).

In this chapter, I introduce Gaussian process (GP) models and advocate their use as a general purpose method for spatiotemporal data. In Section 4.2, I discuss the problem of "pre-whitening" where a set of non-iid spatiotemporally referenced observations is transformed in such a way so that the resulting residuals can be treated as iid, and analyzed with classical statistical tools. A possible objection to GPs as a default method for spatiotemporal data is their high run-time and memory complexity. In Section 4.3, I discuss ways of exploiting structure in the covariance function of the GP to enable scaling to large datasets. I conclude in Section 4.10 with a discussion of recent results on the consistency of GPs and their convergence rates.

4.1 Definitions

A Gaussian process (GP) is a stochastic process over an index set $\mathcal{X}$. It is entirely defined by a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. These two functions are chosen such as to jointly define a normal distribution whenever we draw $f|X$ from a $\mathcal{GP}(\mu,k)$

on a finite set of locations $X := \{x_1, \ldots x_n\}$. More specifically, we have

$$f|X \sim \mathcal{N}(\mu(X), k(X,X)) \text{ where } \mu(X)_i = \mu(x_i) \text{ and } [K(X,X)]_{ij} = k(x_i, x_j). \tag{4.1}$$

By construction this means that $\mu(X)$ is an m dimensional vector, and $k(X,X) \in \mathbb{R}^{d \times d}$ is a positive semidefinite matrix, as discussed in Section 2.1.

Note that this is a realization of a stochastic process at a finite set of locations as described in Chapter 1, and we have *not* introduced a function defined over all inputs X. In fact, while there are some kernels leading to smooth processes (Wahba, 1990), this is in general not the case. In particular, quite often the realization $f(x)$ is nonsmooth while its prior is smooth. A well-known example is the Brownian Bridge. There is a subtle difference between functions and function values in the construction of a GP. For any infinite-dimensional GP, i.e. where the rank of $k(X,X)$ is unbounded, it is only possible to evaluate the GP pointwise. The technical challenge is that distributions over infinite-dimensional objects are nontrivial to define. Evaluating a GP on a finite number of locations sidesteps the entire problem.

As an illustration consider a Gaussian process with mean function $\mu = 0$ and Gaussian Radial Basis Function (RBF) kernel $k(x_i, x_j) = e^{-\|x_i - x_j\|^2}$. These parameters give a GP from which we can draw a realization. Since we want to know its value for a range of locations, we draw f for a grid of points. By construction they are drawn from a multivariate Gaussian distribution with mean $\mu = 0$ and covariance K.

Three different draws are shown in Figure 4.1, where we have used a sufficiently dense grid of points such that the function appears smooth. In a Bayesian framework, these are draws from the prior distribution before seeing any data. How do we update our prior given observations $Z = (X,Y)$? We start by specifying the joint distribution over both observed outputs Y and unobserved outputs Y^*:

$$\begin{bmatrix} Y & Y^* \end{bmatrix} \sim \mathcal{N}(\mu(\vec{x}), K)$$

where we can calculate $K(x_i, x_j)$ for any pair of x's, observed or unobserved, i.e. :

$$K = \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}$$

Since we've observed (X,Y), we can find the conditional distribution using the properties of multivariate Gaussian distributions, see e.g. (Rasmussen and Williams, 2006):

$$Y^*|Y \sim \mathcal{N}(K(X^*,X)K(X,X)^{-1}Y, K(X^*,X^*) - K(X^*,X)K(X,X)^{-1}K(X,X^*))$$

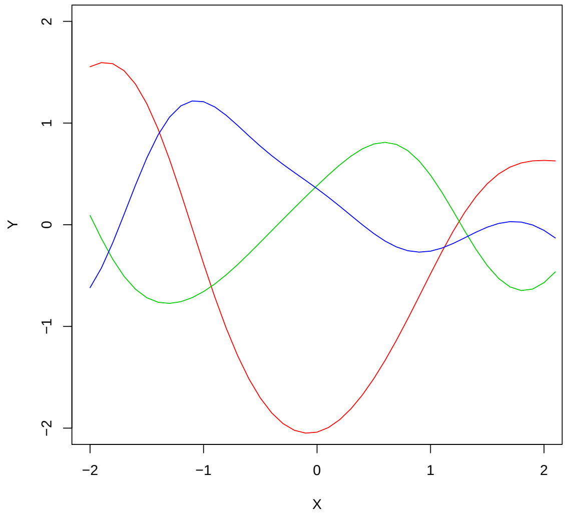


Fig. 4.1 Three draws from a GP prior with mean 0 and Gaussian RBF covariance function.

We give an illustration in Figure 4.2, where the observations $(-1,1),(0,0),(1,1)$ are shown in black circles and 10 posterior function draws f^* are plotted. Notice that there is no uncertainty at the observed points.

In some cases, like modeling computer simulations, this noise-free behavior might be desirable, but for real data generated by nature we need to include an extra noise term. If we believe our noise is iid, we can introduce a Gaussian observation error model, also known as the likelihood function in GP regression. We have the following hierarchical specification, introducing the following notation for a draw from a GP:

$$f \sim \mathcal{GP}(\mu, k) \tag{4.2}$$

$$Y|f \sim \mathcal{N}(f, \sigma^2 I) \tag{4.3}$$

Or equivalently, this says that once we have a draw f, each observed value Y_i is an independent draw from a Gaussian distribution centered at $f(X_i)$ with variance σ^2.

For a set of fixed locations X, this models is conjugate, so we can integrate out f and obtain an equivalent prior distribution over Y:

$$Y|X \sim \mathcal{N}(\mu, K + \sigma^2 I) \tag{4.4}$$

What does this extra variance σ^2 (called the "nugget" in geostatistics) do? In addition to providing extra numerical stability in calculations involving the covariance matrix (and thus

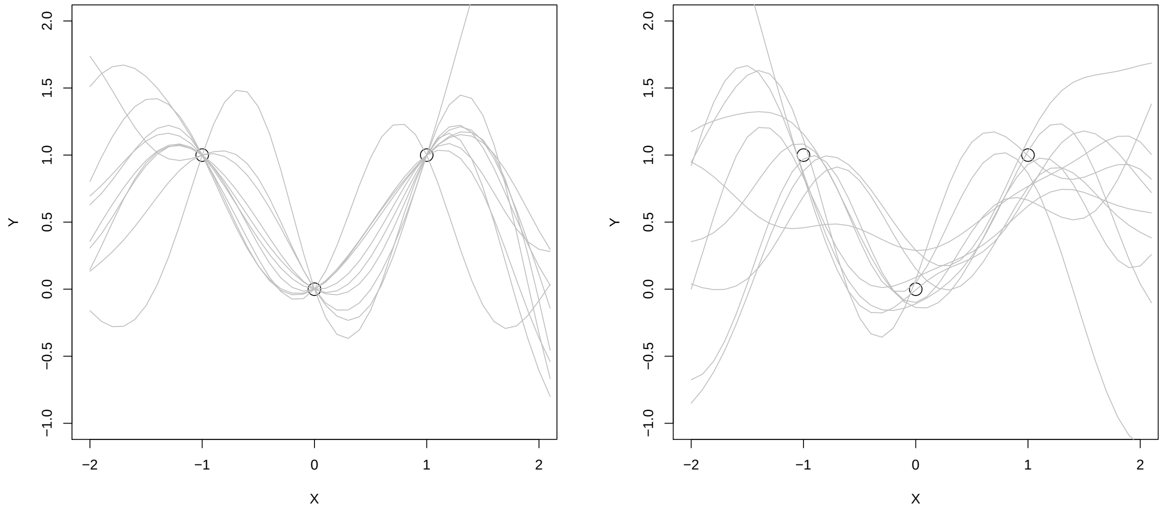


Fig. 4.2 Draws from a Gaussian process posterior with Gaussian RBF kernel after observations at $\{(-1,1),(0,0),(1,1)\}$. Left: noise free observations. Right: noisy observations with $\sigma^2 = 0.2$. Notice the difference in terms of uncertainty at the locations of measurement and the relative similarity otherwise.

being useful in practice), it has the effect of increasing the variance for observations, relative to the cross-covariance between nearby observations. If we use the same K as before, we have the following posterior:

$$\begin{aligned} Y^*|Y &\sim \mathcal{N}(\bar{\mu},\bar{K}) \qquad (4.5)\\ \text{where } \bar{\mu} &= K(X^*,X)(K(X,X)+\sigma^2 I)^{-1}Y \\ \bar{K} &= K(X^*,X^*) - K(X^*,X)(K(X,X)+\sigma^2 I)^{-1}K(X,X^*) \end{aligned}$$

Here $\bar{K}$ is the well-known Schur complement of the joint covariance matrix over X and X'. Note that the noise term σ^2 is only used for observed data Y. If we use this prior, we can draw 10 posterior functions as before. In Figure 4.2 (right) we have plotted these function draws. Notice that there is now some uncertainty, controlled by the parameter σ^2, at the observed points — even if we were to observe $y|x$ at the same location repeatedly, we would have no assurance that the observations would be identical. But as discussed in Chapter 1, it is usually the case that we only have one observation at any given space-time location, so it is very important that we include a likelihood function to model the observation error which is always present in real-world data.

Gaussian likelihoods are particularly nice because they lead to conjugate models, meaning the posterior of the Gaussian process is available in closed form as above. When GPs are used for classification or regression with counts or proportions other likelihoods like Binomial and Poisson are useful, as discussed in Section 4.4 and Section 5.4. However, these do not lead to closed form posteriors so approximate Bayesian methods (or Monte Carlo sampling) are necessary.

4.2 Pre-whitening spatiotemporal data

It is common for observational data to violate the typical assumption of independent and identically distributed (iid) observations. For instance, repeated measurements of neighborhoods or individuals by their nature have a temporal structure. Environmental measurements often have both temporal and spatial structure. This structure poses a particular problem when inferring dependence between random variables. As a motivating example, consider two independently generated autoregressive time series AR(1) on random variables X and Y according to the model

$$x_t = 0.9 \cdot x_{t-1} + \varepsilon_{x,t} \text{ and } y_t = 0.8 \cdot y_{t-1} + \varepsilon_{y,t} \text{ where } \varepsilon_{x,t}, \varepsilon_{y,t} \sim \mathcal{N}(0,1).$$

That is, X and Y are *independent* time series, each of which is corrupted at each step by adding normally distributed iid random variables. Despite the fact that X and Y are independent, the Pearson correlation between X and Y may be large in magnitude, due to the underlying autocorrelation structure of each time series, as shown in Figure 4.3.

While Fisher's z-transformation can be used to derive the distribution of the Pearson correlation statistic under linear independence, this assumes iid observations. But in the case of X and Y, our observations are neither independent nor identically distributed. The general guidance in the time series literature is to fit an appropriate autoregressive model to the data and to obtain residuals from this model Box et al. (2008). The intuition is that this *pre-whitening* should yield residuals which are iid, after which independence testing proceeds as usual. We formalize this notion in the present chapter.

Given observations $(X,S) = \{x_i, s_i\}$ where s_i is a location in space or time, we consider the model:

$$f \sim \mathrm{GP}(0,K)$$

with observation model:

$$X = f(S) + \varepsilon$$

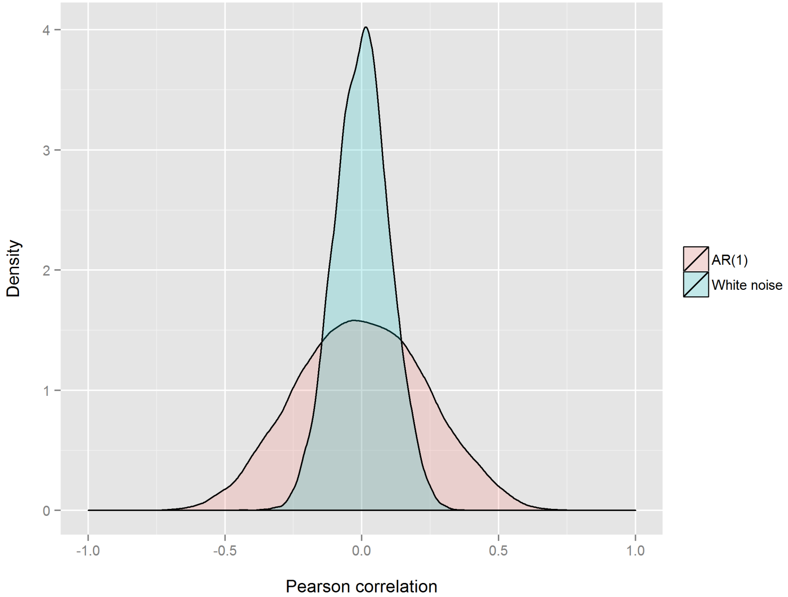


Fig. 4.3 Pairs of time series processes were generated 10,000 times, with $n = 100$ observations for each. Each time, the Pearson correlation between the two processes was calculated. When both pairs were white noise, i.e. iid $\sim \mathcal{N}(0,1)$, 95% of the correlations were between -0.2 and 0.2. But when the two pairs were independently generated AR(1) processes, with $x_t = 0.9x_{t-1} + \varepsilon_{x,t}$ and $y_t = 0.8y_{t-1} + \varepsilon_{y,t}$, only 60% of the correlations were between -0.2 and 0.2. This is an example of the way that temporal autocorrelation can bias an independence test (in this case, linear independence tested with Pearson correlation) that assumes iid data: many more correlations are significant than we would expect by chance. A simple correction is to first pre-whiten x_t and y_t by fitting an AR(1) model and obtaining residuals.

where $\varepsilon \sim \mathcal{N}(0,\sigma^2)$. (Note that this is equivalent to the Gaussian likelihood observation model introduced in the previous section. We have introduced ε as we explicitly care about the residuals.) Thus we want an estimator:

$$\hat{f} = E[f|(x_1,s_1),\ldots,(x_n,s_n)]$$

so that we can obtain residuals $\varepsilon_{xs} = X - \hat{f}(S)$. Notice that because we are using GP regression, all of our observations $(x_1,s_1)\ldots,(x_n,s_n)$ are used to estimate f and we explicitly account for the non-iid nature of our spatial or temporal observations by learning f. An intuitive way to think about this is as smoothing. All of the observations play a role in our posterior prediction of f at location a new or existing location s^* as seen in the algebra: our posterior prediction at location s^* is given by $E[f(s^*)|s^*,X,S] = K_*(K+\sigma^2 I)^{-1}X$. Note further that while for small samples some residual dependence may remain, we have a consistent method so as our sample size increases this dependence will go to 0. We discuss consistency and convergence in Section 4.10.

Because we consider S to be an environmental variable, we make the assumption of independent, additive noise. In other words, if S is a cause of X, we assume $X = f(S) + \varepsilon_{xs}$ with $S \perp\!\!\!\perp \varepsilon_{xs}$. Similarly, if S is a cause of Y, we assume $Y = f(S) + \varepsilon_{ys}$ with $S \perp\!\!\!\perp \varepsilon_{ys}$. Notice that we are not restricting ourselves to deterministic functions f. Any time series model, such as an autoregressive time series, with additive errors fits these requirements. Thus we can use ε_{xs} and ε_{ys} in subsequent independence tests, continuing to assume independent, additive noise for causes, and Markov and faithfulness, without worrying about bias due to an underlying correlation structure. This pre-whitening process, which follows standard practice in the spatial statistics and time series literature (e.g. Frisch and Waugh (1933); Haugh (1976)), is illustrated in Figure 4.4 using the same setup described above, where X and Y are independent AR(1) time series. We choose a particular realization with a large (but spurious) correlation of 0.61 between X and Y and a correspondingly highly significant value from HSIC ($p \leq 7.7 \times 10^{-18}$) for rejecting the null hypothesis of independence. We apply GP regression to X and Y separately as shown in Figure 4.4 to estimate pre-whitened residuals ε_X and ε_Y. These residuals have a very low correlation of 0.01 and a correspondingly insignificant p-value from HSIC of 0.40[1].

The use of GPs for pre-whitening is very powerful because GPs are a consistent non-parametric regression method, as discussed in Section 4.10. However, standard GP algorithms are computational-time and memory intensive, and thus not scalable to large datasets. I address this issue, especially in the context of spatiotemporal data, in the next section.

4.3 Scaling to large-scale datasets[2]

We are given a dataset $\mathcal{D} = (\mathbf{y}, X)$ where $\mathbf{y} = \{y_1, \ldots, y_n\}$ are the outcome variables and $X = \{x_1, \ldots, x_n\}$ are the predictor variables. The outcomes could be real-valued, categorical, counts, etc., and the predictors, for example, could be spatial locations, times, and other covariates.

[1] We note that the correct choice of kernel and method for obtaining residuals matters. This issue is discussed in more detail in Section 2.2. We used a Gaussian RBF kernel and obtained residuals by smoothing. If we had been more concerned with trying to exactly mimic the behavior of a classical autoregressive fit, we would instead need to use the Ornstein-Uhlenbeck process, which is a GP with exponential kernel given by $k(t,t') = \frac{1}{1-\phi^2} \exp(\log(\phi)|t-t'|)$ where $\phi = 0.9$ for x and $\phi = 0.8$ for y, and we would also have performed one-step-ahead forecasting rather than smoothing in order to obtain the residuals. Ultimately, if the practitioner has domain knowledge supporting the use of a particular class of models, such as AR(1), we would absolutely recommend incorporating this knowledge, rather than relying on a generic choice like GP regression with a Gaussian RBF kernel. We advocate GP regression as a generally applicable method, especially in cases for which there is little domain expertise, and we further advocate carefully checking residuals for structure and refining one's modeling choices accordingly.

[2] This presentation follows that of Flaxman et al. (2015b).

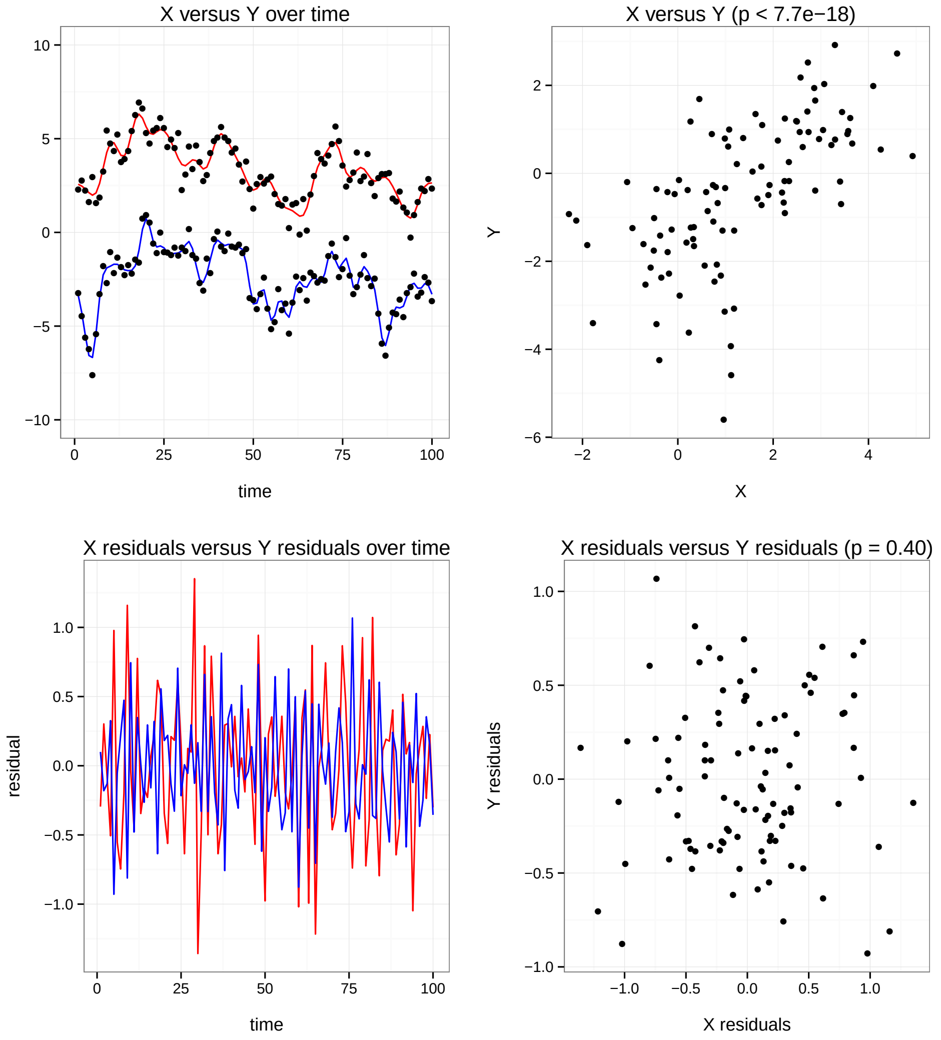


Fig. 4.4 X is a realization of an AR(1) process with $x_t = 0.9 \cdot x_{t-1} + \varepsilon_{x,t}$ and Y is a realization of an AR(1) process with $y_t = 0.8 \cdot y_{t-1} + \varepsilon_{y,t}$. X and Y are independent and $\varepsilon_{x,t}, \varepsilon_{y,t} \sim \mathcal{N}(0,1)$. As shown in Figure 4.3 it is likely that there will be a spurious correlation between X and Y. We chose a specific realization, plotted as the black dots in the top left plot (for visual clarity, $X+2$ and $Y-2$ are shown), in which the correlation is 0.61 with highly significant p-value from HSIC $\leq 7.7 \times 10^{-18}$ (top right plot). We used GP regression with an RBF covariance function to obtain the fitted curves shown in red and blue in the top left. The residuals are shown in the bottom left and compared in the bottom right: the correlation between the residuals is 0.01 with insignificant p-value from HSIC $= 0.40$.

We assume the relationship between the predictors and outcomes is determined by a latent Gaussian process $f(x) \sim \mathcal{GP}(m, k_\theta)$, and a likelihood for the observation model $p(y(x)|f(x))$. As introduced in Section 4.1, the GP is defined by its mean m and covariance function k_θ

(parametrized by $\boldsymbol{\theta}$), such that any collection of function values $\boldsymbol{f} = f(X) \sim \mathcal{N}(\boldsymbol{\mu}, K)$ has a Gaussian distribution with mean $\boldsymbol{\mu}_i = m(x_i)$ and covariance matrix $K_{ij} = k(x_i, x_j | \boldsymbol{\theta})$.

Our goal is to infer the predictive distribution $p(f_* | \mathbf{y}, x_*)$, for any test input x_*, which allows us to sample from $p(\mathbf{y}_* | \mathbf{y}, x_*)$ via the observation model $p(y(x)|f(x))$:

$$p(f_* | \mathcal{D}, x_*, \boldsymbol{\theta}) = \int p(f_* | \mathbf{X}, x_*, \boldsymbol{f}, \boldsymbol{\theta}) p(\boldsymbol{f} | \mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{f} \tag{4.6}$$

We also wish to infer the *marginal likelihood* of the data, conditioned only on kernel hyperparameters $\boldsymbol{\theta}$,

$$p(\mathbf{y} | \boldsymbol{\theta}) = \int p(\mathbf{y} | \boldsymbol{f}) p(\boldsymbol{f} | \boldsymbol{\theta}) d\boldsymbol{f}, \tag{4.7}$$

so that we can optimize this likelihood, or use it to infer $p(\boldsymbol{\theta} | \mathbf{y})$, for kernel learning. Having an expression for the marginal likelihood is particularly useful for kernel learning, because it allows one to bypass the extremely strong dependencies between $\boldsymbol{f}$ and $\boldsymbol{\theta}$ in trying to learn $\boldsymbol{\theta}$. Unfortunately, for all but the Gaussian likelihood (used for standard GP regression), where $p(\mathbf{y} | \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}, \Sigma)$, equations (7.13) and (4.7) are analytically intractable.

4.4 A motivating example: Cox Processes

In this section, we describe the log-Gaussian Cox Process (LGCP), a particularly important spatial statistics model for point process data (Diggle et al., 2013; Møller et al., 1998). While the LGCP is a general model, its use has been limited to small datasets. We focus on this model because of its importance in spatial statistics and its suitability for the Kronecker methods we propose. Note, however, that our methods are generally applicable to Gaussian process models with non-Gaussian likelihoods, such as Gaussian process classification.

An LGCP is a Cox process (inhomogeneous Poisson process with stochastic intensity) driven by a latent log intensity function $\log \lambda := f$ with a GP prior:

$$f(s) \sim \mathcal{GP}(\mu(s), k_\theta(\cdot, \cdot)) . \tag{4.8}$$

Conditional on a realization of the intensity function, the number of points in a given space-time region S is:

$$y_S | \lambda(s) \sim \text{Poisson}\left(\int_{s \in S} \lambda(s)\, ds \right) . \tag{4.9}$$

Following a common approach in spatial statistics, we introduce a "computational grid" (Diggle et al., 2013) on the observation window and represent each grid cell with its centroid, $s_1, \ldots, s_n$.

Let the count of points inside grid cell i be y_i. Thus our model is a Gaussian process with a Poisson observation model and exponential link function:

$$y_i | f(s_i) \sim \text{Poisson}\left(\exp[f(s_i)]\right) . \tag{4.10}$$

4.5 Laplace Approximation

For fixed covariance hyperparameters $\boldsymbol{\theta}$, we wish to infer the distribution over the log-intensity function f where $p(f|Y,S,\boldsymbol{\theta}) \propto p(Y|f)p(f|S,\boldsymbol{\theta})$. Since our observation model $p(Y|f)$ is a Poisson distribution, a closed form expression for $p(f|Y,S,\boldsymbol{\theta})$ is not available.

The Laplace approximation models the posterior distribution of the Gaussian process, $p(\boldsymbol{f}|\mathbf{y},X)$, as a Gaussian distribution, to provide analytic expressions for the predictive distribution and marginal likelihood in Eqs. (7.13) and (4.7). We follow the exposition in Rasmussen and Williams (2006).

Laplace's method uses a second order Taylor expansion to approximate the unnormalized log posterior,

$$\Psi(\boldsymbol{f}) := \log p(\boldsymbol{f}|\mathcal{D}) \overset{\text{const}}{=} \log p(\mathbf{y}|\boldsymbol{f}) + \log p(\boldsymbol{f}|X) , \tag{4.11}$$

centered at the $\hat{\boldsymbol{f}}$ which maximizes $\Psi(\boldsymbol{f})$. We have:

$$\nabla\Psi(\boldsymbol{f}) = \nabla \log p(\mathbf{y}|\boldsymbol{f}) - K^{-1}(\boldsymbol{f} - \boldsymbol{\mu}) \tag{4.12}$$

$$\nabla\nabla\Psi(\boldsymbol{f}) = \nabla\nabla \log p(\mathbf{y}|\boldsymbol{f}) - K^{-1} \tag{4.13}$$

$W := -\nabla\nabla \log p(\mathbf{y}|\boldsymbol{f})$ is an $n \times n$ diagonal matrix since the likelihood $p(\mathbf{y}|\boldsymbol{f})$ factorizes as $\prod_i p(y_i|f_i)$.

We use Newton's method to find $\hat{\boldsymbol{f}}$. The Newton update is

$$\boldsymbol{f}^{\text{new}} \leftarrow \boldsymbol{f}^{\text{old}} - (\nabla\nabla\Psi)^{-1}\nabla\Psi . \tag{4.14}$$

This optimization procedure naively requires $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ storage; standard practice is to compute a Cholesky decomposition of $\nabla\nabla\Psi$ to solve $(\nabla\nabla\Psi)^{-1}\nabla\Psi$.

Given $\hat{\boldsymbol{f}}$, the Laplace approximation for $p(\boldsymbol{f}|\mathbf{y})$ is given by a Gaussian:

$$p(\boldsymbol{f}|\mathbf{y}) \approx \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}}, (K^{-1} + W)^{-1}) . \tag{4.15}$$

Substituting the approximate posterior of Eq. (4.15) into Eq. (7.13), and defining $A = W^{-1} + K$, we find the approximate predictive distribution is (Rasmussen and Williams, 2006, p. 44):

$$p(f_*|\mathcal{D}, x_*, \boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{k}_*^\top \nabla \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}), k_{**} - \boldsymbol{k}_*^\top A^{-1} \boldsymbol{k}_*) \tag{4.16}$$

where $\boldsymbol{k}_* = [k(x_*, x_1), .., k(x_*, x_n)]^\top$ and $k_{**} = k(x_*, x_*)$.

This completes what we refer to as inference with a Gaussian process. We have so far assumed a fixed set of hyperparameters $\boldsymbol{\theta}$. For *learning*, we train these hyperparameters through marginal likelihood optimization. The Laplace approximate marginal likelihood is:

$$\log p(\boldsymbol{y}|X, \boldsymbol{\theta}) = \log \int \exp[\Psi(\boldsymbol{f})] d\boldsymbol{f} \tag{4.17}$$

$$\approx \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2} \boldsymbol{\alpha}^\top K^{-1} \boldsymbol{\alpha} - \frac{1}{2} \log |I + KW|, \tag{4.18}$$

where $\boldsymbol{\alpha} := K^{-1}(\hat{\boldsymbol{f}} - \boldsymbol{\mu})$. Standard practice is to find the $\hat{\boldsymbol{\theta}}$ which maximizes the approximate marginal likelihood of Eq. (4.18), and then condition on $\hat{\boldsymbol{\theta}}$ in Eq. (4.16) to perform inference and make predictions.

Learning and inference require solving linear systems and determinants with $n \times n$ matrices. This takes $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ storage, using standard approaches, e.g., Cholesky decomposition (Rasmussen and Williams, 2006).

4.6 Kronecker Methods

Kronecker approaches have recently been exploited in various GP settings (e.g., Bonilla et al., 2007; Finley et al., 2009; Riihimäki and Vehtari, 2014; Stegle et al., 2011). We briefly review Kronecker methods for efficient GPs, following Saatçi (2011), Gilboa et al. (2013), and Wilson et al. (2014), extending these methods to non-Gaussian likelihoods in the next section.

The key assumptions enabling the use of Kronecker methods is that the GP kernel is formed by a product of kernels across input dimensions and the inputs are on a Cartesian product grid (multidimensional lattice), $x \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_D$. (This grid need not be regular and the $\mathcal{X}_i$ can have different cardinalities.) Given these two assumptions, the covariance matrix K decomposes as a Kronecker product of covariance matrices $K = K_1 \otimes \cdots \otimes K_D$.

Saatçi (2011) shows that the computationally expensive steps in GP regression can be accelerated by exploiting Kronecker structure. Inference and learning require solving linear systems $K^{-1}v$ and computing log-determinants $\log |K|$. Typical approaches require $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space. Using Kronecker methods, these operations only require $\mathcal{O}(Dn^{\frac{D+1}{D}})$ operations and $\mathcal{O}(Dn^{\frac{2}{D}})$ storage, for n datapoints and D input dimensions. In Section 2.3 we presented the key Kronecker algebra results, including efficient matrix-vector multiplication and eigendecomposition.

Wilson et al. (2014) extend these efficient methods to partial grids, by augmenting the data with imaginary observations to form a complete grid, and then ignoring the effects of the imaginary observations using a special noise model in combination with linear conjugate gradients. Partial grids are common, and can be caused by, e.g., government boundaries, which interfere with grid structure.

4.7 Kronecker Methods for Non-Gaussian Likelihoods

We introduce our efficient Kronecker approach for Gaussian processes inference (Section 4.7.2) and learning (Section 4.7.3) with non-Gaussian likelihoods, after introducing some notation and transformations for numerical conditioning.

4.7.1 Numerical Conditioning

For numerical stability, we use the following transformations: $B = I + W^{1/2}KW^{1/2}$, $Q = W^{1/2}B^{-1}W^{1/2}$, $\boldsymbol{b} = W(\boldsymbol{f} - \boldsymbol{\mu}) + \nabla \log p(\boldsymbol{y}|f)$, and $\boldsymbol{a} = \boldsymbol{b} - QK\boldsymbol{b}$. Now $(K^{-1} + W)^{-1} = K - KQK$, from the matrix inversion lemma, and the Newton update in Eq. (4.14) becomes:

$$\boldsymbol{f}^{\text{new}} \leftarrow K\boldsymbol{a} \tag{4.19}$$

The predictive distribution in Eq. (4.16) becomes:

$$p(f_*|\mathcal{D}, x_*, \boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{k}_*^\top \nabla \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}), k_{**} - \boldsymbol{k}_*^\top Q \boldsymbol{k}_*) \tag{4.20}$$

4.7.2 Inference

Existing Kronecker methods for Gaussian likelihoods do not immediately apply to non-Gaussian likelihoods because we are no longer working solely with the covariance matrix K. We use linear conjugate gradients (LCG), an iterative method for solving linear systems which only involves matrix-vector products, to efficiently calculate the key steps of the inference algorithm in Section 4.5. Our full algorithm is shown in Algorithm 1. The Newton update step in Eq. (4.19) requires costly matrix-vector multiplications and inversions of $B = (I + W^{1/2}KW^{1/2})$.

We replace Eq. (4.19) with the following two steps:

$$Bz = W^{-1/2}\boldsymbol{b} \tag{4.21}$$
$$\boldsymbol{\alpha}^{\text{new}} = W^{1/2}z \tag{4.22}$$

For numerical stability, we follow (Rasmussen and Williams, 2006, p. 46) and apply our Newton updates to $\boldsymbol{\alpha}$ rather than $\boldsymbol{f}$. The variable $\boldsymbol{b} = W(\boldsymbol{f}-\boldsymbol{\mu}) + \nabla \log p(\boldsymbol{y}|\boldsymbol{f})$ can still be computed efficiently because W is diagonal, and Eq. (4.21) can be solved efficiently for $\boldsymbol{z}$ using LCG because matrix-vector products with B are efficient due to the diagonal and Kronecker structure.

The number of iterations required for convergence of LCG to within machine precision is in practice independent of n (the number of columns in B), and depends on the conditioning of B. Solving Eq. (4.21) requires $\mathcal{O}(Dn^{\frac{D+1}{D}})$ operations and $\mathcal{O}(Dn^{\frac{2}{D}})$ storage, which is the cost of matrix vector products with the Kronecker matrix K. No modifications are necessary to calculate the predictive distribution in Eq. (4.20). We can thus efficiently evaluate the approximate predictive distribution in $\mathcal{O}(mDn^{\frac{D+1}{D}})$ where $m \ll n$ is the number of Newton steps. For partial grids, we apply the extensions in Wilson et al. (2014) without modification.

4.7.3 Hyperparameter learning

To evaluate the marginal likelihood in Eq. (4.18), we must compute $\log|I+KW|$. Fiedler (1971) showed that for Hermitian positive semidefinite matrices U and V:

$$\prod_i (u_i + v_i) \leq |U+V| \leq \prod_i (u_i + v_{n-i+1}) \tag{4.23}$$

where $u_1 \leq u_2 \leq \ldots \leq u_n$ and $v_1 \leq \ldots \leq v_n$ are the eigenvalues of U and V. To apply this bound let $e_1 \leq e_2 \leq \ldots \leq e_n$ be the eigenvalues of K and $w_1 \leq w_2 \leq \ldots \leq w_n$ be the eigenvalues of W. Then we use that the eigenvalues of W^{-1} are $w_n^{-1} \leq w_{n-1}^{-1} \leq \ldots \leq w_1^{-1}$:

$$\begin{aligned}\log|I+KW| &= \log(|K+W^{-1}||W|)\\ &\leq \log \prod_i (e_i + w_i^{-1}) \prod_i w_i \\ &= \sum_i \log(1+e_i w_i)\end{aligned} \tag{4.24}$$

Putting this together with Equation (4.18) we have our bound on the Laplace approximation's log-marginal likelihood:

$$\log p(\boldsymbol{y}|X,\boldsymbol{\theta}) \geq \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2}\hat{\boldsymbol{\alpha}}^\top K^{-1}\hat{\boldsymbol{\alpha}} - \frac{1}{2}\sum_i \log(1+e_i w_i) \tag{4.25}$$

We chose the lower bound as we use non-linear conjugate gradients for our learning approach to find the best $\hat{\boldsymbol{\theta}}$ to maximize the approximate marginal likelihood. We approximate the necessary gradients using finite differences.

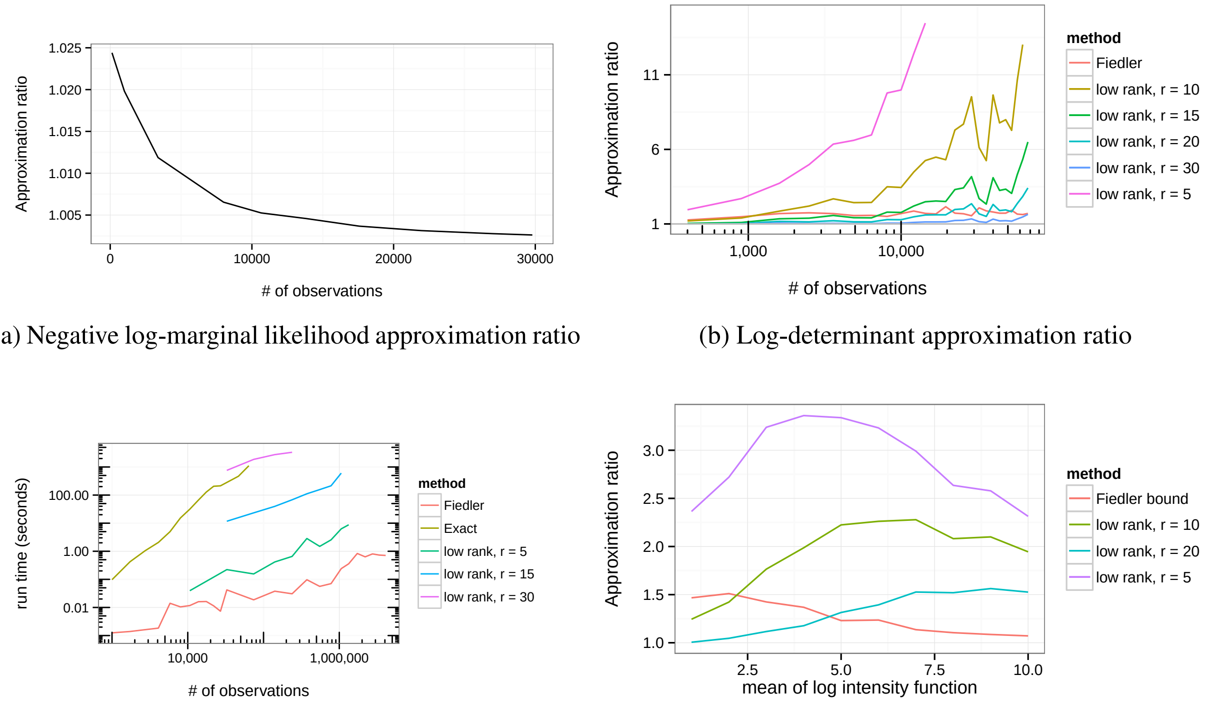
4.7.4 Evaluation of our Learning Approach

The bound we used on the Laplace approximation's log-marginal likelihood has been shown to be the closest possible bound on $|U+V|$ in terms of the eigenvalues of Hermitian positive semidefinite U and V (Fiedler, 1971), and has been used for heteroscedastic regression (Gilboa et al., 2014). However, its most appealing quality is computational efficiency. We efficiently find the eigendecomposition of K using standard Kronecker methods, where we calculate the eigenvalues of $K_1,\ldots,K_D$, each in time $\mathcal{O}(n^{\frac{3}{D}})$. We immediately know the eigenvalues of W because it is diagonal. Putting this together, the time complexity of computing this bound is $\mathcal{O}(Dn^{\frac{3}{D}})$. The log-determinant is recalculated many times during hyperparameter learning, so its time complexity is quite important to scalable methods.[3]

As shown in Figure 4.5a, as the sample size increases the lower bound on the negative log marginal likelihood approaches the negative log marginal likelihood calculated with the true log determinant. This result makes perfect sense for our Bayesian model, because the log-determinant is a complexity penalty term defined by our prior, which becomes less influential with increasing datasizes compared to the data dependent model fit term, leading to an approximation ratio converging to 1.

Next, we compare the accuracy and run-time of our bound to a recently proposed (Groot et al., 2014) log-det approximation relying on a low-rank decomposition of K. In Figure 4.5b we generated synthetic data on an $\sqrt{n}\times\sqrt{n}$ grid and calculated the approximation ratio by dividing the approximate value $\log|I+KW|$ by the true value $\log|I+KW|$ calculated with the full matrix. Our bound always has an approximation ratio between 1 and 2, and it gets slightly worse as the number of observations increases. This contrasts with the low-rank approximation.

[3] An alternative would be to try to exactly compute the eigenvalues of $I+KW$ using LCG. But this would require performing at least n matrix-vector products, which could be computationally expensive. Note that this was not an issue in computing the Laplace predictive distribution, because LCG solves linear systems to within machine precision for $J \ll n$ iterations. Our approach, with the Fiedler bound, provides an approximation to the Laplace marginal likelihood, and a lower bound which we can optimize, at the cost of a single eigendecomposition of K, which is in fact more efficient than a single matrix vector product $B\boldsymbol{v}$.



(a) Negative log-marginal likelihood approximation ratio

(b) Log-determinant approximation ratio

(c) Log-determinant runtime

(d) Log-determinant approximation ratio

Fig. 4.5 We evaluate our bounds on the log determinant in Eq. (4.24) and the Laplace marginal likelihood in Eq. (4.25), compared to exact values and low rank approximations. In a), the approximation ratio is calculated as our bound (Fiedler) on the negative marginal likelihood divided by the Laplace negative marginal likelihood. In b) and d), the approximation ratios are calculated as a given approximation for the log-determinant divided by the exact log-determinant. In c) we compare the runtime of the various methods.

When the rank r is close to $\sqrt{n}$ the approximation ratio is reasonable, but quickly deteriorates as the sample size increases.

In Figure 4.5c we compare the running times of these methods, switching to a 3-dimensional grid. The exact method quickly becomes impractical. For a million observations, a rank-5 approximation takes 6 seconds, a rank-15 approximation takes 600 seconds, while our bound takes only 0.24 seconds. While we cannot compare to the true log-determinant, our bound is provably an upper bound, so the ratio between the low rank approximation and ours is a lower-bound on the true approximation ratio. Here the low-rank approximation ratio is at least 2.8 for the rank-15 approximation and at least 30 for the rank-5 approximation.

Finally, we know theoretically that Fiedler's bound is exact when the diagonal matrix W is equal to spherical noise $\sigma^2 I$, which is the case for a Gaussian observation model.[4] Since the Gaussian distribution is a good approximation to the Poisson distribution in the case of a large mean parameter, we evaluated our log-determinant bound while varying the prior mean $\boldsymbol{\mu}$ of $\boldsymbol{f}$ from 0 to 10. As shown in Figure 4.5d, for larger values of $\boldsymbol{\mu}$, our bound becomes more accurate. There is no reason to expect the same behavior from a low-rank approximation, and in fact the rank-20 approximation becomes worse as the mean of λ increases.

4.7.5 Algorithm Details and Analysis

For inference, our approach makes no further approximations in computing the Laplace predictive distribution, since LCG converges to within machine precision. Thus, unlike inducing points methods like FITC or approximate methods like Nyström, our approach to inference gives the same answer as if we used standard Cholesky methods.

Pseudocode for our algorithm is shown in Algorithm 1. Given $K_1, \ldots, K_D$ where each matrix is $n^{1/D} \times n^{1/D}$, line 2 takes $\mathcal{O}(Dn^{2/D})$. Line 5 repeatedly applies Equation (2.15), and matrix-vector multiplication $(\bigotimes K_d)\, v$ reduces to D matrix-matrix multiplications VK_j where V is a matrix with n entries total, reshaped to be $n^{\frac{D-1}{D}} \times n^{\frac{1}{D}}$. This matrix-matrix multiplication is $\mathcal{O}(n^{\frac{D-1}{D}} n^{\frac{1}{D}} n^{\frac{1}{D}}) = \mathcal{O}(n^{\frac{D+1}{D}})$ so the total run-time is $\mathcal{O}(Dn^{\frac{D+1}{D}})$. Line 7 is elementwise vector multiplication which is $\mathcal{O}(n)$. Line 8 is calculated with LCG as discussed in Section 4.7 and takes $\mathcal{O}(Dn^{\frac{D+1}{D}})$. Lines 4 through 12 comprise the Newton update. Newton's method typically takes a very small number of iterations $m \ll n$ to converge, so the overall run-time is $\mathcal{O}(mDn^{\frac{D+1}{D}})$. Line 13 requires D eigendecompositions of matrices $K_1, \ldots, K_D$ which takes time $\mathcal{O}(Dn^{\frac{3}{D}})$ as discussed in Section 4.7.4. Line 14 is elementwise vector multiplication and addition so it is $\mathcal{O}(n)$. Overall, the runtime is $\mathcal{O}(Dn^{\frac{D+1}{D}})$. There is no speedup for $D = 1$, and for $D > 1$ this is nearly linear time. This is much faster than the standard Cholesky approach which requires $\mathcal{O}(n^3)$ time. The memory requirements are given by the total number of entries in $K_1, \ldots K_p$: $\mathcal{O}(Dn^{\frac{2}{D}})$. This is smaller than the storage required for the n observations, so it is not a major factor. But it is worth noting because it is much less memory than required by the standard Cholesky approach of $\mathcal{O}(n^2)$ space.

[4] The entries of W are equal to the second derivative of the likelihood of the observation model, so in the case of the Poisson observation model with exponential link function, $W_{ii} = -\nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f}) = \exp[\hat{\boldsymbol{f}}_i]$.

ALGORITHM 1: Kronecker GP Inference and Learning

```
**Input:** $\boldsymbol{\theta}, \boldsymbol{\mu}, K, p(\boldsymbol{y}|\boldsymbol{f}), \boldsymbol{y}$
Construct $K_1, \ldots, K_D$
$\boldsymbol{\alpha} \leftarrow 0$
**repeat**
    $\boldsymbol{f} \leftarrow \boldsymbol{K\alpha} + \boldsymbol{\mu}$ { Eq. (2.15)}
    $W \leftarrow -\nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f})\{Diagonal\}$
    $\boldsymbol{b} \leftarrow W(\boldsymbol{f} - \boldsymbol{\mu}) + \nabla p(\boldsymbol{y}|\boldsymbol{f})$
    Solve $B\boldsymbol{z} = W^{-\frac{1}{2}}\boldsymbol{b}$ with CG    # Eq. (4.21)
    $\Delta\boldsymbol{\alpha} \leftarrow W^{\frac{1}{2}}\boldsymbol{z} - \boldsymbol{\alpha}$ {Eq. (4.22)}
    $\hat{\xi} \leftarrow \arg\min_{\xi} \Psi(\boldsymbol{\alpha} + \xi\Delta\boldsymbol{\alpha})$ {Line Search}
    $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \hat{\xi}\Delta\boldsymbol{\alpha}$ {Update}
**until** convergence of $\Psi$
$\boldsymbol{e} = \mathrm{eig}(K)$ {exploit Kronecker structure}
$Z \leftarrow \boldsymbol{\alpha}^{\top}(\boldsymbol{f} - \boldsymbol{\mu})/2 + \sum_i \log(1 + \boldsymbol{e}_i W_i)/2 - \log p(\boldsymbol{y}|\boldsymbol{f})$
**Output:** $\boldsymbol{f}, \boldsymbol{\alpha}, Z$
```

4.8 Model Specification

We propose to combine our fast Kronecker methods for non-Gaussian likelihoods, discussed in Section 4.7, with Cox processes, which we introduced in Section 4.4. We will use this model for crime rate forecasting in Section 7.4.

With large sample sizes but little prior information to guide the choice of appropriate covariance functions, we turn to a class of recently proposed expressive covariance functions called Spectral Mixture (SM) kernels (Wilson and Adams, 2013b). These kernels model the *spectral density* given by the Fourier transform of a stationary kernel ($k = k(\tau) = k(x - x')$) as a scale-location mixture of Gaussians. Since mixtures of Gaussians are dense in the set of all distribution functions and Bochner's theorem shows a deterministic relationship between spectral densities and stationary covariances, SM kernels can approximate any stationary covariance function to arbitrary precision. For 1D inputs z, and $\tau = z - z'$, an SM kernel with Q components has the form

$$k(\tau) = \sum_{q=1}^{Q} w_q \exp(-2\pi^2\tau^2 v_q)\cos(2\pi\tau\mu_q)\,. \tag{4.26}$$

w_q is the weight, $1/\mu_q$ is the period, and $1/\sqrt{v_q}$ is the length-scale associated with component q. In the spectral domain, μ_q and v_q are the mean and variance of the Gaussian for component q. Wilson et al. (2014) showed that a combination of Kronecker methods and spectral mixture kernels distinctly enables structure discovery on large multidimensional datasets – structure discovery that is not possible using other popular scalable approaches, due to the limiting approximations in these alternatives.

For our space-time data, in which locations s are labeled with coordinates (x,y,t), we specify the following separable form for our covariance function k_θ:

$$k_\theta((x,y,t),(x',y',t')) = k_x(x,x')k_y(y,y')k_t(t,t')$$

where k_x and k_y are Matérn-5/2 kernels for space and k_t is a spectral mixture kernel with $Q = 20$ components for time. We used Matérn-5/2 kernels because the spatial dimensions in this application vary smoothly, and the Matérn kernel is a popular choice for spatial data Stein (1999).

We also consider the negative binomial likelihood as an alternative to the Poisson likelihood. This is a common alternative choice for count data Hilbe (2011), especially in cases of overdispersion and we find that it has computational benefits. The GLM formulation of the negative binomial distribution has mean m and variance $m + \frac{m^2}{r}$. It approaches the Poisson distribution as $r \to \infty$.

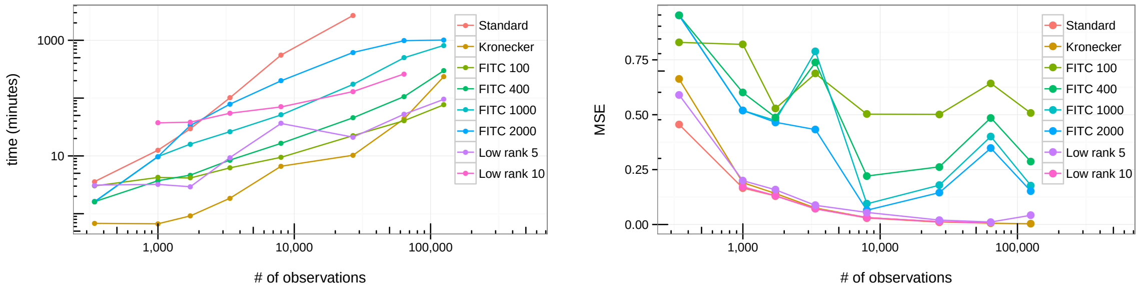
4.9 Experiments

We evaluate our methods on synthetic and real data, focusing on runtime and accuracy for inference and hyperparameter learning. Our methods are implemented in GPML (Rasmussen and Nickisch, 2010) and they are available with tutorials online[5]. We apply our methods to spatiotemporal crime rate forecasting, comparing with FITC, SSGPR (Lázaro-Gredilla et al., 2010), low rank Kronecker methods (Groot et al., 2014), and Kronecker methods with a Gaussian observation model.

4.9.1 Synthetic Data

To demonstrate the vast improvements in scalability offered by our method we simulated a realization from a GP on a grid of size $n \times n \times n$ with covariance function given by the product of three SM kernels. For each realization $f(s_i)$, we then drew $y_i \sim \text{NegativeBinomial}(\exp(f(s_i) + 1)$. Using this as training data, we ran non-linear conjugate gradients to learn the hyperparameters that maximized the lower bound on the approximate marginal likelihood in equation (4.25), using the same product of SM kernels. We initialized our hyperparameters by taking the true hyperparameter values and adding random noise. We compared our new Kronecker methods to

[5] www.gaussianprocess.org/gpml/code



(a) Run-time

(b) Accuracy

Fig. 4.6 Run-time and accuracy (mean squared error) of optimizing the hyperparameters of a GP with the Laplace approximation, comparing our new Kronecker inference methods to standard GP inference, FITC, and Kronecker with low rank. The standard method has cubic running time. Each experiment was run with 5-fold crossvalidation but error bars are not shown for legibility. There is no significant difference between the standard and Kronecker methods in terms of accuracy. For grids of size $10 \times 10 \times 10$ observations and greater, FITC has significantly lower accuracy than Kronecker and standard methods.

standard methods and FITC with varying numbers of inducing points. In each case, we used the Laplace approximation. We used 5-fold crossvalidation, relearning the hyperparameters for each fold and making predictions for the latent function values $\boldsymbol{f}_i$ on the 20% of data that was held out. The average MSE and running times for each method on each dataset are shown in Figure 4.6. We also calculated the log-likelihood of our posterior predictions for varying numbers of observations n for FITC-100, as shown in Table 4.3 in the Appendix. Our method achieved significantly higher predictive log-likelihood than FITC-100 for $n \geq 1000$.

In our final synthetic test, we simulated *100 million observations* from a GP on an 8 dimensional grid, possibly the largest dataset that has ever been modeled with a Gaussian process. This is particularly exceptional given the non-Gaussian likelihood. In this case, we had a simple covariance structure given by a squared exponential (RBF) kernel with different length-scales per dimension. We successfully evaluated the marginal likelihood in 27 minutes.

4.9.2 Crime Rate Forecasting in Chicago

The City of Chicago makes geocoded, date-stamped crime report data publicly available through its data portal[6]. For our application, we chose crimes coded as “assault” which includes all “unlawful attacks” with a weapon or otherwise. Assault has a marked seasonal pattern, peaking in the summer. We used a decade of data from January 1, 2004 to December 31, 2013, consisting of 233,088 reported incidents of assault. We trained our model on data from the

[6] http://data.cityofchicago.org

first 8 years of the dataset (2004-2011), and made forecasts for each week of 2012 and 2013. Forecasting this far into the future goes well beyond what is currently believed to be possible by practitioners.

LGCPs have been most widely applied in the 2-dimensional case, and we fit spatial LGCPs to the training data, discretizing our data into a 288×446 grid for a total of 128,448 observations. Posterior inference and learned hyperparameter are shown in Section 4.9.3 of the Appendix.

For our spatiotemporal forecasting, we used Spectral Mixture (SM) kernels for the time dimension, as discussed in Section 4.8. Specifically, we consider $Q = 20$ mixture components. For hyperparameter learning, our spatial grid was 17×26, corresponding to 1 mile by 1 mile grid cells, and our temporal grid was one cell per week, for a total of 416 weeks. Thus, our dataset of 233,088 assaults was discretized to a grid of size 183,872. Both of these sample sizes far exceed the state-of-the-art in fitting LGCPs, and indeed in fitting most GP regression problems without extreme simplifying assumptions or approximations.

To find a good starting set of hyperparameters, we used the hyperparameter initialization procedure in Wilson et al. (2014) with a Gaussian observation model. We also rescaled counts by the maximum count at that location, log-transformed, and then centered so that they would have mean 0. We ran non-linear conjugate gradient descent for 200 iterations. Using the hyperparameters learned from this stage, we switched to the count data and a negative binomial likelihood. We then ran non-linear conjugate gradient descent for another 200 iterations to relearn the hyperparameters and also the variance of the negative binomial.

The spatial hyperparameters that we learned are $\sigma^2 = 0.2231$, $\lambda_1 = 0.11$ and $\lambda_2 = 0.02$. This means that at this high resolution, with so much temporal data, there was little smoothing in space, with nearby locations allowed to be very different. Yet due to the multiplicative structure of our covariance function, our posterior inference is able to "borrow strength" such that locations with few observations follow a globally-learned time trend. We learned 60 temporal hyperparameters, and the spectral mixture components with the highest weights are shown in Figure 4.8, visualized in the covariance and frequency domains. We also show what posterior time series predictions would be if only a particular spectral component had been used, roughly giving an idea of the "explanatory" power of separate spectral components. We interpret the components, by decreasing weight, as follows: component 1 has a period and length-scale larger than the observation window thus picking up a decreasing trend over time. Components 2 (with period 1 month) and 4 pick up very-short-scale time variation, enabling the model to fit the observed data well. Component 3 picks up the yearly periodic trend (the spike in the spectral domain is at $0.02 = \frac{1}{52.1}$). Component 5 picks up a periodic trend with length longer than a year – 97 weeks, a feature for which we do not have any explanation. The exact hyperparameters are in Table 4.2 in the Appendix.

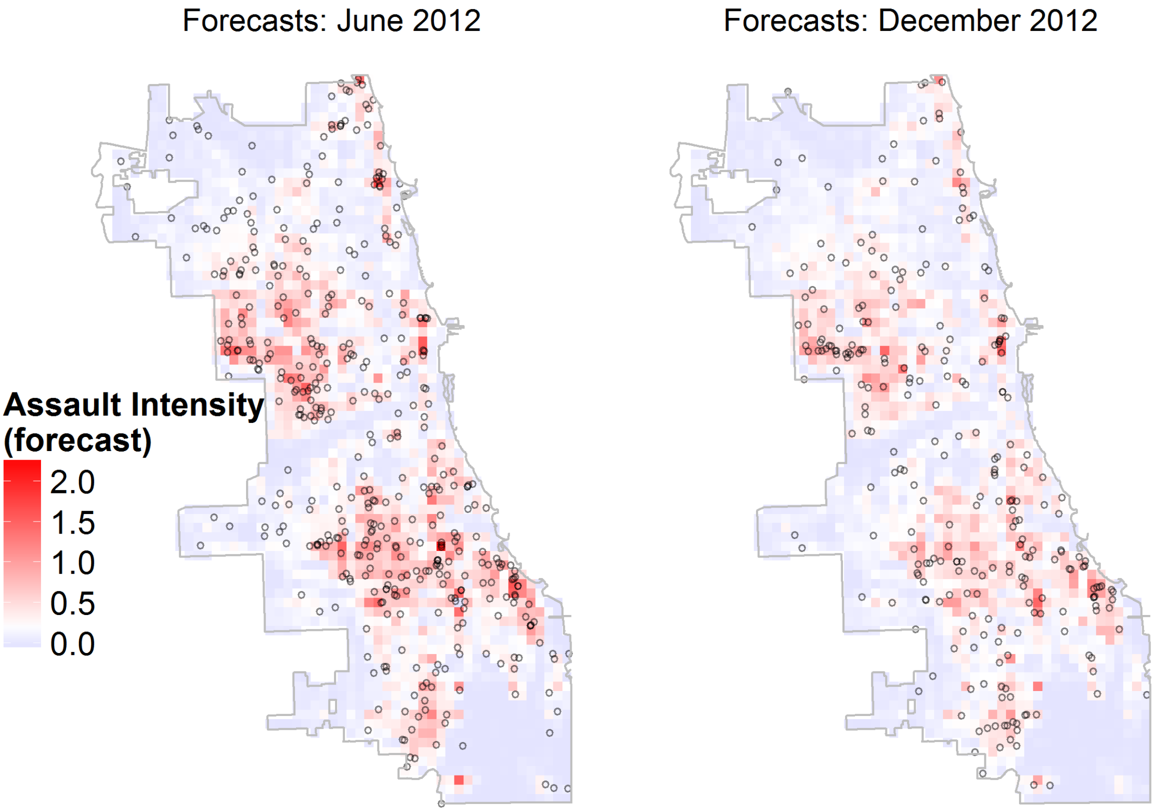


Fig. 4.7 Local area posterior forecasts of assault one year into the future with the actual locations of assaults shown as black dots. The model was fit to data from January 2004 to December 2011, and the forecasts were made for the first week of June 2012 (left) and December 2012 (right).

After learning the hyperparameters, we made predictions for the entire 8 years of training data and 2 years of forecasts. In Figure 4.10 in the Appendix we show the time series of assaults for 9 neighborhoods with our predictions, forecasts, and uncertainty intervals. Next, we rediscretized our original point pattern to a grid of size 51×78 ($n = 1.6$ million observations) and made spatial predictions 6 months and 1 year into the future, as shown in Figure 4.7, which also includes the observed point pattern of crimes. Visually, our forecasts are quite accurate. The accuracy and runtime of our method and competitors is shown in Table 4.1. The near 0 RMSE for predictions at the training data locations (i.e. the training error) for Kronecker Gaussian SM-20 indicates overfitting, while our model, Kronecker NegBinom SM-20, has a more reasonable RMSE of 0.79, out-performing the other models. The forecasting RMSE of our model was not significantly different than SSGPR or Kronecker Gaussian, while it outperformed FITC.

But RMSE does not take forecasting intervals (posterior uncertainty) into account. Kronecker Gaussian and SSGPR had overly precise posterior estimates. Forecast log-likelihood is the probability of the out-of-sample data (marginalizing out the model parameters), so we can use it to directly compare the models, where higher likelihoods are better. The Kronecker Gaussian approach has the lowest forecast log-likelihood. FITC was not overconfident, but its posterior forecasts were essentially constant. Our model has the highest forecast log-likelihood, showing a balance between a good fit and correct forecasting intervals. Kronecker Gaussian methods showed the fastest run-times due to the availability of a closed form posterior. FITC was very slow, even though we only used 100 inducing points.

	KronNB SM-20	KronNB SM-20 Low Rank	KronGauss SM-20	FITC-100 NB SM-20	SSGPR-200
Training RMSE	0.79	1.13	10^{-11}	2.14	1.45
Forecast RMSE	1.26	1.24	1.28	1.77	1.26
Forecast log-likelihood	-33,916	-172,879	-352,320	-42,897	-82,781
Run-time	2.8 hours	9 hours	22 min.	4.5 hours	2.8 hours

Table 4.1 Kron NB SM-20 (our method) uses Kronecker inference with a negative binomial observation model and an SM kernel with 20 components. KronNB SM-20 Low Rank uses a rank 5 approximation. KronGauss SM-20 uses a Gaussian observation model. FITC 100 uses the same observation model and kernel as KronNB SM-20 with 100 inducing points and FITC inference. SSGPR-200 uses a Gaussian observation model and 200 spectral points. Carrying forward the empirical mean and variance has a forecast RMSE of 1.84 and log-likelihood of -306,430.

q	Weight	Period	Length-scale
1	52.72	10813.9	133280.2
2	5.48	4.0	1.1
3	0.33	52.1	27700.8
4	0.05	22.0	1.6
5	0.02	97.4	7359.1

Table 4.2 The top five spectral mixture components learned for the temporal kernel in the LGCP fit to 8 years of assault data. The components are visualized in Figure 4.8 where component q corresponds to the row of the table.

4.9.3 A two-dimensional LGCP

We used a product of Matérn-5/2 kernels: $k_x(d)$ with length-scale λ_x and variance σ^2 and $k_y(d)$ with length-scale λ_y and variance fixed at 1: $k((x,y),(x',y')) = k_x(|x-x'|)k_y(|y-y'|)$.

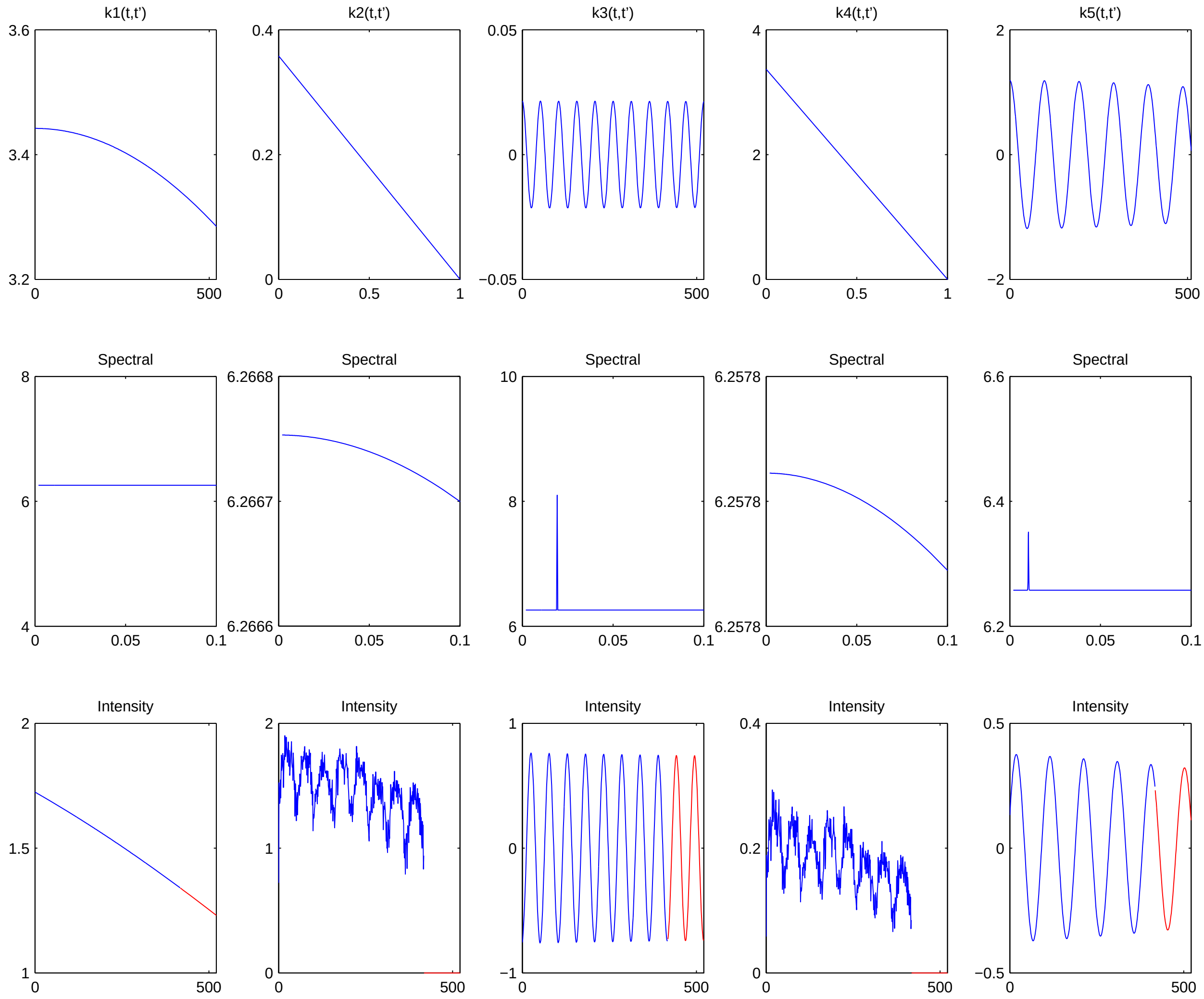


Fig. 4.8 The five spectral mixture components with highest weights learned by our model are shown as a covariance (top) and spectral density (middle). In the bottom row, time series predictions were made on the dataset (ignoring space) using only that component. Red indicates out-of-sample forecasts.

We discretized our data into a 288×446 grid for a total of 128,448 observations. Locations outside of the boundaries of Chicago – about 56% of the full grid—were treated as missing. In Figure 4.9 we show the location of assaults represented by dots, along with a map of our posterior intensity, log-intensity, and variance of the number of assaults. It is clear that our approach is smoothing the data. The hyperparameters that we learn are $\sigma^2 = 5.34$, $\lambda_x = 2.23$, and $\lambda_y = 2.24$, i.e., length-scales for moving north-south and east-west were found to be nearly identical for these data; by assuming Kronecker structure our learning happens in a fashion analogous to Automatic Relevance Determination Neal (1996).

N	Standard	Kronecker	FITC-100
125	-62.12	-61.52	-61.20
343	-157.47	-157.80	-159.21
1000	-445.48	-443.87	-455.84
1728	-739.56	-740.31	-756.95
8000	-3333.10	-3333.66	-3486.20

Table 4.3 Predictive log-likelihoods are shown corresponding to the experiment in Figure 4.6. A higher log-likelihood indicates a better fit. The differences between the standard and Kronecker results were not significant but the difference between FITC-100 and the others was significant (two-sample paired t-test, $p \leq .05$) for $n \geq 1000$.

4.10 Conclusion

We conclude this chapter by mentioning the literature on the consistency of GPs and its relevance to our pre-whitening procedure and our working scaling up GPs. Choi and Schervish (2007) demonstrate almost sure convergence for GP regression under mild conditions while Van Der Vaart and Van Zanten (2011) provide convergence rates for GP regression. The take-away is that given sufficient data, GP regression will uncover the true latent surface underlying the data. GP regression is thus a very interesting method because it is consistent *and* non-parametric. The promise of non-parametric methods is that their complexity grows with the size of the dataset and the promise of a consistent method is that it will converge with sufficient sample sizes. With plentiful data the challenge becomes to find efficient inference methods to realize this promise in practice. For the spatiotemporal models we considered above, our scalable methods (and previous scalable Kronecker methods for Gaussian observation models) enable routine GP inference and learning with very large datasets.

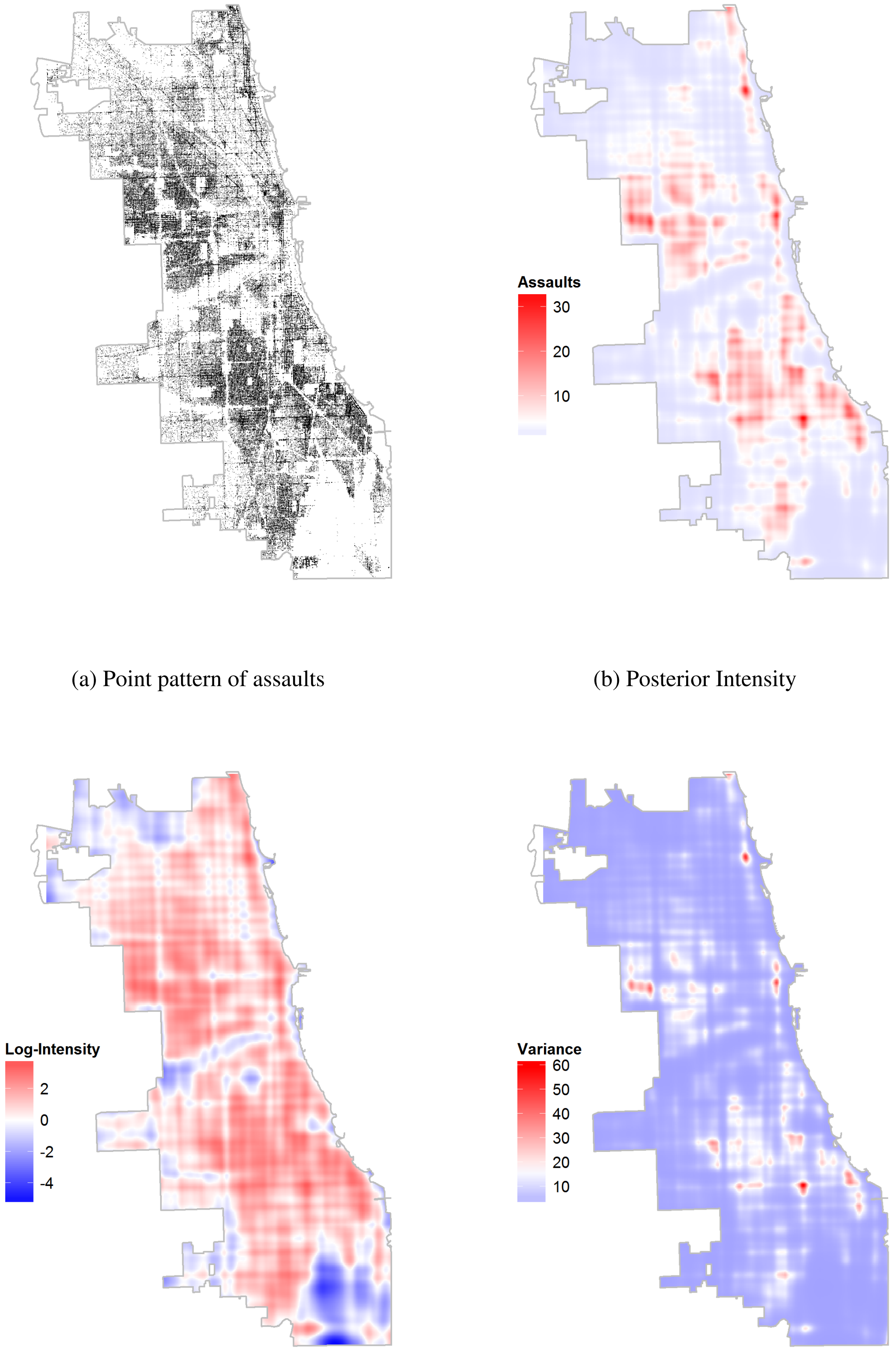
I prove convergence of the pre-whitening procedure using the result in Van Der Vaart and Van Zanten (2011), that there is some sequence $r_n \to 0$ for sample size n such that $\hat{f}$ converges to f with:

$$E_f \|\hat{f} - f\|_2^2 \leq r_n^2 \tag{4.27}$$

where the convergence rate of r_n depends on our choice of kernel, but under a variety of conditions given in Van Der Vaart and Van Zanten (2011) we are guaranteed that it decreases to 0 in n. Since residuals are given by the vector $\hat{f} - f$, we would like a bound on the covariance off the diagonal, i.e. $\mathrm{Cov}((\hat{f} - f)_i, (\hat{f} - f)_j)$, for all i and j. This is bounded above by the covariance on the diagonal:

$$
\begin{aligned}
&\text{Cov}((\hat{f}-f)_i,(\hat{f}-f)_j) && (4.28)\\
&\leq Var((\hat{f}-f)_i) && (4.29)\\
&\leq E[(\hat{f}-f)_i^2]-E[(\hat{f}-f)_i]^2 && (4.30)\\
&\leq E[(\hat{f}-f)_i^2] && (4.31)\\
&\leq r_n^2 && (4.32)
\end{aligned}
$$

The last step follows because the sum of the squared residuals are $\leq r_n^2$ by Eq. (4.27), so any particular squared residual is also $\leq r_n^2$.



(a) Point pattern of assaults

(b) Posterior Intensity

(c) Posterior Latent Log-Intensity

(d) Posterior Variance

Fig. 4.9 We fit a log Gaussian Cox Process to the point pattern of reported incidents of assault in Chicago (a) and made posterior estimates of the intensity surface (b). The latent log-intensity surface is visualized in (c) and the posterior variance is visualized in (d).

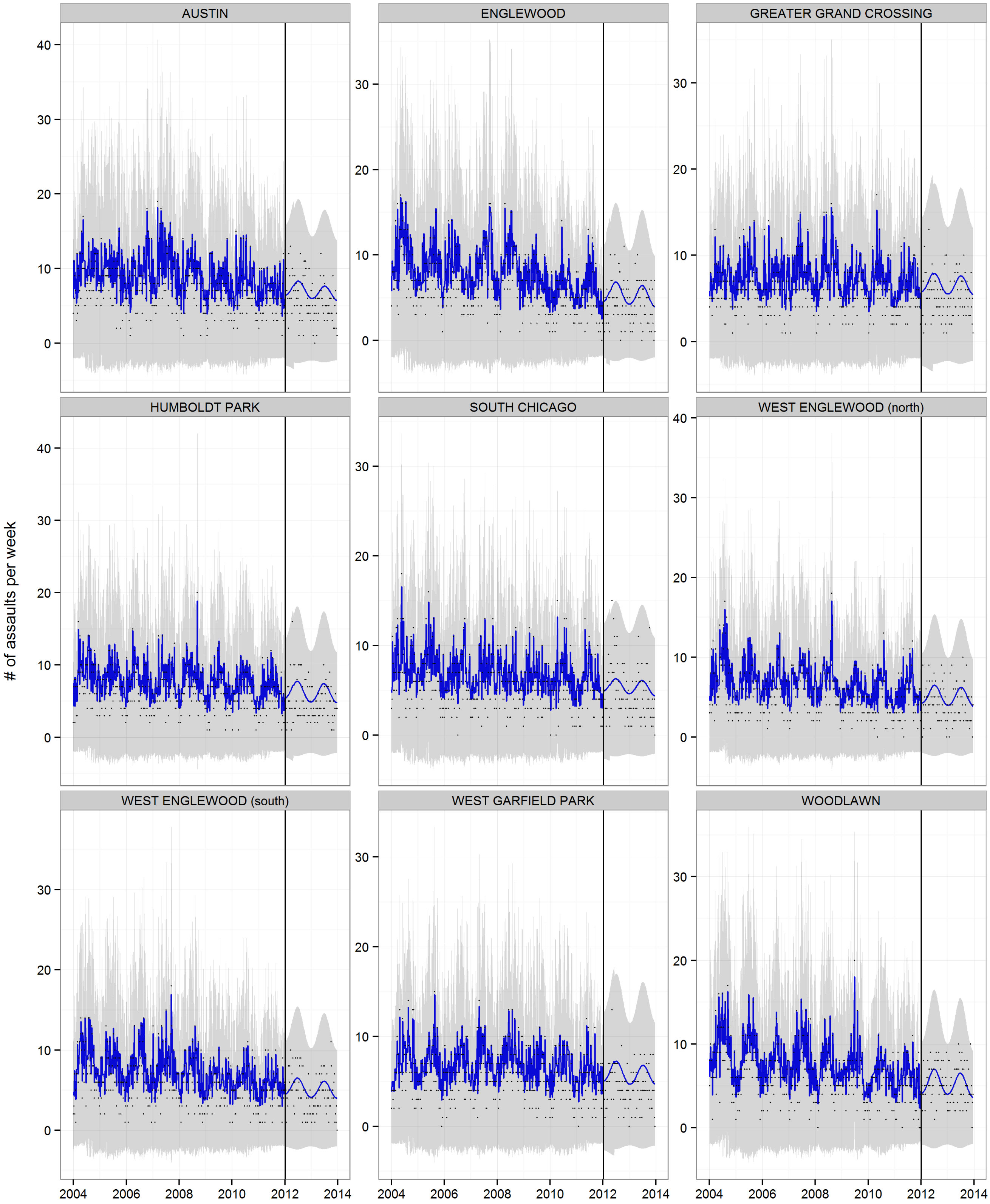


Fig. 4.10 We show the time series of weekly assaults in the nine neighborhoods with the most assaults in Chicago. The blue line shows our posterior prediction (training data, first 8 years of data) and forecast (out-of-sample, last 2 years of data, to the right of the vertical bar). Observed counts are shown as dots. 95% posterior intervals are shown in gray.

Chapter 5

Ecological inference[1]

In this chapter, I present a new solution to the "ecological inference" problem, of learning individual-level associations from aggregate data. This problem has a long history and has attracted much attention, debate, claims that it is unsolvable, and purported solutions. Unlike other ecological inference techniques, my method makes use of unlabeled individual-level data by embedding the distribution over these predictors into Hilbert space using the kernel mean embeddings introduced in Section 2.4. and recent learning theory results for *distribution regression*. Unlike previous approaches, my novel approach to distribution regression exploits the connection between Gaussian process regression and kernel ridge regression, giving a coherent, Bayesian approach to learning and inference and a convenient way to include spatial information in the specification of the model. My approach is highly scalable as it relies on FastFood, a randomized explicit feature representation for kernel embeddings introduced in Section 2.6.

I apply my approach to the challenging political science problem of modeling the voting behavior of demographic groups based on aggregate voting data. We consider the 2012 US Presidential election, and ask: what was the probability that members of various demographic groups supported Barack Obama, and how did this vary spatially across the country? My results match standard survey-based exit polling data for the small number of states for which it is available, and serve to fill in the large gaps in this data, at a much higher degree of granularity.

5.1 Introduction

I start by giving an example of the ecological inference problem. The name ecological refers to the idea of ecological correlations (Robinson, 1950), that is correlations between variables

[1] This chapter is drawn from Flaxman et al. (2015c)

observed for a group of individuals, as opposed to individual correlations, where the individuals are the unit of analysis. The ecological inference problem has much in common with the "modifiable areal unit problem" (Openshaw, 1984) and Simpson's paradox. Simply put, it is the problem of inferring individual correlations from ecological correlations. One way to understand the reason it is called a "problem" is to consider a two-by-two contingency table, with unknown entries inside the table, and known marginals. As shown in the contingency table below, we might know that a certain electoral district's voting population is 43% men and 57% women and that in the last election, the outcome was 63% in favor of the Democratic candidate and 37% in favor of the Republican candidate. These percentages correspond to the numbers of individuals shown below:

	Democrat	Republican	
Men	?	?	1,500
Women	?	?	2,000
	2,200	1,300	

Is it possible to infer the joint and thus conditional probabilities, for example can we ask, what was the Democratic candidate's vote share among women voters? It is clear that only very loose bounds can be placed on these probabilities without any more information. Based on the fact that rows and columns must sum to their marginals, we know, e.g. that the number of Democrats who are men is between 0 and 1,500. These types of deterministic bounds have been around since the 1950's, under the name the method of bounds (Duncan and B, 1953).

What if we are given a set of electoral districts, where for each we know the marginals of the two-by-two contingency table, but none of the inner entries? Then, thinking statistically, we might be tempted to run a regression, predicting the electoral outcomes based on the gender breakdowns of the districts. But this approach, formalized as Goodman's method (Goodman, 1959) a few years after the method of bounds was proposed, can easily lead us astray—there is not even a guarantee that outcomes be bounded between 0 and 1, and it ignores potentially useful information provided by deterministic bounds.

The ecological inference problem has a long history of solutions, counter-solutions, and it is often taught with a note of grave caution and stark warnings that ecological inference is to be avoided at all costs, usually in favor of individual-level surveys. As with Simpson's paradox, it should come as no surprise that correlations at one level of aggregation can and do flip signs at other levels of aggregation. But abandoning all attempts at ecological inference in favor of surveys is not feasible or appropriate in many circumstances—relevant respondents are no longer alive to answer historical questions of interest; subjects are reluctant to answer questions about sensitive topics like drug usage or cheating—meaning social scientists have been hard-pressed and even discouraged from studying many interesting and important questions. Ecological inference problems appear in demography, sociology, geography, and political science, and—as discussed in King (1997)—landmark legislation in the US such as the Voting Rights Act requires a solution to the ecological inference problem to understand racial voting patterns[2].

This problem has attracted a variety of approaches over the years as summarized in King (1997), which also proposes a Bayesian statistical modeling framework incorporating the method of bounds (thus uniting the deterministic and probabilistic approaches). King (1997) sparked a renewed interest in ecological inference, much of which is summarized in King et al. (2004). A parametric Bayesian approach to this setting was proposed in Jackson et al. (2006) and a semiparametric approach was proposed in Prentice and Sheppard (1995).

My method differs from existing methods in four ways. First, it uses more information than is typically considered in a standard ecological regression setting: I assume that we have access to representative *unlabeled* individual-level data. In the voting example, this means having a sample of individual-level census records ("microdata") about each electoral district. Second, my method incorporates spatial variation. Spatial data is a common feature of ecological regressions Third, while my method may be applied to the difficult ecological inference problem of making individual level predictions from aggregate data, I propose that it is most well-suited to a related ecological problem, common in political science: inferring the unobserved behavior of subgroups of a population based on the aggregate behavior of the groups of which they are part. For my application, this means inferring the voting behavior of men and women separately by electoral district, given aggregate voting information by district. Finally, my work is nonparametric. Kernel embeddings are used to capture all moments of the probability distribution over covariates, and Gaussian process regression is used to non-parametrically model the dependence between predictors and labels.

A related line of work, "learning from label proportions" (Kueck and de Freitas, 2005; Patrini et al., 2014; Quadrianto et al., 2009), has the individual-level goal in mind, and aims to build a classifier for individual instances based only on group level label proportions. While in principle, this approach could be used in my setting, since we are only interested in subgroup level predictions the extra task of estimating individual level predictions is probably not worth the effort considering we are working with $n = 10$ million individuals. Note that there is also a relevant parallel literature on data privacy and de-anonymization, e.g. Narayanan and Shmatikov (2009).

[2] Long-standing solutions have proved quite inadequate: in one court case involving the Voting Rights Act, a qualified expert testified, based on Goodman's method, that the percentage of blacks who were registered to vote in a certain electoral district exceeded 100% (King, 1997). This evidently false claim was apparently made earnestly.

My method is based on recent advances in distribution regression (Gärtner et al., 2002; Szabo et al., 2015), which I generalize to address the ecological inference case. Previous work on distribution regression has relied on kernel ridge regression, but I use Gaussian process regression instead, thus enabling me to incorporate spatial variation, learn kernel hyperparameters, and provide posterior uncertainty intervals, all in a fully Bayesian setting. For scalability (my experiments use $n = 10$ million individuals), I use a randomized explicit feature representation ("FastFood") (Le et al., 2013) rather than the kernel trick.

I provide the necessary background on distribution regression in Section 5.2. I formalize the ecological inference problem in Section 5.3 and propose my method in Section 5.4. I apply it to the case of the 2012 US presidential election in Section 7.4, comparing my results to survey-based exit polls.

5.2 Background: distribution regression

In this section, I present distribution regression, the task of learning a classifier or a regression function that maps probability distributions to labels. The problem is fundamentally challenging because we only observe the probability distributions through groups of samples from these distributions. Specifically, our dataset is structured as follows:

$$\left(\{x_1^j\}_{j=1}^{N_1}, y_1\right), \left(\{x_2^j\}_{j=1}^{N_2}, y_2\right), \ldots \left(\{x_n^j\}_{j=1}^{N_n}, y_n\right) \tag{5.1}$$

where group i has a single real-valued label y_i and N_i individual observations (e.g. demographic covariates for N_i individuals) denoted $x_i^j \in \mathbb{R}^d$.

To admit a theoretical analysis, it is assumed that the probability distributions themselves are drawn randomly from some unknown meta distribution of probability distributions. The intuition behind why distribution regression is possible is that if each group of samples are iid draws from a distribution which is itself an iid drawn from the meta distribution, then we will be able to learn.

Recently, this "two-stage sampled" structure was analyzed, showing that a ridge regression estimator is consistent (Szabo et al., 2015) with polynomial rate of convergence for almost any meta-distribution of distributions that are sufficiently smooth. We use the obvious empirical estimator of the kernel mean embedding introduced in Section 2.4:

$$\widehat{\mu_X} = \frac{1}{N} \sum_j \phi(x^j) \tag{5.2}$$

The basic approach to distribution regression is as follows: use this kernel mean estimator for each group separately to estimate:

$$\widehat{\mu_1} = \frac{1}{N_1}\sum_{j=1}^{N_1}\phi(x_1^j), \quad \ldots, \quad \widehat{\mu_n} = \frac{1}{N_n}\sum_{j=1}^{N_n}\phi(x_n^j) \tag{5.3}$$

Next, use kernel ridge regression (Saunders et al., 1998) to learn a function f:

$$y = f(\widehat{\mu}) + \varepsilon \tag{5.4}$$

where the objective is to minimize the L_2 loss subject to a "ridge" complexity penalty weighted by a positive constant λ:

$$\hat{f} = \text{arg min}_{f\in\mathcal{H}_f}\sum_i [y_i - f(\widehat{\mu_i})]^2 + \lambda\|f\|^2_{\mathcal{H}_f} \tag{5.5}$$

In Szabo et al. (2015) a variety of kernels for f corresponding to the Hilbert space $\mathcal{H}_f$ are considered. We follow the simplest choice of the linear kernel $k(\widehat{\mu_i},\widehat{\mu_j}) = \langle\widehat{\mu_i},\widehat{\mu_j}\rangle$, motivated by the fact that we are already working in Hilbert space. (Thus, we could equivalently say that we are simply using ridge regression!) Following the standard derivation of kernel ridge regression (Saunders et al., 1998), we can find the function f in closed form for a new test group μ_*:

$$f(\mu_*) = k^*(K+\lambda I)^{-1}[y_1,\ldots,y_n]^T \tag{5.6}$$

where $k^* = [\langle\widehat{\mu_1},\mu_*\rangle,\ldots,\langle\widehat{\mu_n},\mu_*\rangle]$ and $K_{ab} = \langle\widehat{\mu_a},\widehat{\mu_b}\rangle$.

Naively implementing distribution regression using the kernel trick is not scalable in the setting I consider: to compute just one entry in K requires computing $K_{ab} = \langle\widehat{\mu_a},\widehat{\mu_b}\rangle = \frac{1}{N_aN_b}\sum_{j_1 j_2} k(x_a^{j_1},x_b^{j_2})$. This computation is $O(N^2)$ (where we assume for simplicity $N_i = N$, $\forall i$) so computing K is $O(n^2N^2)$. In my application, $N \approx 10^4$, so I need a much more scalable approach. Since we ultimately only need to work with the mean embeddings μ_i rather than the individual observations x_i^j, we use the explicit feature representation introduced in Section 2.6 to drastically reduce our computational costs.

Gaussian process regression was presented in Chapter 4. I review the well-known connection between the posterior mean in GP regression and the kernel ridge regression estimator of Equation (5.6). If we wish to make a prediction at a new location s^*, the standard predictive equations for GP regression (Rasmussen and Williams, 2006), derived by conditioning a multivariate Gaussian distribution, tell us that:

$$y^* \mid s^*, X, \mathbf{y} \sim \mathcal{N}(k^*(K+\sigma^2 I)^{-1}\mathbf{y}, k^{**} - k^*(K+\sigma^2 I)^{-1}k^{*\top}) \tag{5.7}$$

We can immediately see the connection between the kernel ridge regression estimator in Equation (5.6) and the posterior mean of the GP in Equation (5.7). (A superficial difference is that in Equation (5.6) our predictors are $\widehat{\mu}_i$ while in Equation (5.7) they are generic locations s_i, but this difference will go away in Section 5.4 when I propose using GP regression for distribution regression.) The predictive mean of GP regression is exactly equal to the kernel ridge regression estimator, with σ^2 corresponding to λ. In ridge regression, a larger penalty λ leads to a smoother fit (equivalently, less overfitting), while in GP regression a larger σ^2 favors a smoother GP posterior because it implies more measurement error. For a full discussion of the connections see (Cristianini and Shawe-Taylor, 2000, Sections 6.2.2-6.2.3).

5.3 Ecological Inference

In this section I state the ecological inference problem that I intend to solve. I use the motivating example of inferring Barack Obama's vote share by demographic subgroup (e.g. men versus women) in the 2012 US presidential election, without access to any individual-level labels. Vote totals by electoral precinct are publicly available, and these provide the labels in our problem. Predictors are in the form of demographic covariates about individuals (e.g. from a survey with individual level data like the census). The challenge is that the labels are aggregate, so it is impossible to know which candidate was selected by any particular individual. This explains the terminology: "ecological correlations" are correlations between variables which are only available as aggregates at the group level (Robinson, 1950)

We use the same notation as in Section 5.2. Let $x_i^j \in \mathbb{R}^d$ be a vector of covariates for individual i in region j. Let w_i^j be survey weights[3]. Let y_i be labels in the form of two-dimensional vectors (k_i, n_i) where k_i is the number of votes received by Obama out of n_i total votes in region i. Then our dataset is:

$$\left(\{x_1^j\}_{j=1}^{N_1}, y_1\right), \left(\{x_2^j\}_{j=1}^{N_2}, y_2\right), \ldots, \left(\{x_n^j\}_{j=1}^{N_n}, y_n\right) \tag{5.8}$$

We will typically have a rich set of covariates available, in addition to the demographic variables we are interested in stratifying on, so the x_i^j will be high-dimensional vectors denoting gender, age, income, education, etc.

Our task is to learn a function f from a demographic subgroup (which could be everyone) within region i to the probability that this demographic subgroup supported Obama, i.e. the number of votes this group gave Obama divided by the total number of votes in this group.

[3] Covariates usually come from a survey based on a random sample of individuals. Typically, surveys are reported with survey weights w_i^j for each individual to correct for oversampling and non-response, which must be taken into account for any valid inference (e.g. summary statistics, regression coefficients, standard errors, etc.).

5.4 My method

In this section I propose my new ecological inference method. Our approach is illustrated in a schematic in Figure 5.1 and formally stated in Algorithm 2.

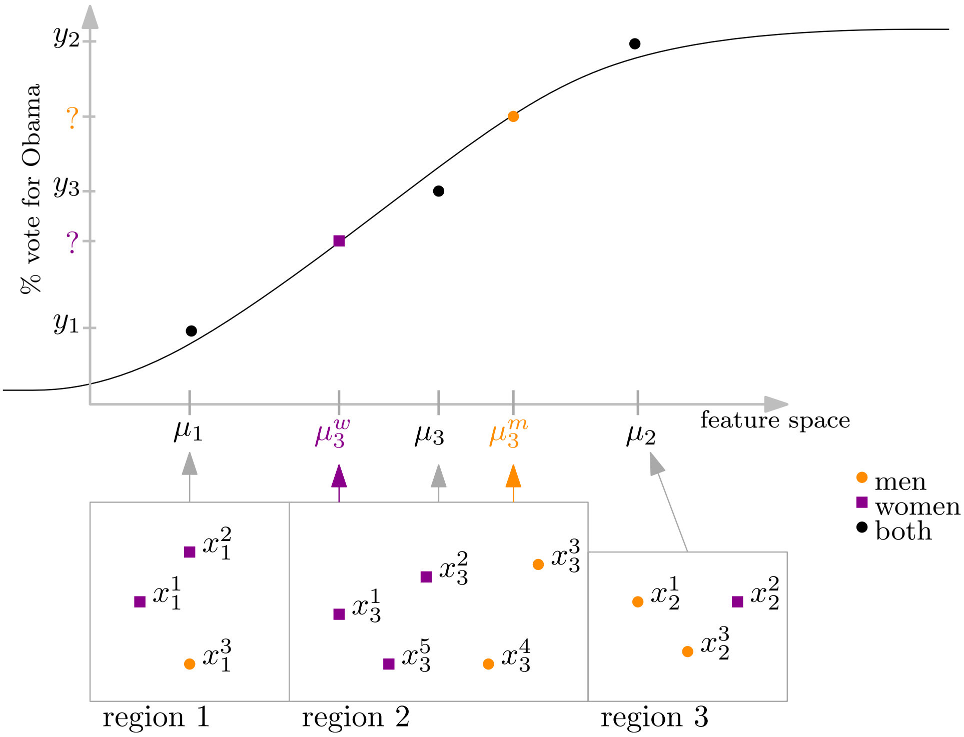


Fig. 5.1 Illustration of my approach. Labels y_1, y_2 and y_3 are available at the group level giving Obama's vote share in regions 1, 2, and 3. Covariates are available at the individual level giving the demographic characteristics of a sample of individuals in regions 1, 2, and 3. We project the individuals from each group into feature space using a feature map $\phi(x)$ and take the mean by group to find high-dimensional vectors μ_1, μ_2 and μ_3, e.g. $\mu_1 = \frac{1}{3}(\phi(x_1^1) + \phi(x_1^2) + \phi(x_1^3))$. Now my problem is reduced to supervised learning, where we want to learn a function $f : \mu \to y$. Once we have learned f I make subgroup predictions for men and women in region 3 by calculating mean embeddings for the men $\mu_3^m = \frac{1}{2}(\phi(x_3^3) + \phi(x_3^4))$ and women $\mu_3^w = \frac{1}{3}(\phi(x_3^1) + \phi(x_3^2) + \phi(x_3^5))$ and then calculating $f(\mu_3^m)$ and $f(\mu_3^w)$. For a more rigorous description of my algorithm see Algorithm 2.

Recall the two-stage distribution regression approach introduced in Section 5.2. My method has a similar approach. To begin, I use FastFood as introduced in Section 2.6 with an RBF kernel to produce an explicit feature map ϕ and calculate the mean embeddings[4], one for each

[4] Distribution regression with explicit random features was previously considered in Oliva et al. (2014) using Rahimi and Recht (2008) to speed up an earlier distribution regression method based on kernel density estimation (Poczos et al., 2013). This approach has comparable statistical guarantees to distribution regression using RKHS-

ALGORITHM 2: Ecological inference algorithm

```
Input: $\left(\{(x_1^j, w_1^j)\}_{j=1}^{N_1}, s_1, y_1\right), \ldots, \left(\{(x_n^j, w_n^j)\}_{j=1}^{N_n}, s_n, y_n\right)$ for $i = 1 \ldots n$ do
    Calculate $\widehat{\mu_i}$ using Eq. (5.9) with FastFood.
    Calculate $\mu_i^m$ using Eq. (5.13) with FastFood.
end
Learn hyperparameters $\hat{\theta} = (\sigma_x^2, \sigma_s^2, \ell)$ of the GP model specified by Eqs. (5.10)–(5.11)
    with observations $y_i$ at locations $(\widehat{\mu_1}, s_1), \ldots, (\widehat{\mu_n}, s_n)$ using gradient descent and the
    Laplace approximation.
Make posterior predictions using $\hat{\theta}$ at locations $(\mu_1^m, s_1), \ldots, (\mu_n^m, s_n)$ using the Laplace
    approximation.
Output: Posterior means and variances for $y_1^m, \ldots, y_n^m$
```

region i, of Equation (5.3) with survey weights:

$$\widehat{\mu_1} = \frac{\sum_j w_1^j \phi(x_1^j)}{\sum_j w_1^j}, \quad \ldots, \quad \widehat{\mu_n} = \frac{\sum_j w_n^j \phi(x_n^j)}{\sum_j w_n^j} \tag{5.9}$$

Next, instead of kernel ridge regression, I use GP regression. Recall that unlike in distribution regression our labels y_i are given by vote counts (k_i, n_i). We use a Binomial likelihood as the observation model in GP regression (this is sometimes known as a logistic Gaussian process (Riihimäki and Vehtari, 2014)). We transform each component of the latent real-valued vector $\mathbf{f}$ by the logistic link function $\sigma(\mathbf{f}) = \frac{1}{1+e^{-\mathbf{f}}}$ and we use the following observation model as our likelihood function:

$$k_i | f(x_i) \sim \text{Binomial}(n_i, \sigma(f(x_i))) \tag{5.10}$$

where we use the formulation for the Binomial distribution of n_i trials and probability of success $\sigma(f(x_i))$. This is the generalized linear model (GLM) specification for binary data, combining a Binomial distribution with logistic link function (Dobson, 2002, Ch. 7).

The predictors in our GP are the mean embeddings $\widehat{\mu_1}, \ldots, \widehat{\mu_n}$. We also include spatial information in the form of 2-dimensional spatial coordinates s_i giving the centroid of region i. Putting these predictors together I adopt an additive covariance structure:

$$\mathbf{f} \sim \mathcal{GP}(0, \sigma_x^2 \langle \widehat{\mu}_i, \widehat{\mu}_j \rangle + k_s(s_i, s_j)) \tag{5.11}$$

Where I have used a linear kernel between mean embeddings weighted by a variance parameter σ_x^2. Since the mean embeddings are already in feature space using the FastFood approximation

mean embeddings but inferior empirical performance (Szabo et al., 2015). As far as I am aware, using FastFood kernel mean embeddings for distribution regression is a novel approach.

to the RBF kernel, we are approximately using the RBF kernel. For the spatial coordinates I use the Matérn covariance function which is a popular choice in spatial statistics (Handcock and Stein, 1993), with $\nu = 3/2$, length-scale ℓ and variance parameter σ_s^2:

$$k(s,s') = \sigma_s^2 \left(1 + \frac{\|s - s'\|\sqrt{3}}{\ell}\right) \exp\left(-\frac{\|s - s'\|\sqrt{3}}{\ell}\right) \tag{5.12}$$

Various other kernel choices for space might improve performance, e.g. the SM kernel introduced in Chapter 2 or a kernel which allowed for discontinuities across state boundaries (this could also be captured by putting fixed state-level effects in the mean function).

By adding together the linear kernel between mean embeddings and the spatial covariance function, we allow for a smoothly varying surface over space and demographics. The intuition is that this additive covariance encourages predictions for regions which are nearby in space and have similar demographic compositions to be similar; predictions for regions which are far away or have different demographics are allowed to be less similar. GP regression with a spatial covariance function is equivalent to the spatial statistics technique of kriging—we are effectively smoothly interpolating y values over a very high dimensional space of predictors. Another way to think about additivity is that we are accounting for a spatially autocorrelated error structure in the predictions we get from the covariates alone. (We also considered a multiplicative structure, which had slightly worse performance. Note that an additive covariance structure with logistic link function actually corresponds to a multiplicative effect, just as in standard logistic regression.)

Equations (5.10)-(5.11) complete my hierarchical model specification. For non-Gaussian observation models like Equation (5.10), the posterior prediction in Equation (5.7) is no longer available in closed form due to non-conjugacy. We follow the standard approach for GP classification and logistic Gaussian processes and use the Laplace approximation (Riihimäki and Vehtari, 2014; Williams and Barber, 1998) as in Section 4.5. The Laplace approximation gives an approximate posterior distribution for $\mathbf{f}$, from which we can calculate a posterior distribution over the k_i of Equation (5.10) as explained in detail in (Rasmussen and Williams, 2006, Section 3.4.2). The Laplace approximation also allows us to calculate the marginal likelihood, which is the probability of the observed data, integrating out $\mathbf{f}$, which we maximize to learn σ_x^2, σ_s^2, and ℓ with gradient ascent.

Once I have learned the best set of hyperparameters for my model I can make predictions for any demographic subgroup of interest. To predict the fraction of men who voted for Obama, I create new mean embedding vectors by gender and region, modifying Equation (5.9):

$$\widehat{\mu_i^m} = \frac{\sum_{j^m} w_1^j \phi(x_1^j)}{\sum_{j^m} w_1^j}, \ \forall i \tag{5.13}$$

where j^m are the indices of the observations of men in region i and $\widehat{\mu_i^m}$ is the mean embedding of the covariates for the men in region i. I then make posterior predictions using the Laplace approximation as above at these new gender-region predictors. Notice that for a new μ^* this requires calculating $k^* = [k_{1*}, k_{2*}, \ldots, k_{n*}]$ of Equation (5.7) where $k_{i*} = \sigma_x^2 \langle \widehat{\mu}_i, \mu_* \rangle + k_s(s_i, s_*)$ using Equation (5.11). Thus new predictions will be similar to existing predictions in regions with similar covariates and they will be similar to existing predictions at the same (and nearby) locations.

My algorithm is stated in Algorithm 2. I now analyze its complexity. Lines 2–2 are calculated by streaming through the data for individuals. For each individual, calculating the FastFood feature transformation $\phi(x_i^j)$ takes $\mathcal{O}(p \log d)$ where $x_i^j \in \mathbb{R}^d$ and $\phi(x_i^j) \in \mathbb{R}^p$. To save memory, there's no need to store each $\phi(x_i^j)$. I simply update the weighted average $\widehat{\mu}_i$ by adding $w_j^i \phi(x_i^j)$ to it. Notice that the demographic subgroup considered in line 2 is simply a subset of the observations calculated in line 2, so there is no added cost to calculate the μ_i^m or indeed a set of $\mu_i^{m_1}, \ldots, \mu_i^{m_q}$ for q different demographic subgroups of interest. Overall, if we have N individuals the for loop takes time $\mathcal{O}(Np \log d)$. Usually $p \ll N$ and $d \ll N$ so this is practically linear and trivially parallelizable.

On line 2 to learn the hyperparameters in the GP regression requires calculations involving the covariance matrix $K \in \mathbb{R}^{n \times n}$. Each entry in K requires computing a dot product $\langle \widehat{\mu}_i, \widehat{\mu}_j \rangle$ which takes $\mathcal{O}(p)$ and it requires computing the Matérn kernel for the spatial locations, which is a fast arithmetic calculation. Once we have K, the Laplace approximation is usually implemented with Cholesky decompositions for numerical reasons. The runtime of computing the marginal likelihood and relevant gradients is $\mathcal{O}(n^3)$ (Rasmussen and Williams, 2006), and gradient ascent usually takes less than a hundred steps to converge. Posterior predictions on line 2 require calculating $k^* \in \mathbb{R}^{1 \times n}$ for each μ_i^m so this is $O(n^2)$. Reusing the Cholesky decompositions above means predictions can be made in $\mathcal{O}(n^2)$. GP regression requires $\mathcal{O}(n^2)$ storage. Overall, we expect $n \ll N$, so my algorithm is practically $\mathcal{O}(N)$, with little extra computational cost arising from the GP regression as compared to the work of streaming through all the observations. The N observations do not need to be stored in memory, so the overall memory complexity is only $\mathcal{O}(n^2)$.

5.5 Experiments

In this section, I describe my experimental evaluation, using data from the 2012 US Presidential election, and compare my results to survey-based exit polls, which are only available for the 18 states for which large enough samples were obtained. Our method enables us to fill in the full picture, with much finer-grained spatial estimation and results for a much richer variety of demographic variables. This demonstration shows the applicability of my new method to a large body of political science literature (see, e.g. Gelman et al. (2008)) on voting patterns by demographics and geography. Because voting behavior is unobservable and due to the ecological inference problem, previous work has been mostly based on exit polls or opinion polls.

I obtained vote totals for the 2012 US Presidential Election at the county level[5]. Most voters chose to either re-elect President Barack Obama or vote for the Republican party candidate, Mitt Romney. A small fraction of voters ($< 2\%$ across the country) chose a third party candidate. Separately, I obtained data from the US Census, specifically the 2006-2010 American Community Survey's Public Use Microdata Sample (PUMS). The American Community Survey is an ongoing survey that supplements the decennial US census and provides demographically representatives individual-level observations. PUMS data is coded by public use microdata areas (PUMAs), contiguous geographic regions of at least 100,000 people, nested within states. I used the 5-year PUMS file (rather than a 1-year or 3-year sample) because it contains a larger sample and thus there is less censoring for privacy reasons. To merge the PUMS data with the 2012 election results, I created a mapping between counties and PUMAs[6], merging individual-level census data and aggregating vote totals as necessary to create larger geographic regions for which the census data and electoral data coincided. The mapping between PUMAs and counties is many-to-many, so I was effectively finding the connected components. Since counties and PUMAs do not cross state borders, none of the geographic regions I created cross state borders. An example is shown in Figure 5.2.

In total, I ended up with 837 geographic regions ranging from Orleans Parish in New Orleans, which voted 91% for Barack Obama to Davis County, a suburb of Salt Lake City, Utah which voted 84% for Mitt Romney. For the census data, I excluded individuals under the age of 18 (voting age in the US) and non-citizens (only citizens can vote in presidential elections). There were a total of 10,787,907 individual-level observations, or in other words, almost 11 million people included in the survey. The mean number of people per geographic region was 12,812 with standard deviation 21,939.

[5] https://github.com/huffpostdata/election-2012-results

[6] using the PUMA 2000 codes and the tool at http://mcdc.missouri.edu/websas/geocorr12.html

There were 223 variables in the census data, including both categorical variables such as race, occupation, and educational attainment and real valued variables such as income in past 12 months (in dollars) and travel time to work (in minutes). We divided the real-valued variables by their standard deviation to put them all on the same scale. For the categorical variables with D categories, I converted them into D dimensional 0/1 indicator variables, i.e. for the variable "when last worked" with categories 1 = "within the past 12 months," 2 = "1-5 years ago," and 3 = "over 5 years ago or never worked" I mapped 1 to $[1\ 0\ 0]^T$, 2 to $[0\ 1\ 0]$ and 3 to $[0\ 0\ 1]$.

Putting together the indicator variables and real-valued variables, I ended up with 3,251 variables total. For every single individual-level observation, I used FastFood with an RBF kernel to generate a 4,096-dimensional feature representation. Using Equation (5.9) I calculated the weighted mean embedding for each region. The result was a set of 837 vectors which were 4,096-dimensional.

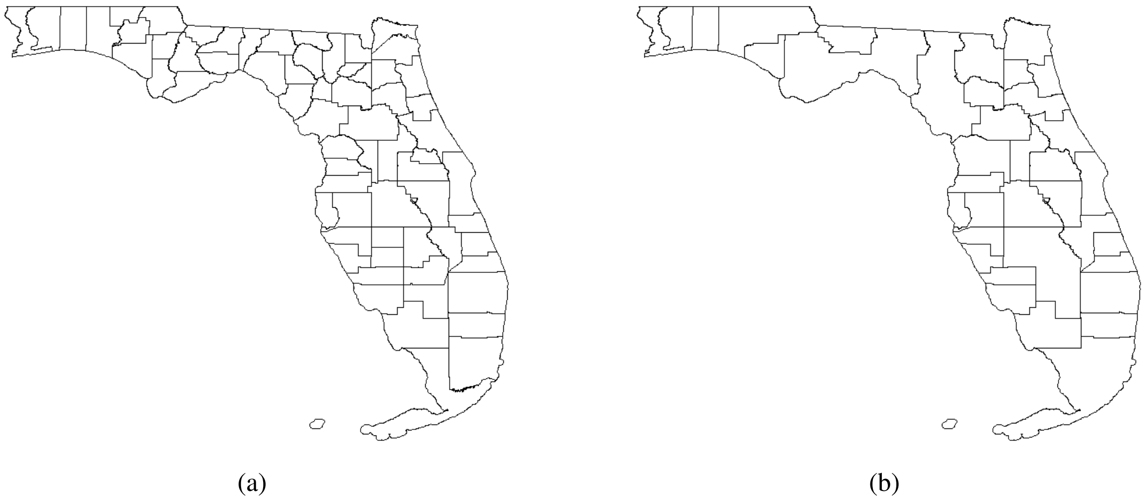
We treated the vote totals for Obama and Romney as is, discarded the remaining third party votes as the exit polls I use for validation did not report third party votes. Thus for each region, I had a positive integer valued 2-dimensional label giving the number of votes for Obama and the total number of votes.

We focused on the ecological inference problem of predicting Obama's vote share by the following demographic groups: women, men, income $\leq$ US\$50,000 per year, income between \$50,000 and \$100,000 per year, income $\geq$ 100,000 per year, ages 18-29, 30-44, 45-64, and 64 plus. For each region, I used the strategy outlined above, restricting our census sample to only those observations matching the subgroup of interest and creating new mean embedding predictors as in Equation (5.13), μ_i^{subgroup}. We made predictions for each region-demographic pair.

All of my models were fit using the GPstuff package with scaled conjugate gradient optimization and the Laplace approximation (Vanhatalo et al., 2013). Since $n \ll N$, the time required to fit the GP model and make predictions is much less than the time required to preprocess the data to create the mean embeddings at the beginning of Algorithm 2.

5.6 Results

I learned the following hyperparameters for my GP: $\sigma_s^2 = 0.18$, $\ell = 7.92$, and $\sigma_x^2 = 4.56$. The σ^2 parameters can be roughly interpreted as the "fraction of variance explained" so the fact that σ_x^2 is much larger than σ_s^2 means that the demographic covariates encoded in the mean embedding are much more important to the model than the spatial coordinates. The length-scale for the Matérn kernel is a little more than half the median distance between locations, which indicates that it is performing a reasonable degree of smoothing. I used 10-fold crossvalidation


(a)

(b)

Fig. 5.2 Election outcomes were available for the 67 counties in Florida shown in (a). Demographic data from the American Community Survey was available for 127 public use microdata areas (PUMAs) in Florida, which sometimes overlapped parts of multiple counties and sometimes contained multiple counties. We merged counties and PUMAs as described in text to create a set of disjoint regions with the result of 37 electoral regions as shown in (b).

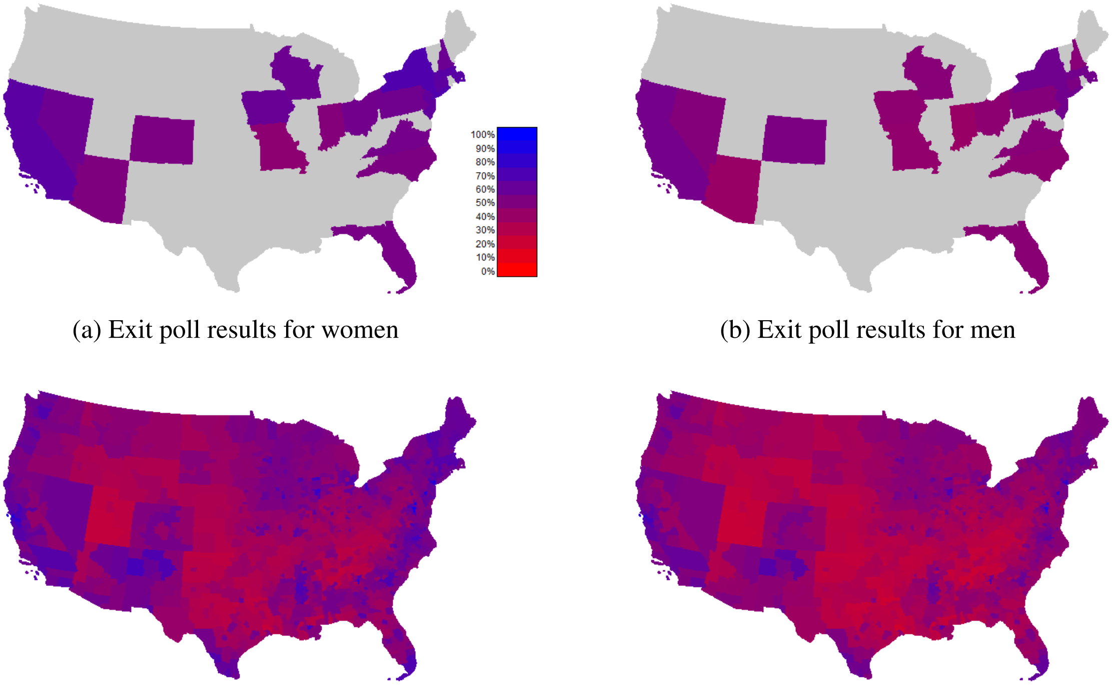
to evaluate my model and ensure that it was not overfitting, an important consideration as generalization performance is critical. The root mean squared error of the model was 2.5 and the mean log predictive density was -1.9. Predictive density is a useful measure because it takes posterior uncertainty intervals into account. For comparison, predicting the national average of Obama receiving 51.1% of the vote in every location has a root mean squared error of 8.3. As a sensitivity analysis, I also considered a multiplicative model, for which the performance was comparable.

To validate my models, I compared to the 2012 exit polls, conducted by Edison Research for a consortium of news organizations. National results were based on interviews with voters in 350 randomly chosen precincts, and state results in 18 states were based on interviews in 11 to 50 random precincts. In these interviews, conducted as voters left polling stations, voters were asked who they voted for and a variety of demographic questions about themselves. Bias due to factors such as unrepresentativeness of the sampled precincts and inadequate coverage of early or absentee voters could be an issue (Barreto et al., 2006). The national results had a margin of error (corresponding to a 95% confidence interval) of 4 percentage points[7] and the state results had a margin of error of between 4 and 5 percentage points (New York Times,

[7] This presumably corresponds to a sample size of only $n = 600$ individuals, since the usual margin of error reported by news organizations is $1.96\sqrt{\frac{.5^2}{n-1}}$

2012). For comparing to the 18 state-level exit polls, I aggregated my geographic regions by state, weighting by subgroup population.

As a preview of the results by gender, income, and age, and to get an idea of the power of my method, Figure 5.3 shows four maps visualizing Obama's support among women and men. In Figures 5.3a–5.3b, I show the results from the exit polls, at the state level, for only 18 states. In Figures 5.3c–5.3d I fill in the missing picture, providing estimates for 837 different regions.



(a) Exit poll results for women

(b) Exit poll results for men

(c) Ecological regression results for women

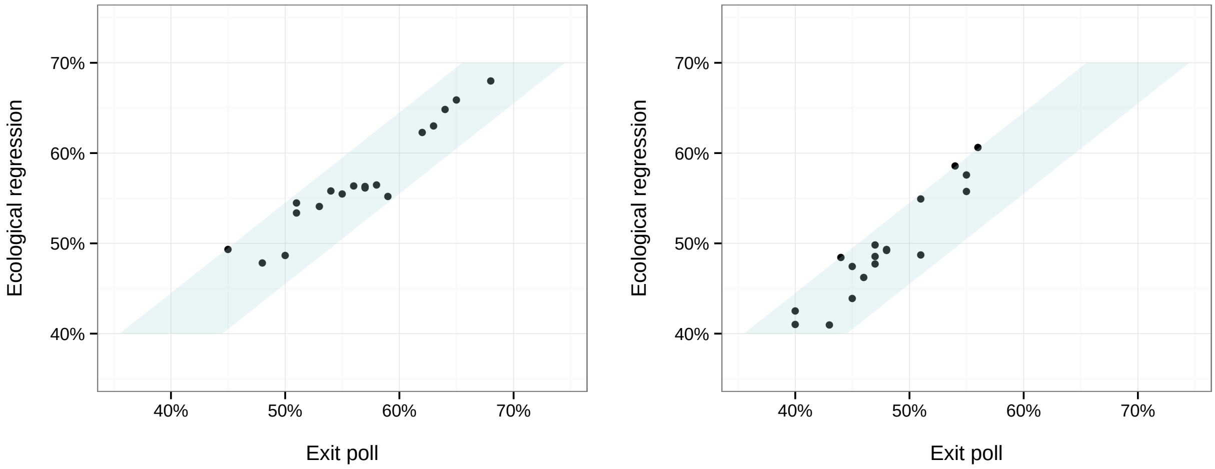
(d) Ecological regression results for men

Fig. 5.3 Support for Obama among women (a) and men (b) in the 18 states for which exit polling was done; due to cost, no representative data was collected for the majority of states or for regions smaller than states. Support for Obama among women (c) and men (d) in 837 different regions as inferred using my ecological regression method.

5.6.1 Gender

Voting by gender is shown in Figure 5.4, where I compare my results to the exit poll results. The fit is quite good, with correlations equal to 0.96 for men and 0.94 for women. The inference that I am most interested in is the gender gap—i.e. how much larger was Obama's vote share among women than among men? In Figure 5.5 I show a histogram of the gender gap by geographic region. The weighted average for the entire country is that women supported Obama by 6.2

percentage points more than men (95% CI 5.2, 7.3). This matches the exit polls which showed a 7 percentage point gap (95% CI -1.1, 15).



(a) Women

(b) Men

Fig. 5.4 My model's ecological predictions (y-axis) of the probability of voting for Obama by gender compared to estimates obtained from an exit poll (x-axis). The blue line shows a 95% confidence interval around the 45° line, corresponding to uncertainty due to exit poll's margin of error of 4 percentage points.

5.6.2 Income

Voting by income is shown in Figure 5.6, where I compare my results to the exit poll results. In this plot, I have included both 95% uncertainty intervals and indicated the 95% margin of error from the exit poll. For low incomes ($\leq$ \$50,000) the correlation is 0.85, for medium incomes (between \$50,000 and \$100,000) the correlation is 0.90, and for high incomes ($\geq$ \$100,000) the correlation is 0.67. Compared to my gender predictions, it is clear that my model is not performing as well in terms of its mean predictions. On the other hand, it is clear that my model's uncertainty intervals are doing what they are designed to do: the large uncertainties in the posterior predictions for the high income group accurately reflect how much we should believe our posterior (recall that in the Bayesian paradigm these are "credibility intervals" rather than frequentist confidence intervals). To explore the reasons my predictions are less accurate, I considered the assumptions underlying distribution regression. In the two-stage analysis of Szabo et al. (2015), distributions are drawn from a meta distribution. When I make a prediction for a test distribution, if the test distribution is in a low density region of the meta distribution then I should not expect a reliable prediction. To test this assumption, I calculated

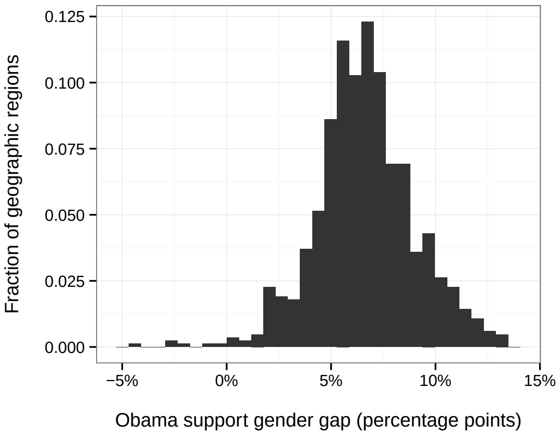


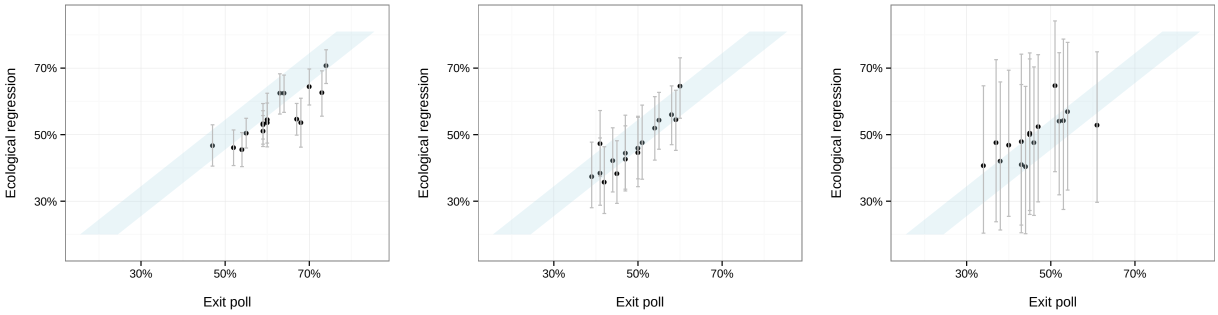
Fig. 5.5 Gender gap is calculated as Obama support among women minus Obama support among men.

k_* in Equation (5.7) separately for each observation for low, medium, and high incomes as k_* provides a measure of the similarity between a test distribution and the distributions used to fit the model. The distribution of the values of k_* are shown in Figure 5.7, where they are compared to the entries in K, i.e. the "in-sample" regions for which I know the labels. While the distributions for low and medium income are quite close to the overall distribution, the distribution for high income is quite far. This is a useful diagnostic for distribution regression in general and ecological inference in particular. It might be possible to correct for this type of bias, as in the covariate shift literature (Gretton et al., 2009).

5.6.3 Age

Voting by age is shown in Figure 5.8. The correlations are 0.60 (ages 18-29), 0.90 (30-44), 0.92 (45-64), and 0.90 (65 years or older). We do not include posterior uncertainty intervals for clarity, but as in the previous section, these seem to be properly calibrated: the average variance is 0.10 for ages 18-29, 0.06 for ages 30-44, 0.05 for ages 45-64, and 0.13 for ages 64 years or older.

In Table 5.1, I compare my national estimates to estimates from the nationally representative exit poll. With the exception of the youngest age group, where I significantly overestimate Obama's support, my predictions are quite accurate, with my posterior predictions matching the exit polls to within just a few percentage points. Our results are weighted based on the percent of the population, rather than the percent of the voting age population. For example, 22% of residents in the US are aged 18-29, but only 19% of voters (according to the exit polls)



(a) Income ≤ \$50,000

(b) Income between \$50,000 and \$100,000

(c) Income ≥ \$100,000

Fig. 5.6 My model's ecological predictions (y-axis) of the probability of voting for Obama by income compared to estimates obtained from an exit poll (x-axis). The blue line shows a 95% confidence interval around the 45° line, corresponding to uncertainty due to exit poll's margin of error of 4 percentage points. The gray error bars are 95% uncertainty intervals around my posterior prediction.

were aged 18-29. This means that my mean embedding vectors are slightly biased, an issue that I intend to address in future work.

Age group	Ecological inference	% of residents	Exit poll [95% CI]	% of voters
18-29	70	22	60 [56, 64]	19
30-44	52	24	52 [48, 56]	27
45-64	45	35	47 [43, 51]	38
65+	43	18	44 [40, 48]	16

Table 5.1 For the US, I compare my ecological inference to nationally representative exit polls.

5.7 Conclusion

In this chapter, I developed a new method to address the long-standing ecological inference problem. Our method makes use of information often left unused in standard ecological regression, that of unlabeled, individual-level data. I formulated a novel and scalable Gaussian process distribution regression method which naturally incorporates spatial information and enables Bayesian inference. Our model generated posterior predictions and uncertainty intervals and where the predictions were less accurate, the uncertainty intervals were larger.

My Bayesian version of distribution regression points the way towards a coherent approach to kernel learning for distribution regression, although scalability is an issue. My approach

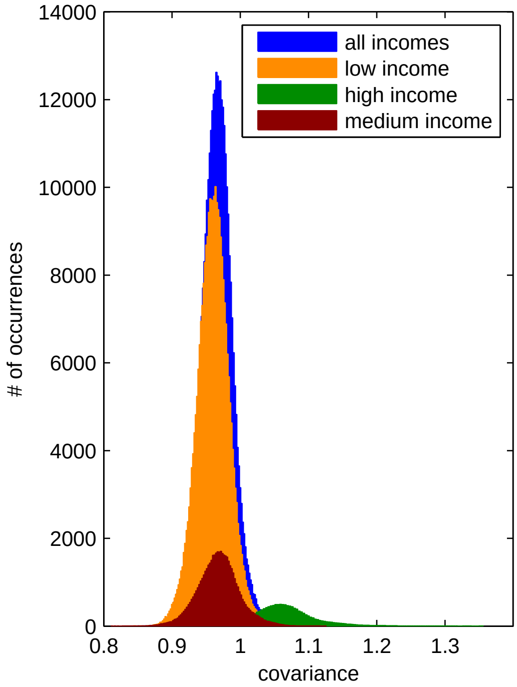
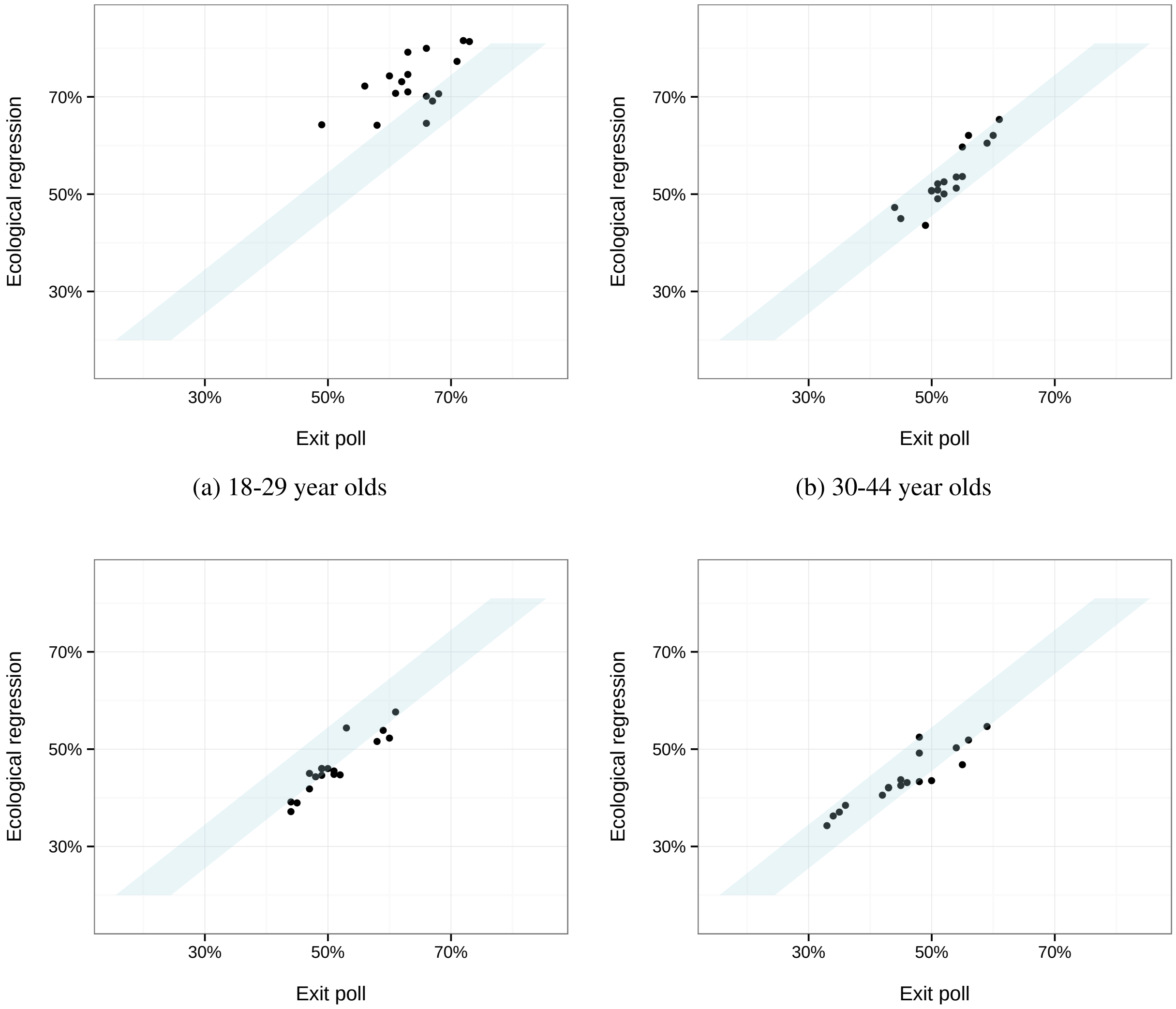


Fig. 5.7 I calculated the vector k_* of Equation (5.7) by income (low, medium, high) for each region and compared the distribution of the values of k_* to the distribution of the values of K ("all incomes").

also highlights the potential for GLM approaches to distribution regression—my Gaussian process regression framework could be immediately extended to continuous, categorical, or multivariate output settings and to including other structure in the input space, such as graph or temporal constraints.

My new method could be used in to trying to answer a variety of social science and public policy questions, especially to answer questions for which carrying out population-representative surveys is impossible or in settings in which the goal is to combine together two different group-level surveys. I used my method in an important political science setting, that of understanding voting patterns by demographic group. I were thus able to move towards filling an important gap in the political science literature about the 2012 US presidential election due to the lack of representative exit polls covering all 50 US states. My model's predictions were quite accurate, despite the fact that I did not actually use all of the information at my disposal; I could have trained my model using exit polling data, where available, and I expect that this approach would have made my predictions even more accurate.



(a) 18-29 year olds

(b) 30-44 year olds

(c) 45-64 year olds

(d) 65 years or older

Fig. 5.8 My model's ecological predictions (y-axis) of the probability of voting for Obama by age compared to estimates obtained from an exit poll (x-axis).

Chapter 6

Algorithmic causal inference for spatiotemporal data[1]

6.1 Introduction

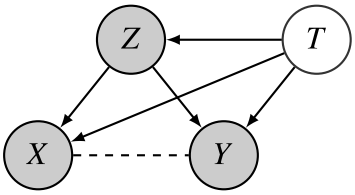
Many algorithmic approaches to causal inference rely on statistical tests of independence between variables. The most popular default methods are the Fisher z-score, Pearson correlation (and partial correlation), and more recently the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2008). More generally, the entire framework of graphical models for causal inference (Pearl, 2009) relies crucially on assumptions about d-separation in graphs, and testing these assumptions with observational data requires applying a valid conditional independence test.

In Section 4.2, we gave a motivating example in which tests of independence (equivalently, association) spuriously reported large correlations when used on non-iid data, due to the underlying autocorrelation structure (see Figure 4.3). Causal inference tools such as the PC algorithm (Spirtes et al., 2001) rely not just on independence testing but also on conditional independence testing, asking whether $X \perp\!\!\!\perp Y \mid Z$. It is not clear a priori what effect non-iid data will have in this case. If the true model is that $X \perp\!\!\!\perp Y \mid Z$, underlying autocorrelation affecting both X and Y might lead us to believe that $X \not\!\perp\!\!\!\perp Y \mid Z$.

Example 1 *Consider the graphical model below: It illustrates the problem of confounding due to non-iid data. T represents time. Shaded nodes X, Y, and Z are observed, and T may be*

[1] This chapter is drawn from Flaxman et al. (2015b).

either observed or unobserved.



The true causal relationship is that $X \perp\!\!\!\perp Y \mid Z$. However, if T is unobserved, it acts as a latent confounding variable, meaning that a spurious edge may be inferred between X and Y, i.e. a conditional independence test rejects the hypothesis $X \perp\!\!\!\perp Y \mid Z$. Once T is observed and controlled for, a conditional independence test will correctly conclude that $X \perp\!\!\!\perp Y \mid Z$.

We are not the first to point out that *every* scientific observation was generated at some specific point in time (Cressie and Wikle, 2011). But in most cases, this information is discarded for convenience. In Section 7.4, we consider a spatial dataset and a temporal dataset each of which is usually analyzed as if the data were iid. We perform tests (Moran's I for spatial data (Moran, 1950) and partial autocorrelation for time series data) which conclusively reject the hypothesis that the observations are iid, and show how causal inference algorithms yield more reasonable results after controlling for the underlying spatial and temporal autocorrelation. Our framework also opens up the possibility of causal inference with structured data, and we develop a novel approach to Gaussian process regression and independence testing which we apply to textual data to determine which language is a translation of another for pairs of texts.

We propose a simple framework for using Gaussian process regression to reduce questions about conditional independence with non-iid data to questions about unconditional independence with iid data, which can be answered with HSIC. Mechanically, our approach is similar to that taken in recent papers on bivariate causal orientation (Peters et al., 2014), in which it is termed *Regression with Subsequent Independence Test* (RESIT), but the motivation is different. The most similar approach to ours is the conditional independence tests proposed by Moneta et al. (2011), which are specifically designed for time series data modeled by a vector autoregression (VAR) model, and thus not directly applicable to, e.g. spatial data. Insofar as our method combines kernel-based independence tests with the PC algorithm, it is similar to the Kernel PC algorithm proposed by Tillman et al. (2009), but our conditional independence tests are different. The strategy we propose is straightforward, generally applicable wherever Gaussian processes can be used, and it works for both pre-whitening non-iid data and for testing conditional independence.

6.2 Contributions

6.2.1 Approach

The centerpiece of the approach is to use regression to remove dependence: on space, time, or a set of conditioning variables. We assume that we have random variables (X, Y, Z), observed at locations S (in time, space, or on a network). Conditional independence testing then proceeds in the following three steps:

1. We first use separate Gaussian process (GP) regressions of $X|S$, $Y|S$ and $Z|S$ to obtain residuals

$$r_x = x - \hat{\mathbf{E}}[x|s] \text{ and } r_y = y - \hat{\mathbf{E}}[y|s] \text{ and } r_z = z - \hat{\mathbf{E}}[z|s], \tag{6.1}$$

   thus pre-whitening each variable and eliminating its dependence on S.

2. Next, we again use GP regression to obtain residuals

$$\varepsilon_{xz} = r_x - \hat{\mathbf{E}}[r_x|r_z] \text{ and } \varepsilon_{yz} = r_y - \hat{\mathbf{E}}[r_y|r_z] \tag{6.2}$$

   from regressing both r_x and r_y on r_z separately.

3. Finally, we use HSIC to test for independence:

$$\varepsilon_{xz} \perp\!\!\!\perp \varepsilon_{yz}. \tag{6.3}$$

At a mechanical level, in each step we use Gaussian process regression to obtain residuals for which we have controlled for variation—in the case of the dependence structure of the data, we are controlling for, say, temporal variation. In the case of conditional independence, we are controlling for the variation due to variable Z.

Strategies like the above are standard practice in statistical modeling. In econometrics, this approach is justified by the Frisch-Waugh-Lovell theorem (Frisch and Waugh, 1933) (which was originally stated in a time series context), which proves that in the case of linear regression, partial correlations can be calculated by finding the correlations between residuals. In the spatial statistics and time series literature, pre-whitening by fitting models and obtaining residuals, removing trends, and taking first differences are all standard approaches (Box et al., 2008). However, to our knowledge, a full formulation of this strategy, combining a non-parametric regression and independence test, has not been stated explicitly before. Moreover, beyond the case of linear models with Gaussian noise, the conditions under which it holds are not known. In Section 7.2, we state precise conditions under which our test is valid.

We believe that our method can serve as a default template when testing for conditional independence with non-iid data. It is equally useful as a simple method for testing for conditional independence even when observations are iid, in which case the pre-whitening step can be skipped. We highlight a few reasons for relying on GP regression for pre-whitening and conditioning, rather than using parametric tests or relying solely on kernel-based tests:

1. Gaussian processes provide a principled Bayesian approach. Yet, for regression their convenient analytic form means that hyperparameters can be learned much more efficiently than in many other fully Bayesian models since we can integrate out additive noise. This provides considerable computational savings and increased numerical accuracy.

2. A variety of packages already exist to fit Gaussian process regression (Kalaitzis et al., 2013; Karatzoglou et al., 2004; Rasmussen and Nickisch, 2010; Vanhatalo et al., 2013), which perform inference using either optimization methods, grid search, or sampling (MCMC) strategies.[2]

3. In the case of time series and especially spatial data, the Gaussian process framework is a long-standing, proven method, typically referred to as "kriging" in geostatistics (Salkauskas, 1982). In applied fields where it has been used, practitioners are adept at designing appropriate covariance functions (Mercer kernels) adapted to their problem domains. For instance, the Matérn kernel is a popular choice.

   With spatiotemporal data, much recent work has focused on designing classes of sophisticated non-separable and non-stationary covariance functions for capturing complex dependencies (Gneiting et al., 2007). These covariance functions could be directly imported into the kernel-based statistical tests, but their use requires model-checking and diagnostics. Recent work suggests that complicated time series dynamics can be automatically fit through combinations of covariances (Duvenaud et al., 2013; Wilson and Adams, 2013a).

4. By design, Gaussian processes allow for easy graphical model-checking: diagnostic plots can be inspected to check for autocorrelation and overfitting.

5. Our new GP formulation for structured inputs and outputs, as introduced in Section 6.2.7, opens up the possibility of conditional independence and causal inference with structured data such as text, images, and anything else on which a Mercer kernel can be defined.

[2] See also http://www.gaussianprocess.org/#code for more details.

6. In the case of real-valued data, our formulation allows for testing conditional independence without first discretizing the conditioning set. This is useful because discretization is fraught with information loss—we may lose the relevant time scale or we might even introduce dependence due to the quantization level inherent in binning.

6.2.2 Related Work

We are only aware of one general test (Zhang et al., 2008) for unconditional independence with non-iid data. It requires precisely specifying the dependence structure of the data as a graphical model, and then decomposing this model into cliques, exploiting the connection between the exponential family of distributions and kernels over graphical models. The analysis is by no means simple—for instance, it has not been extended to a lattice structure; this is unfortunate because assuming that points are on a lattice is a basic starting point in the spatial statistics literature.

In the case of conditional independence, several tests have been proposed, including a test based on characteristic functions (Su and White, 2007), the Normalized Conditional Cross-Covariance Operator (NOCCO) (Fukumizu et al., 2007), Kernel-based Conditional Independence (KCI) (Zhang et al., 2011), a scale invariant measure (Reddi and Póczos, 2013), a scalable method called conditional correlation independence (CCI) (Ramsey, 2014), and a permutation-based conditional independence test (Doran et al., 2014). However, these tests will all be biased for non-iid data, just like the unconditional tests. While CCI does not address the non-iid case, for conditional independence it takes an approach with a similar flavor to our method, and makes similar asymptotic claims. However, CCI is based on a finite basis expansion, so consistency only holds in the limit as the number of basis functions goes to infinity along with the number of samples, whereas we use a consistent non-parametric regression method, so consistency holds in the large-sample limit.

A few works focus specifically on the time series domain, but it is not clear if they can be generalized to spatial or continuous / partially observed time series data. Moneta et al. (2011) proposed a conditional independence test, appropriate for time series data that can be modeled as a vector autoregressive (VAR) process, based on calculating divergence between density estimates using smoothing kernels. Besserve et al. (2013) proposed a powerful kernel cross-spectral density operator for characterizing independence between time series and Chwialkowski and Gretton (2014) explored the behavior of HSIC for random processes (e.g., time series data), showing a new consistent estimate of the p-value for non-iid data, but neither of these works address the conditional independence case.

Even with iid data, these tests have not found widespread application. Closed form distributions under the null are not available, except in the cases of KCI and the test in Su and

White (2007), so permutation testing is required. Valid permutation testing of $X \perp\!\!\!\perp Y \mid Z$ must preserve the marginal structure $X \perp\!\!\!\perp Z$ and $Y \perp\!\!\!\perp Z$. Assuming that Z is categorical, for each value of Z one can consider permuting X. But when Z is real-valued, discretization is necessary first. Clustering is a common approach, as in Tillman et al. (2009). By contrast, our regression-based approach naturally handles categorical, real-valued, and even structured (image or text) data.

6.2.3 Causal Inference Methods

We focus on two classes of causal inference methods: constraint-based causal structure learning algorithms exemplified by the PC algorithm and bivariate causal orientation methods, i.e. the additive non-Gaussian (ANG) framework (Hoyer et al., 2008) and the Continuous Additive Noise Model (CANM) framework (Peters et al., 2014).

The PC algorithm learns an equivalence class of partially directed acyclic graphs (PDAGs) which are consistent with the conditional independencies entailed by the data, as tested with statistical tests for conditional independence. After learning this "skeleton," the algorithm finds V-structures, also known as colliders, of the form $A \rightarrow B \leftarrow C$ which are consistent with the learned conditional independencies and orients edges accordingly. For example, a V structure $A \rightarrow B \leftarrow C$ would be implied by $A \perp\!\!\!\perp C$ and $A \not\!\perp\!\!\!\perp C \mid B$. Finally, the algorithm orients any other edges it can to be consistent with the edges it has already oriented, so long as these orientations do not introduce any new V structures or cycles. Once a PDAG is learned, independence relations can be read off the graph using the rules of d-separation. For a detailed discussion of the PC algorithm see Spirtes et al. (2001) and for causal DAGs and d-separation see Pearl (2009).

The bivariate causal orientation methods compare two models, a forward model: $Y = f_1(X) + \varepsilon_1$ and a backwards model: $X = f_2(Y) + \varepsilon_2$. After fitting non-parametric regressions to obtain residuals $\hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$, an independence test such as HSIC is used to test whether $\hat{\varepsilon}_1 \perp\!\!\!\perp X$ and $\hat{\varepsilon}_2 \perp\!\!\!\perp Y$. If, for example, $\hat{\varepsilon}_1 \perp\!\!\!\perp X$ but $\hat{\varepsilon}_2 \not\!\perp\!\!\!\perp Y$ we reject the backward model and retain the forward model, $X \rightarrow Y$. For a detailed discussion see Peters et al. (2014).

6.2.4 Testing Conditional Independence by Regression and Unconditional Independence

We start by assuming both faithfulness and the Markov condition, the same assumptions made for the PC algorithm:

Faithfulness There exists a causal DAG G and a probability distribution over random variables X,Y,Z such that if $X \perp\!\!\!\perp Y \mid Z$ then X and Y are d-separated by Z in graph G

Markov If X and Y are d-separated by Z in G, then $X \perp\!\!\!\perp Y \mid Z$.

Second, we assume that we have access to a conditional regression estimator to remove the dependence on Z from X and Y. More specifically, we assume that this can be done in an additive fashion:

Consistent Regressors We assume that we have consistent non-parametric regressors $\hat{m}_x(Z)$ and $\hat{m}_y(Z)$ that converge to $\mathbf{E}[X|Z]$ and $\mathbf{E}[Y|Z]$ respectively, such as Gaussian process regression.

Additive Noise Model If Z is the cause of X or Y, we assume an additive independent noise model. That is, if Z causes X (respectively Y) then $X = f(Z) + \varepsilon$ where $Z \perp\!\!\!\perp \varepsilon$. Notice that we are not assuming in this case that $Y \perp\!\!\!\perp \varepsilon$, or that the noise is always additive. For example, if the true structure is $X \leftarrow Z \leftarrow Y$ then we assume $X = f(Z) + \varepsilon$ but we do not assume $Y = g(Z) + \varepsilon_2$ or $Z = g(Y) + \varepsilon_2$.

Finally, we assume that we have a valid method for testing unconditional independence between random variables, such as HSIC. Given these assumptions, our method can be summarized in the following simple algorithm:

1. Obtain residuals $\varepsilon_{xz} = X - \hat{m}_x(Z)$ and $\varepsilon_{yz} = Y - \hat{m}_y(Z)$

2. Test whether $\varepsilon_{xz} \perp\!\!\!\perp \varepsilon_{yz}$.

We claim that $\varepsilon_{xz} \perp\!\!\!\perp \varepsilon_{yz} \iff X \perp\!\!\!\perp Y \mid Z$.

We remark upon the assumptions underlying our method. As explained in Section 4.10, Choi and Schervish (2007) demonstrate almost sure convergence for GP regression under mild conditions while Van Der Vaart and Van Zanten (2011) provide convergence rates for GP regression. Additive noise models underlie many standard regression techniques such as linear regression, kernel ridge regression, GP regression, and generalized additive models. Furthermore, it is straightforward to test this assumption in our framework: if we assume that $X = f(Z) + \varepsilon$ we can check this assumption by using GP regression to regress X on Z to estimate $\hat{\varepsilon}$. Then we use HSIC to check whether $\hat{\varepsilon} \perp\!\!\!\perp Z$.

As previously discussed in Section 4.2, an application of our method to synthetic data is illustrated in Figure 4.4 in the case where Z represents time.

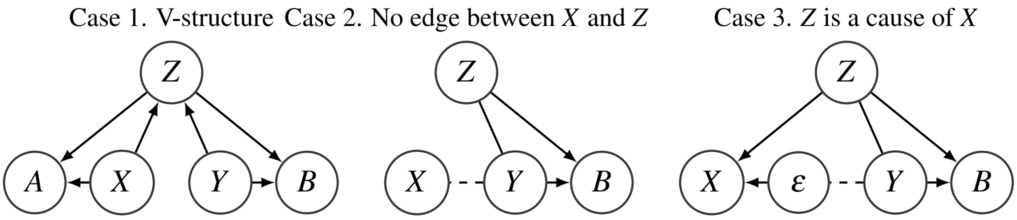

Case 1. V-structure
Case 2. No edge between X and Z
Case 3. Z is a cause of X
Z
A
X
Y
B
ε

Fig. 6.1 Three cases of dependence between (X,Y,Z), corresponding to the cases in the proof of Theorem 4 that $X \perp\!\!\!\perp Y \mid Z$ if and only if $X - \mathbf{E}[X|Z] \perp\!\!\!\perp Y - \mathbf{E}[Y|Z]$. We define auxiliary variables $A := X - \mathbf{E}[X|Z]$ and $B := Y - \mathbf{E}[Y|Z]$ which are uniquely determined by their parents. **Case 1**: we have a V-structure $X \rightarrow Z \leftarrow Y$, so we see that $X \not\perp\!\!\!\perp Y|Z \Rightarrow X - \mathbf{E}[X|Z] \not\perp\!\!\!\perp Y - \mathbf{E}[Y|Z]$ because A and B are d-connected. **Case 2**: If there is no edge between X and Z, any path from X to B must go through Y. **Case 3**: Z and ε cause X, so the only possible path from ε to B is through Y.

Theorem 4 *Given structural assumptions of faithfulness and the Markov assumptions, and assuming that we have consistent regressors with an additive noise model, whenever Z is a cause of X or Y, it follows that*

$$X \perp\!\!\!\perp Y \mid Z \textit{ if and only if } X - \mathbf{E}[X|Z] \perp\!\!\!\perp Y - \mathbf{E}[Y|Z].$$

Proof We consider three cases for the structure of the causal graph G corresponding to the joint distribution of X, Y, and Z below. For each, we prove both the forward and reverse directions of the theorem. The associated graphical models are given in Figure 6.1. Our three cases are exhaustive due to symmetry (i.e. given three variables, we might need to switch the variables called X and Y) and the fact that they cover all possible dependencies in a DAG between X and (Y,Z), and all possible dependencies between Z and Y.

Case 1 Assume that we have a graph G with V-structure as in Figure 6.1

$$X \rightarrow Z \leftarrow Y.$$

This immediately implies $X \not\perp\!\!\!\perp Y \mid Z$, so we do not need to prove anything for the forward direction. We prove the reverse direction by contradiction. Thus we assume

$$X - \mathbf{E}[X|Z] \perp\!\!\!\perp Y - \mathbf{E}[Y|Z] \text{ but } X \not\perp\!\!\!\perp Y \mid Z.$$

and specifically this is because we have the V-structure $X \rightarrow Z \leftarrow Y$. Adding a new set of variables $A := X - \mathbf{E}[X|Z]$ with parents X and Z and $B := Y - \mathbf{E}[Y|Z]$ with parents Y and Z to the DAG, as shown in Figure 6.1 does not change the model since these random variables are entirely determined by (X,Z) and (Y,Z) respectively. Now we see that the path $A \leftarrow Z \rightarrow B$ d-connects A and B. By the faithfulness assumption it follows that $A \not\perp\!\!\!\perp B$, which is a contradiction.

Case 2 If there is no edge between Z and X or between Z and Y or both, the test reduces to that of testing unconditional independence between X and Y. Without loss of generality let us assume there is no edge between X and Z. $\mathbf{E}[X|Z]$ is a constant, call it c, so $X - \mathbf{E}[X|Z] = X - c$. As before, add the auxiliary variable $B := Y - \mathbf{E}[Y|Z]$ with parents Y and Z to the DAG as in the Figure 6.1, Case 2. Then we are testing whether $X - c \perp\!\!\!\perp B$. Since c is constant, this holds if and only if $X \perp\!\!\!\perp B$. Finally, $X \perp\!\!\!\perp B$ if and only if $X \perp\!\!\!\perp Y$: if X and Y are d-connected by a path p, then we can add the edge from Y to B to the path p to make X and B d-connected. If instead X and Y are d-separated, then so are X and B because any path from X to B must go through Y.

Case 3 Z is a cause of X or Y or both, so assume without loss of generality that Z is a cause of X. Then by assumption we can write $X = f(Z) + \varepsilon$ with $Z \perp\!\!\!\perp \varepsilon$ where $\varepsilon = X - \mathbf{E}[X|Z]$ and check $\varepsilon \perp\!\!\!\perp Y - \mathbf{E}[Y|Z]$. Once again, we add a variable $B := Y - \mathbf{E}[Y|Z]$ with parents Y and Z to the DAG as shown in Figure 6.1, Case 3. Now we prove $X \perp\!\!\!\perp Y \mid Z \iff \varepsilon \perp\!\!\!\perp B$ by considering two subcases.

Subcase 1 If there is an edge in either direction between ε and Y in Figure 6.1, Case 3 then X and Y are d-connected by the path $X \leftarrow \varepsilon - Y$ and ε and B are d-connected by the path $\varepsilon - Y \rightarrow B$, so we conclude $X \not\perp\!\!\!\perp Y \mid Z$ and $\varepsilon \not\perp\!\!\!\perp B$ by faithfulness. Thus we have proved the forward and reverse directions for this subcase.

Subcase 2 If there is no edge between ε and Y in Figure 6.1, then ε and B are d-separated, since X is a collider in the path $\varepsilon \rightarrow X \leftarrow Z \rightarrow B$, which is thus blocked. Moreover X and Y are d-separated given Z, since Z blocks the path $X \leftarrow Z - Y$. By the Markov assumption, this implies $X \perp\!\!\!\perp Y \mid Z$ and $\varepsilon \perp\!\!\!\perp B$. This proves the forward and reverse directions for this subcase.
■

6.2.5 Testing Conditional Independence with GP Regression and HSIC

Based on the above general framework, we propose the use of Gaussian process (GP) regression, which is almost surely consistent assuming Gaussian errors Choi and Schervish (2007) with good rates of convergence Van Der Vaart and Van Zanten (2011), and the Hilbert-Schmidt

Independence Criterion (HSIC) for testing for independence[3]. Neither of these choices is crucial—we picked GP regression because of its long history in spatial statistics and widespread use as a convenient non-parametric regression method. We picked HSIC because it is equal to 0 if and only if the distributions under consideration are independent, whenever the kernel is characteristic. But in cases where domain knowledge could be used to guide the choice of independence test or pre-whitening method, these will of course be preferable to generic choices[4].

In the following we assume without loss of generality that X (and Y) is embedded in a vector space. That is, we assume that regression on X is well defined. Hence, given observations $(X,Y,Z) = \{x_i, y_i, z_i\}$ we use GP regression to fit the models $X = f(Z) + \varepsilon_1$ and $Y = g(Z) + \varepsilon_2$. This requires specifying (possibly different) covariance functions over Z. If $k(z,z')$ is a covariance function (Mercer kernel) over Z then we could sample directly from the GP prior, where we follow general practice and set the mean to 0:

$$f \sim \mathcal{GP}(0,k)$$

Let K be the Gram matrix where $K_{ij} = k(z_i, z_j)$. Conditional on the observations (X,Z) our data follows a multivariate Gaussian distribution. For a new location

$$x^* \mid X,Z,z^* \sim \mathcal{N}(K_*(K+\sigma^2 I)^{-1}X, K_{**} - K_*(K+\sigma^2 I)^{-1}(K_*)^T)$$

where $K_* = [k(z^*,z_1),\ldots,k(z^*,z_n)]$ and the σ^2 term is added because we assume that our observations are noisy, as discussed above.

We are not actually interested in observations at new locations, but in making point predictions at the existing locations. In other words we use the Gaussian process as a smoother. Hence we replace K_* by K and find a vector of mean predictions: $\hat{X} = K(K+\sigma^2 I)^{-1}X$. The residuals are:

$$\varepsilon_{xz} = X - \hat{X} = X - K(K+\sigma^2 I)^{-1}X = (I + \sigma^{-2}K)^{-1}X \tag{6.4}$$

Using a possibly different kernel, say l with kernel matrix L, we obtain residuals from smoothing Y via $\varepsilon_{yz} = Y - \hat{Y} = (I + \sigma^{-2}L)^{-1}Y$ in exactly the same way.

[3] Since GP regression is a.s. consistent, it is possible that our estimated residuals will not converge to their true values on a set of measure 0, but in the worst case all that this could do is bias our estimate of HSIC on a set of measure 0, so the standard estimate of HSIC will still be consistent.

[4] Note that guarantees about consistency and convergence for GP regression apply in the large sample limit. It is entirely possible that for misspecified models and / or small samples, generic methods like GP regression will fail to remove all dependence.

Now, as proved above, $\varepsilon_{xz} \perp\!\!\!\perp \varepsilon_{yz}$ if and only if $X \perp\!\!\!\perp Y \mid Z$. To test the hypothesis $\varepsilon_{xz} \perp\!\!\!\perp \varepsilon_{yz}$ we use HSIC, which requires specifying kernels on the residuals, say $\tilde{p}$ and $\tilde{q}$ on ε_{xz} and ε_{yz} respectively. This leads to the kernel matrices P and Q. The associated HSIC test statistic is $\frac{1}{n^2}\mathrm{tr}(PHQH)$ for the centering matrix $H = I - \frac{1}{n}11^T$.

6.2.6 Pre-whitening with GPs for causal inference

Our pre-whitening algorithm was presented in Section 4.2. In the case of the PC algorithm, an alternative approach would be to always include S in the conditioning set when testing for conditional independence. With enough data, this should be equivalent to the two-stage process we proposed. But because we believe that there is the potential for an important autocorrelation structure which we need to worry about, we think it is better to explicitly adjust for it in every variable first. This approach saves on computational time and modeling complexity: for moderately sized datasets, we can use a fully Bayesian analysis and carefully inspect the results of our pre-whitening step for each variable. By contrast, the PC algorithm could entail many conditional independence tests, so we need these to be automatic and relatively fast. Many conditional independence tests also rely on categorical conditioning sets, which are often obtained by first discretizing; this approach will be very difficult since observations are usually not repeated in space or time.

Finally, for the two-variable causal orientation task, e.g. as addressed by the RESIT framework, space or time would need to be included as part of the regression and again as part of the independence test, turning what was a simple univariate regression followed by unconditional independence test into two more complicated steps, a multivariate regression followed by conditional independence test.

6.2.7 Gaussian processes for Structured Data

Since Gaussian processes depend on defining a kernel between observations, they can be used for highly structured data such as images and text. Given domains $\mathcal{X}, \mathcal{Y}$ we can define a joint GP over both domains, that is, using a kernel $k : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ such that random variables f, indexed by $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are drawn from a multivariate Gaussian distribution with covariance matrix given by k, as evaluated on the index set and with mean function $\mu : (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ as evaluated on the index set.

A special but particularly interesting case arises whenever the kernel function k is given by a product over kernels on $\mathcal{X}$ and $\mathcal{Y}$ respectively, i.e. whenever

$$k((x,y),(x',y')) = k_x(x,x')k_y(y,y').$$

Such a situation occurs, e.g. in multivariate GP regression where $\mathcal{Y} = \{1, \ldots, d\}$, i.e. where $\mathcal{Y}$ denotes the coordinate index of the regression problem and where k_y denotes the correlation between the coordinate-wise regressions. Likewise, when $\mathcal{Y}$ is the domain of images or documents, we therefore end up modeling the similarity between structured objects in $\mathcal{Y}$ using their covariates in $\mathcal{X}$.

We now exploit the duality discussed by Williams (1998) between feature space representations and GPs to introduce estimates of feature functions on $\mathcal{Y}$. That is, we will adhere to the GP treatment for the covariate-dependent part of the kernel via $k_x(x,x')$ and use a feature space representation for the label-dependent part $l(y,y') = \langle \psi(y), \psi(y') \rangle$. The main motivation is that this will allow us to reason about feature space embeddings of distributions and of conditional probability distributions efficiently.

Before we do so, recall scalar GP regression as introduced in Chapter 4. There one assumes that the random variable f, as indexed by $x \in \mathcal{X}$ follows a normal distribution with covariance function k and mean function μ. The idea is to extend the predictive distribution $Y^*|Y,X,X^*$, as captured by Equation 4.5. We now extend this to vector valued functions and subsequently to general index sets. In the standard treatment, we assume that:

$$f(X), f(X^*) \mid X, x^* \sim \mathcal{N}(0, K)$$

where $K_{ij} = k(x_i, x_j) + \delta_{ij}\sigma^2$. So conditioning we find:

$$f(X^*) \mid Y, X, x^* = \mathcal{N}(K(x^*,X)(K+\sigma^2 I)^{-1}y, K(x_*,x_*) - K(x_*,x)(K+\sigma^2 I)^{-1}K(x,x_*))$$

What if $f(X)$ isn't in $\mathcal{R}$, such as $Y \in \mathcal{R}^d$ or whenever Y is a string or an image? We begin with $\mathcal{Y} = \{1, \ldots, d\}$, thus we could view the scalar case as $\mathcal{Y} = \{1\}$ and therefore with estimates in $\mathbb{R}^{\mathcal{Y}} = \mathbb{R}^1$. In general, the challenge is to deal with possible normalization problems of distributions over infinite-dimensional objects. The trick is to consider *evaluating* the GP on $\mathcal{Y}$ only on relevant points $y \in \mathcal{Y}$ rather than considering a possibly infinite dimensional set of evaluations.

For computational convenience of derivation we adopt the argument of Williams (1998), i.e. that $f(y) = \langle v, \psi(y) \rangle$ is a linear function in the space of the features $\psi(y)$, where $v \sim \mathcal{N}(0, \mathbf{1})$ and therefore $f \sim \mathcal{GP}(0, l)$. It is understood that the kernel satisfies $l(y,y') = \langle \psi(y), \psi(y') \rangle$. This is entirely consistent whenever ψ is finite-dimensional. For the purpose of evaluation on a finite number of terms, we can always assume that ψ denotes the Cholesky factors of the covariance matrix L.

We now assume that we are given features $\psi(y_1), \ldots \psi(y_n)$, which are drawn from a Gaussian process with kernel k and mean 0. That is, we assume that this holds for any one-dimensional projection of $\psi(y)$ onto a unit-vector. Using Equation (4.5) we have that

$$\begin{aligned} \psi(Y^*) \mid Y &\sim \mathcal{N}(\bar{\mu}, \bar{K}) \\ \text{where } \bar{\mu} &= K(X^*, X)(K(X,X) + \sigma^2 I)^{-1} \psi(Y) \\ \bar{K} &= K(X^*, X^*) - K(X^*, X)(K(X,X) + \sigma^2 I)^{-1} K(X, X^*) \end{aligned} \tag{6.5}$$

In it we used the shorthand $\psi(Y) = (\psi(y_1), \ldots \psi(y_n))$ and analogously $\psi(Y^*) = (\psi(y_1^*), \ldots, \psi(y_{n^*}^*))$, whenever Y^* is a set. For instance, whenever $\mathcal{Y} = \{1, \ldots d\}$ and $\psi(y) = e_y$, this simply decomposes into d decoupled Gaussian processes. More generally, we can evaluate by taking inner products with test functions $\psi(y)$. At this point the evaluation reduces to kernel computations $l(y, y')$.

As before, we employ the GP not for prediction but for smoothing only, i.e. we are mostly interested in the residuals $\hat{\psi}_i - \psi(y_i)$ at locations x_i rather than the predictions $\hat{\psi}_i$ themselves. Since we will ultimately use HSIC, we do not need to explicitly compute the residuals; rather we need to compute the Gram matrix R of the residuals with

$$\begin{aligned} R_{ij} &= \mathbf{E}_{\hat{\psi}} \left[\left\langle \hat{\psi}_i - \psi(y_i), \hat{\psi}_j - \psi(y_j) \right\rangle \right] \\ &= \operatorname*{Cov}_{\hat{\psi}} \left[\left\langle \hat{\psi}_i, \hat{\psi}_j \right\rangle \right] + \left\langle \mathbf{E}_{\hat{\psi}} \left[\hat{\psi}_i \right] + \psi(y_i), \mathbf{E}_{\hat{\psi}} \left[\hat{\psi}_j \right] - \psi(y_j) \right\rangle \end{aligned} \tag{6.6}$$

The second line follows from the fact that $\mathrm{Cov}(A, B) = \mathbf{E}[AB] - \mathbf{E}[A][B]$. To evaluate this expression we use the fact that the covariance is given in Equation (6.5). Its contribution to the entire matrix R is

$$K - K(K + \sigma^2 I)^{-1} K = K(I + \sigma^{-2} K)^{-1}. \tag{6.7}$$

where we used the Woodbury matrix identity[5]. Next we use the fact that

$$\mathbf{E}_{\hat{\psi}} \left[(\hat{\psi}_1, \ldots \hat{\psi}_n) \right] - \psi(Y) = K(K + \sigma^2 I)^{-1} \psi(Y) - \psi(Y) = -(I + \sigma^{-2} K)^{-1} \psi(Y) \tag{6.8}$$

Again using the Woodbury matrix identity. Taking inner products and plugging this back into Equation (6.6) we obtain

$$R = K(I + \sigma^{-2} K)^{-1} + (I + \sigma^{-2} K)^{-1} L (I + \sigma^{-2} K)^{-1} \tag{6.9}$$

Note that R decomposes into two parts: first the contribution of the residuals due to smoothing in K. This converges to $\sigma^2 I$ for small σ^2, i.e. whenever we assume that there is little additive

[5] $(A+B)^{-1} = A^{-1} - A^{-1}(B^{-1} + A^{-1})^{-1} A^{-1}$

noise associated with $y|x$ the contribution to the residuals matrix is very small off-diagonal and equal to σ^2 on the diagonal. Secondly, we have an appropriately smoothed term between K and L. Again, this vanishes for small additive noise but it also vanishes whenever K and L are coherent.

Once we have access to R, we can use HSIC to test independence: $\widehat{HSIC}(R,X) = \frac{1}{n^2}\text{tr}RHKH$. If we choose characteristic kernels for K (possibly a different K relative to the one used in the GP regression) and L then we do not need to consider doing either a further embedding of the residuals, or solving the pre-image problem and applying a different embedding. The reason is that what we care about testing is the independence between the residuals and X. Although we do not have access to the residuals, calculating HSIC only requires access to the Gram matrices corresponding to the feature space representation of the residuals and X. This is exactly what we have in the form of R and K respectively.

6.3 Experiments

6.3.1 Spatial data

The Boston Housing dataset, originally investigated in Harrison Jr and Rubinfeld (1978) has been widely used in statistics and machine learning. In the original paper, data was collected in 1970 and used in an analysis of the willingness of Boston area residents to pay for better air quality, based on an economic model and regression analysis. As discussed in Pace and Gilley (1997), it is usually analyzed without taking into consideration the fact that the data are spatially observed.

There is significant spatial clustering in every single variable in the dataset, as revealed by Moran's I test (using a similarity matrix calculated as the reciprocals of the spatial distances between observations, p-values for each variable were significant, thus rejecting the null hypothesis of no spatial clustering) and confirmed by HSIC, which was used to test for independence between the locations in space (using an RBF kernel) and each variable separately. In addition to adding spatial coordinates to each observation, Pace and Gilley (1997) also corrected a few errors in the original dataset.

The variables in the dataset are crime rate (crim), proportion of residential land zoned for lots over 25,000 sq. ft (zn), proportion of non-retail business acres per town (indus), indicator variable for whether tract bounds the Charles River (chas), nitric oxides concentration (nox), average number of rooms per dwelling (rm), proportion of owner-occupied units built prior to 1940 (age), weighted average of distances to five Boston employment centers (dis), index of accessibility to radial highways (rad), full-value property-tax rate (tax), pupil-teacher ratio

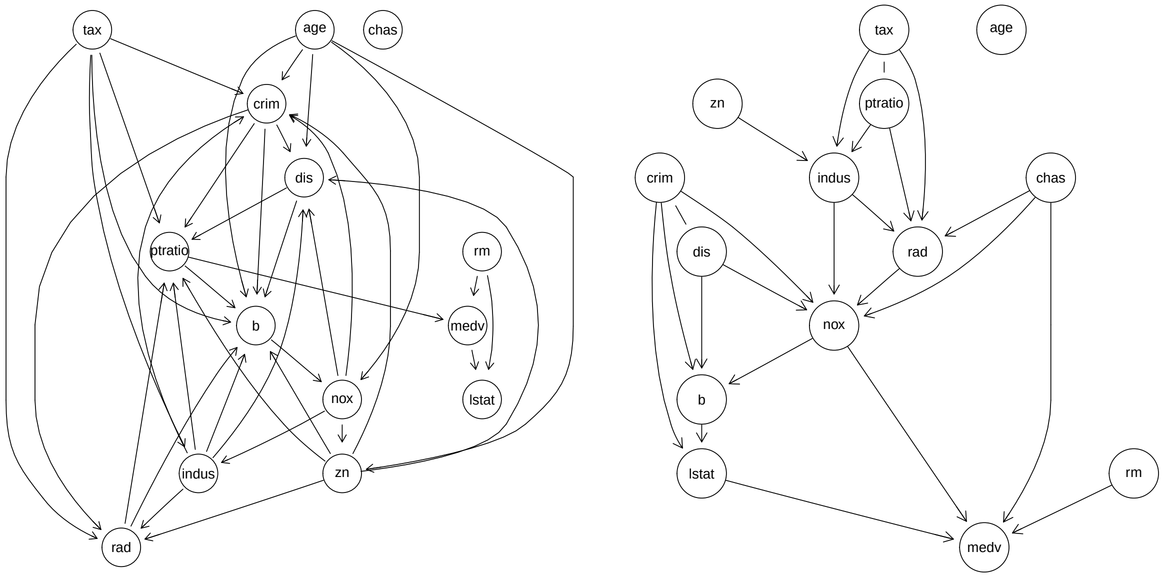


Fig. 6.2 Boston Housing data. Left: the PC algorithm was run on the data without pre-whitening (the data exhibits spatial autocorrelation), using GP/HSIC for conditional independence tests. The outcome variable of interest, median house value (medv) is caused by the number of rooms (rm) and the parent teacher ratio (ptratio), and it is a cause of the percentage of lower status people in the population (lstat). Other edge orientations seem dubious: nitric oxide concentration (nox), a measure of pollution causes industrial business activitity (indus), residential land zoned for large lots (zn), distance to employment centers (dis), and crime (crim). The substantive question in the original paper (Harrison Jr and Rubinfeld, 1978) was about the effect of pollution (nox) on house value (medv), but in the graph shown, there is no direct causal effect of pollution on house value. Right: after pre-whitening the data to remove spatial autocorrelation, the PC algorithm was run on it. The resulting causal graph has many fewer edges than the graph on the left. The outcome variable of interest, median house value (medv) is caused by percentange of lower status people in the population (lstat), number of rooms (rm), whether the tract bounds the Charles River (chas), and nitric oxide concentration (nox), a measure of pollution and the predictor variable of interest in the original paper (Harrison Jr and Rubinfeld, 1978). The graph shows that nitric oxide concentration (nox) is caused by industrial business activity (indus) (the opposite was found in the graph on the left), which is reasonable, but also by crime (crim) which seems unlikely.

by town (ptratio), polynomial transformation of proportion of blacks by town (b), percentage of lower status people in the population (lstat), and median value of owner-occupied homes (medv). The usual task with this dataset is to predict the median value of owner-occupied homes.

In the original analysis (Harrison Jr and Rubinfeld, 1978), the authors carefully state their prior, theoretical beliefs about the statistical (but not necessarily causal) relationship between each of the predictors in the dataset and the dependent variable. They included two “structural” variables which they expect to be related to home value, number of rooms and proportion of owner units built prior to 1940, eight neighborhood variables, two accessibility (in terms of transportation) variables, and two air pollution variables. Zhang et al. (2011) demonstrated KCI with the PC algorithm on this data. For the variable of interest, median value of house (medv), they found that number of rooms (rm), percentage of lower status people in the population (lstat), proportion of owner-occupied units built prior to 1940 (age), and crime rate (crim) are all parents of house value, with directed edges implying that these variables all cause house value.

We used the corrected dataset given in Pace and Gilley (1997). We ran the PC algorithm as implemented in the R package `pcalg` (Kalisch et al., 2012) using our new GP/HSIC approach for conditional independence with $\alpha = 0.001$. The results are shown in Figure 6.2. Throughout, we use the Gamma approximation to calculate p-values from HSIC. In this case, the outcome variable of interest, median house value, is caused by the number of rooms and parent teacher ratio and it is a cause of the percentage of lower status people in the population. There is no direct causal effect of pollution on house value.

Next, we pre-whitened each variable using the spatial coordinates with a GP regression in which the hyperparameters of the squared exponential (RBF) covariance function are learned by maximizing the marginali likelihood using gradient descent (the default in GPStuff (Vanhatalo et al., 2013)). Using this new dataset, we ran the PC algorithm again with $\alpha = 0.001$, as shown in Figure 6.2 (right). The resulting causal graph has many fewer edges. The percentage of lower status people in the population and number of rooms cause house value, as in Zhang et al. (2011). In addition, an indicator variable for whether the house is near the Charles River (which was not considered in Zhang et al. (2011)) also causes house value. Unlike the original graph in Figure 6.2 (left), we find that nitric oxide concentration, an indicator of air pollution, is a direct cause of house value, which addresses the original hypothesis explored by the authors in Harrison Jr and Rubinfeld (1978). Furthermore, nitric oxide concentration is now found to be caused by industrial business activity, rather than the converse when using unwhitened data. But we see that nitric oxide concentration is also apparently caused by crime, which seems unlikely.

6.3.2 Time series data

We consider the ozone dataset used in Breiman and Friedman (1985). This daily data clearly exhibits temporal autocorrelation, with 330 observations made over the course of 358 days. In Figure 6.3 (left) we show the results of the PC algorithm run on the data as is, with conditional independence tests using GP regression for conditioning and HSIC for independence testing.

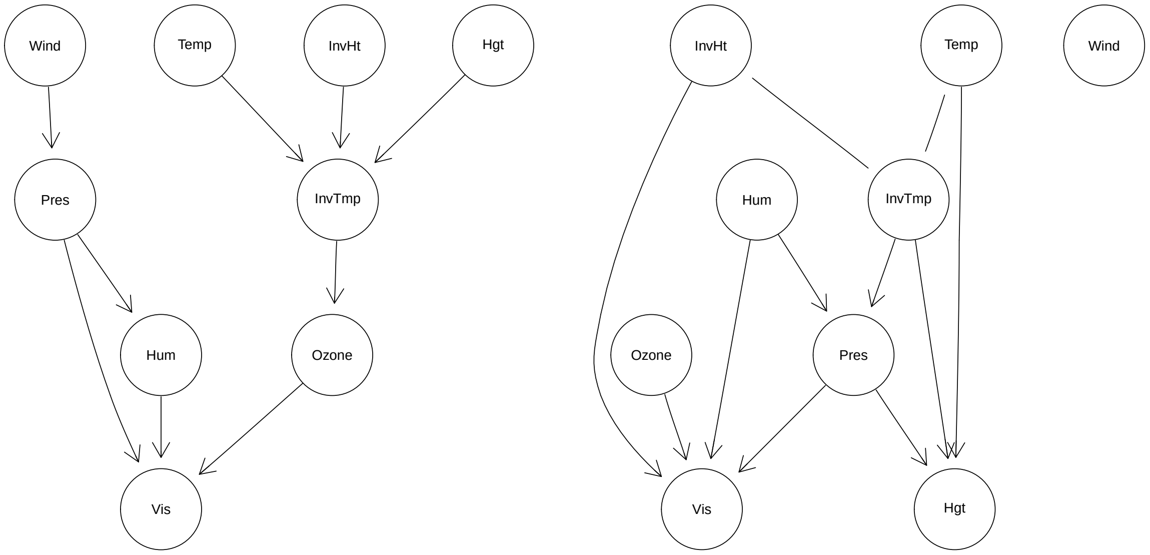


Fig. 6.3 Left: We used the PC algorithm with a dataset of environmental observations related to ozone in Upland, CA without first pre-whitening the time series data. The inferred causal CPDAG says that the temperature at the temperature inversion in the atmosphere (InvTmp) directly causes ozone, which causes visibility (Vis). Right: We pre-whitened the data using GP regression. Then we used the PC algorithm with the same dataset as previously. Ozone is still a cause of visibility (Vis), but no variables in the dataset were found to be a cause of ozone.

We set $\alpha = .05$ and used the standard version of the PC algorithm implemented in `pcalg` (Kalisch et al., 2012). We used the Gamma approximation to calculate p-values for HSIC.

The ozone variable is directly caused by the temperature at the temperature inversion in the atmosphere (InvTmp) and is a cause of visibility. In Figure 6.3 (right) contains the results of the PC algorithm run on the data after each variable has first been pre-whitened. To pre-whiten, we used GP regression with an exponential covariance function for time (which is analogous to an autoregressive fit), learning the hyperparameters from the data by maximizing the marginal likelihood with gradient descent. Now we see that the ozone variable has no parents, and is still a cause of visibility. Wind is no longer connected to any nodes, and two edges that were directed are no longer directed.

Next, we turn to the causal orientation (RESIT) framework for edge orientation, and consider one of the pairs of data[6] that our replication of Peters et al. (2014) showed was misoriented using the same method considered in that paper, Gaussian process regression followed by HSIC, comparing a forward and backward model. Pair 51 consists of daily ozone and temperature data from Switzerland, where the ground truth is that temperature causes ozone. As shown in Figure 6.4, there is an underlying time trend, and a partial autocorrelation plot

[6] http://webdav.tuebingen.mpg.de/cause-effect

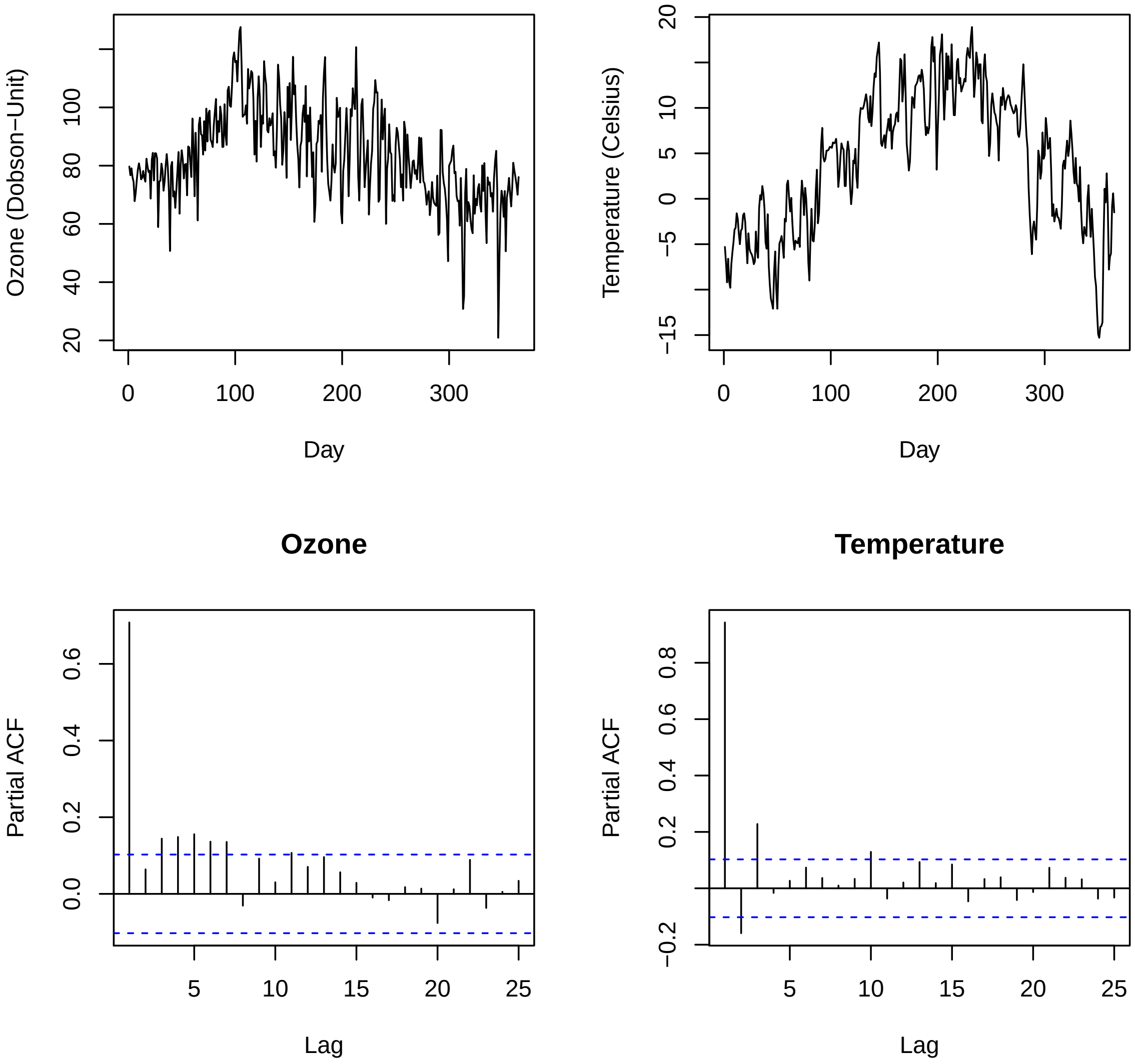


Fig. 6.4 Ozone and temperature data from Switzerland. A partial autocorrelation plot reveals significant temporal autocorrelation in both the ozone and temperature data. Before pre-whitening, bivariate causal orientation suggests incorrectly that ozone causes temperature. After pre-whitening, bivariate causal orientation correctly concludes that temperature causes ozone.

reveals temporal autocorrelation. Considering the data as is, the p-value of the forward model ("ozone causes temperature") is 0.002 and the p-value of the backwards model ("temperature causes ozone") is 4×10^{-7}. Thus, the causal orientation method fails, incorrectly predicting the forward model because it fits better. After pre-whitening, the p-values change. The forward model is still 0.002 but the backwards model is 0.34. The backwards model thus fits better and the edge is correctly oriented.

6.3.3 Textual Data

We consider a novel causal orientation problem: given pairs of translated sentences in two languages X and Y, determine whether X "causes" Y (meaning that the sentence in language Y was translated from the sentence in language X) or vice versa. We use the OpenOffice documentation corpus (Tiedemann, 2009) which consists of sentence-aligned documentation in English, French, Spanish, Swedish, German, and Japanese. We use our Gaussian process formulation for structured data to calculate residuals, and then we test whether these residuals are independent of the predictor, as in the RESIT framework. We use a spectrum kernel (also called a string kernel, the default in Karatzoglou et al. (2004)) which matches substrings of length $m = 3$.

The corpus is relatively large with 30,000-40,000 observations, so we use a bootstrap approach: for a pair of languages X, Y, we take a small sample ($n = 400$) and calculate a Gram matrix for the residuals R for the forward model X causes Y for half the sample ($n = 200$). Then we use HSIC to test whether R is independent of X on the other half of the sample ($n = 200$). We do the same for the reverse direction. We repeat this process 500 times with different subsets of the data, and report the fraction of times that we predict the forward direction based on comparing the p-values of the forward and reverse directions. (Larger p-values indicate better fits, so we accept the direction with the larger p-value.)

The results are in Figure 6.5. The OpenOffice documentation was originally written in German for its predecessor, StarOffice. When it was purchased by Sun Microsystems, the documentation was translated into English. Subsequently, translations were made from English to other languages, and new additions to the documentation were made in English[7]. Thus we consider English to be the "cause" of every other language except German. The algorithm correctly orients forward edges from English to every other language except German. The algorithm also orients forward edges from German to every language, which makes sense since German is a cause (though not direct) of every other language.

6.4 Conclusion

We proposed a simple, unified framework for coherently addressing the problem of algorithmic causal inference with non-iid observations, e.g., when data points are distributed in space and

[7] Uwe Fischer, personal communication, 9 July 2014.

time, and demonstrated its use on two real datasets. When using the PC algorithm or any other method based on independence tests, non-iid data presents a problem, and we showed how a pre-whitening step, using Gaussian process regression, can address this problem. We further showed how this same idea, of obtaining residuals from a GP regression, can be used to turn an unconditional independence test like HSIC into a conditional independence test.

We also showed that highly structured data, like text, can be considered in a causal framework, again using GP regression. In this case, we presented a novel formulation of a GP for structured inputs and outputs. The key derivation was that of the Gram matrix of the residuals, because once this is calculated, we can use HSIC to test independence.

HSIC is but one of the many measures of statistical independence which have been proposed. It might be fruitful to consider other measures instead, such as mutual information or distance correlation (Székely et al., 2009) or to determine whether the consistency results in Kpotufe et al. (2014) hold for our method. We do believe that Gaussian process regression is the most flexible and general tool for the purposes of pre-whitening non-iid data, due to its long-standing use in the spatial statistics and time series literature. In future work, we intend to look more deeply at the connections between GP regression and kernel-based measures of independence.

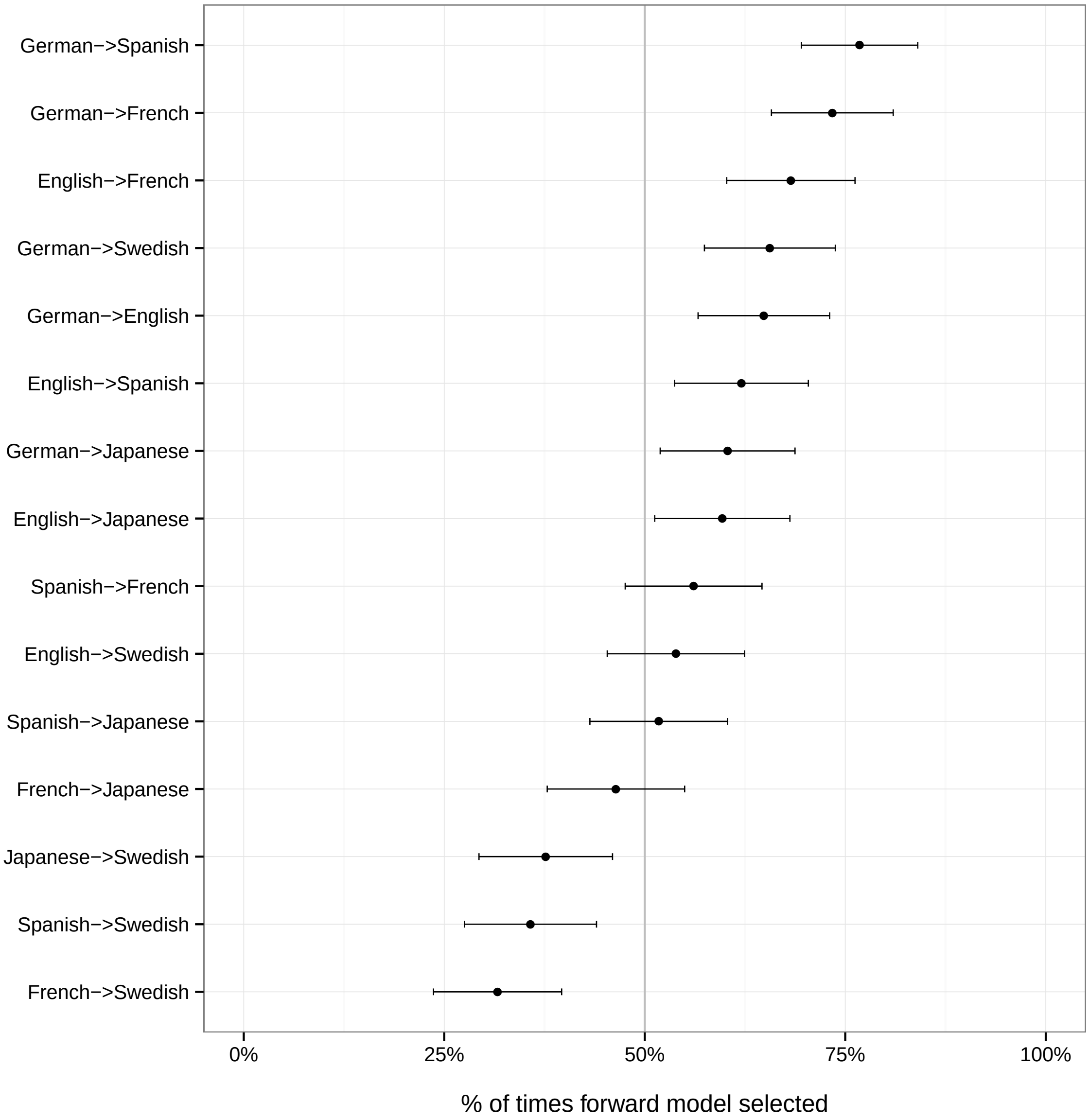


Fig. 6.5 Causal orientation with text: given pairs of sentences in two languages, the task is to determine which language is a translation of the other. We used a bootstrapping approach to repeatedly apply a causal orientation algorithm based on Gaussian processes for structured data and HSIC for independence testing to determine which language "causes" the other language. Shown are the fraction of times that the algorithm selected the forward causal direction, along with 95% confidence intervals. The top line, for example, means that in comparing German and Spanish, the algorithm concluded that German caused Spanish 77% of the time. The sentences come from OpenOffice documentation, portions of which were originally written in German and translated into English. After this one time translation, which occurred when Sun Microsystems bought what was then StarOffice, new documentation was written in English and English became the source language for translations into Spanish, Swedish, French, Japanese, and back into German. The algorithm thus correctly orients edges such that German and English are the cause of every other language. The algorithm definitively concludes that German causes English.

Chapter 7

Scalable, fully Bayesian spatiotemporal inference

In the previous chapters, I advocated the use of Gaussian process-based models as a general purpose method for spatiotemporal data, demonstrating their flexibility and utility in a variety of settings. Throughout, I mainly addressed the problem of hyperparameter learning by maximizing the marginal likelihood of the model, integrating out the latent GP (approximately, if necessary). In this chapter, I take a fully Bayesian approach to hyperparameter learning, placing priors on the kernel hyperparameters. The advantages of this approach are that the posterior uncertainty intervals should be more accurate, and the inference should be more robust to multimodality or likelihood surfaces with many local optima. The disadvantage is computational: by placing priors on hyperparameters we end up with non-conjugate models, so we need to turn to Monte Carlo sampling schemes for posterior inference. The central modeling choice with GPs is the specification of a kernel, and although a palette of kernel choices are available as catalogued in Section 2.2, it can be very hard to estimate a kernel from data, even with scientific knowledge. Prior distributions on kernel hyperparameters can help by more accurately capturing our uncertainty about the kernel.

In this chapter I address the computational challenges that stand in the way of a fully Bayesian approach to Gaussian process modeling. My approach works in the case of separable kernels and grid-structured inputs, which together induce structure in the design matrix. My approach is applicable to any Markov Chain Monte Carlo (MCMC) scheme for Bayesian inference, and I demonstrate its use within Hamiltonian Monte Carlo as implemented in the probabilistic programming language Stan. My implementation is much faster and uses much less memory than standard approaches, thus opening up many new avenues for scalable Bayesian modeling on orders of magnitude larger datasets (see Figure 7.1). By placing priors on kernel hyperparameters, our model becomes more general than standard GP regression or classification. To demonstrate, we show how to efficiently learn structured kernels (e.g. those with a multiplicative structure) and non-parametric priors over categorical variables. I propose a clustering through factor analysis method, address the limitation of separable structure, show how to automatically handle missing observations (i.e. for incomplete grids), and extend our methods for non-Gaussian observation models.

Efficiency gains from structured covariance functions within GP models have been exploited previously with MCMC (e.g. Finley et al. (2009)). There is much recent work on approximate methods for Gaussian processes and kernels (e.g. Hensman et al. (2013); Lázaro-Gredilla et al. (2010); Rue et al. (2009); Yang et al. (2015)). My work builds on previous work exploiting structured kernels and especially Kronecker methods for GP learning and inference (e.g. Flaxman et al. (2015d); Gilboa et al. (2013); Groot et al. (2014); Riihimäki and Vehtari (2014); Saatçi (2011); Stegle et al. (2011); Wilson et al. (2014)).

In Section 7.1 I provide background on MCMC, GPs, and Kronecker inference. In Section 7.2 I develop our scalable learning and inference methods. In Section 7.3 we demonstrate the novel modeling approaches that our methods enable. In Section 7.4 I compare our model's performance on synthetic data to standard implementations of elliptical slice sampling and HMC, and we demonstrate our novel modeling approaches on a real dataset. Implementations in Stan are provided in the Section 7.6.

7.1 Background

7.1.1 Markov Chain Monte Carlo sampling

Markov Chain Monte Carlo sampling schemes are methods for numerical inference in Bayesian models in which random samples are iteratively generated, where the limiting distribution of the samples is the posterior over the parameters in the model. For non-conjugate models, MCMC is the default inference method. In the hierarchical GP models I consider, with priors over kernel hyperparameters, the posterior is not a Gaussian process, which is why I use MCMC. A critical subroutine, executed each time a new draw is generated, is the evaluation of the log of the probability density of the posterior at the current values of the parameters, which can be very costly in a GP model as described below.

Another reason that MCMC inference for GP models is challenging is that the parameters in Gaussian process models are tightly correlated, so off-the-shelf methods like Metropolis-Hastings and Gibbs sampling, which are known to have slow convergence in the case of correlated parameters, have not proved effective[1]. Early work by Neal focused on Hamiltonian Monte Carlo (HMC) Neal (1996), a method drawn from the physics literature that uses gradient information to make sampling more efficient. More recent work has focused on variants of slice sampling, especially elliptical slice sampling Agarwal and Gelfand (2005); Murray and Adams (2010); Murray et al. (2010) which provides a variant on Metropolis-Hastings without tuning parameters by adaptively selecting the step size. My methods could be used in either of these schemes; to demonstrate the modeling advantages that our approach enables, I implement them using a probabilistic programming language (Stan), which uses HMC.

7.1.2 Gaussian processes

Given observations $(X,Y) = \{(x_1,y_1),\ldots,(x_n,y_n)\}$, let $k_{\boldsymbol{\theta}}(\cdot,\cdot)$ be a kernel with hyperparameters $\boldsymbol{\theta}$ and corresponding covariance matrix $K_{\boldsymbol{\theta}}$. Placing a prior on $\boldsymbol{\theta}$, the hierarchical specification is:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \tag{7.1}$$

$$\mathbf{f} \mid X, \boldsymbol{\theta} \sim \mathcal{N}(\mu, K_{\boldsymbol{\theta}}) \tag{7.2}$$

$$y_i \mid f(x_i) \sim \mathcal{N}(f(x_i), \sigma^2), \ \ \forall i \tag{7.3}$$

The computational difficulty of an MCMC scheme for this model arises from Eq. (7.2) which requires the computation of a multivariate Gaussian pdf:

$$\mathcal{N}(\mu, K_{\boldsymbol{\theta}}) = (2\pi)^{-n/2} |K_{\boldsymbol{\theta}}|^{-1/2} e^{-\frac{1}{2}\mu^\top K_{\boldsymbol{\theta}}^{-1}\mu} \tag{7.4}$$

Forming $K_{\boldsymbol{\theta}}$ takes storage $\mathcal{O}(n^2)$ and it takes time $\mathcal{O}(n^3)$ to calculate its inverse and log-determinant, using e.g. the Cholesky decomposition Rasmussen and Williams (2006). This costly operation occurs for each draw of a sampler, and HMC, it occurs for each "leapfrog" step, many of which are taken per draw.

I will primarily focus on the case of a Gaussian observation model. This will conveniently allow us to sidestep the issues that arise from trying to sample $\mathbf{f}$. For fixed hyperparameters, a GP prior with Gaussian observation model is conjugate. Usually this is used to find the posterior distribution over $\mathbf{f}$ in closed form. My setting is even simpler: we only need to have a way of calculating the likelihood of our observations y, integrating out $\mathbf{f}$. I have the following

[1] In a medium data spatial statistics setting, Diggle et al. [2013] report fitting a GP model with MCMC: after running for 18 million iterations, they retained every 18,000th iteration to yield a sample size of 1,000.

standard result Rasmussen and Williams (2006):

$$y \mid X, \boldsymbol{\theta}, \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, K_{\boldsymbol{\theta}} + \sigma^2 I) \tag{7.5}$$

7.1.3 Kronecker inference

A series of recent papers Flaxman et al. (2015d); Gilboa et al. (2013); Groot et al. (2014); Riihimäki and Vehtari (2014); Saatçi (2011); Wilson et al. (2014) has developed a set of inference techniques for the case of GP inference and prediction with separable covariance structures and inputs with a Cartesian product structure. I extend this line of work to an MCMC framework.

Assume that the covariance function $k(\cdot,\cdot)$ decomposes with Kronecker structure so that $k = k_1 \otimes k_2 \otimes \cdots \otimes k_d$ meaning that $k(x,x') = k_1(x_1,x'_1)k_2(x_2,x'_2)\cdots k(x_d,x'_d)$ for $x \in \mathcal{R}^d$. Further assume that we have a grid of input locations given by the Cartesian product $(x_1^1,\ldots,x_1^N) \times (x_2^1,\ldots,x_2^N)\cdots \times (x_d^1,\ldots,x_d^N)$ where for notational convenience we assume the grid has the same size N in each dimension. Then the covariance matrix K corresponding to $k(\cdot,\cdot)$ has $N^d \times N^d$ entries, but it can be calculated by first calculating the smaller $N \times N$ covariance matrices $K_1,\ldots,K_d$ and then calculating the Kronecker product $K = K_1 \otimes K_2 \otimes \cdots \otimes K_d$. The assumption that our data lies on a grid occurs naturally in various settings: images Wilson et al. (2014), spatiotemporal point patterns Flaxman et al. (2015d), and as we will illustrate below, time series and categorical data (where grid cells correspond to cells in a contingency table.)

We rely on standard Kronecker algebra results Saatçi (2011), specifically efficient Kronecker matrix-vector multiplication to calculate expressions like $(K_1 \otimes K_2 \otimes \cdots \otimes K_d)v$, efficient eigendecomposition of Kronecker matrices, and efficient Cholesky factorization of Kronecker matrices.

7.2 Theory

7.2.1 Inference

A key subroutine in any MCMC scheme is the evaluation of the log of the probability density function (pdf) of the posterior. In a GP model, this means calculating the pdf of a Gaussian distribution shown in Eq. (7.4). Following Saatçi (2011), I show how to efficiently evaluate the pdf in the case of a Kronecker-structured covariance matrix $K_{\boldsymbol{\theta}}$ and also in the case of a covariance matrix $K_{\boldsymbol{\theta}} + \sigma^2 I$, which will arise when I analytically integrate out **f**.

Working with the log of the pdf, and considering the case of $K = K_1 \otimes K_2$ (the extension to higher dimensions follows as in Saatçi (2011)), the log of Eq. (7.4) is:

$$-\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|K_1 \otimes K_2| - \frac{1}{2}y^\top (K_1 \otimes K_2)^{-1} y \tag{7.6}$$

for observations y, where I have assumed for simplicity that $\boldsymbol{\mu} = 0$. Applying standard Kronecker algebra results (see Section 2.3), I calculate:

$$\log|K_1 \otimes K_2| = N_2 \log|K_1| + N_1 \log|K_2| \tag{7.7}$$

where K_1 is $N_1 \times N_1$ and K_2 is $N_2 \times N_2$. If we let Y be the reshaped column-major $N_2 \times N_1$ matrix corresponding to y, so that $\mathrm{vec}(Y) = y$ (where vec stacks columns of a matrix), then we have:

$$(K_1 \otimes K_2)^{-1} y = K_1^{-1}(Y K_2^{-1})^\top \tag{7.8}$$

And we apply any standard linear solver to evaluate these matrix products. In general, for n training points on a d-dimensional grid, the time complexity is $\mathcal{O}(dn^{\frac{d+1}{d}})$ where $n = N^d$ and $N_1 = N_2 = \cdots = N_d = N$ Saatçi (2011) .

For a Gaussian observation model after integrating out $\mathbf{f}$, our sampler needs to be to evaluate the log of the pdf in Eq. (7.5). We can use eigendecomposition where $K_1 = Q_1^\top \Lambda_1 Q_1$, $K_2 = Q_2^\top \Lambda_2 Q_2$ gives:

$$K_1 \otimes K_2 = (Q_1^\top \otimes Q_2^\top)(\Lambda_1 \otimes \Lambda_2)(Q_1 \otimes Q_2) \tag{7.9}$$

and since the Q matrices are orthonormal:

$$K_1 \otimes K_2 + \sigma^2 I = (Q_1^\top \otimes Q_2^\top)(\Lambda_1 \otimes \Lambda_2 + \sigma^2 I)(Q_1 \otimes Q_2) \tag{7.10}$$

Thus we can calculate:

$$\log|K_1 \otimes K_2 + \sigma^2 I| = N_1 N_2 \sum_{ij} \log(\Lambda_{1ii}\Lambda_{2jj} + \sigma^2) \tag{7.11}$$

$$(K_1 \otimes K_2 + \sigma^2 I)^{-1} y = \Big((Q_1^\top \otimes Q_2^\top)(\Lambda_1 \otimes \Lambda_2 + \sigma^2 I)^{-1}(Q_1 \otimes Q_2)\Big) y \tag{7.12}$$

Eq. (7.11) follows because the eigenvalues of K_1 and K_2 are given by the diagonal of Λ_1 and Λ_2 and the computation is $\mathcal{O}(N_1 N_2)$ or in general $\mathcal{O}(n)$. For Eq. (7.12), we use a Kronecker matrix-vector product to calculate $(Q_1 \otimes Q_2)y$. The middle term $(\Lambda_1 \otimes \Lambda_2 + \sigma^2 I)^{-1}$ is diagonal, and multiplying a diagonal matrix by a vector is just the elementwise product. Finally, we have one more Kronecker matrix-vector product to calculate, as above. Eigendecomposition is $\mathcal{O}(N^3)$ for an $N \times N$ matrix. Additional speed improvements are available for the calculation of the smaller K_i matrices: e.g., FFT works well for stationary kernels and regular gridded input Ripley (2009) and Toeplitz methods work well for stationary kernels in one-dimension with regularly spaced inputs Cunningham et al. (2008).

7.2.2 Prediction

I extend the ideas introduced above to efficiently infer the posterior $p(\mathbf{f}^*|y,X,x^*)$ at a new location x^*. For a fixed $K_{\boldsymbol{\theta}}$ we have the following standard result Rasmussen and Williams (2006):

$$p(\mathbf{f}^*|y,X,x^*,\boldsymbol{\theta}) = \mathcal{N}(K^*_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \sigma^2 I)^{-1}y, K^{**}_{\boldsymbol{\theta}} - K^*_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \sigma^2 I)^{-1}K^{*\top}_{\boldsymbol{\theta}}) \tag{7.13}$$

where $K^*_{\boldsymbol{\theta}} = [k_{\boldsymbol{\theta}}(x^*,x_1),\ldots,k_{\boldsymbol{\theta}}(x^*,x_n)]$ and $K^{**}_{\boldsymbol{\theta}} = k_{\boldsymbol{\theta}}(x^*,x^*)$. A naive implementation would have time complexity $\mathcal{O}(n^3)$. To calculate the mean in Eq (7.13), we can again exploit Kronecker structure with the eigendecompositions in Eqs. (7.9) and (7.12):

$$K^*_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \sigma^2 I)^{-1}y = K^*_{\boldsymbol{\theta}}(Q_1^\top \otimes Q_2^\top)(\Lambda_1 \otimes \Lambda_2 + \sigma^2 I)^{-1}(Q_1 \otimes Q_2)y \tag{7.14}$$

We now apply Kronecker-matrix vector multiplication to the first two terms and the last two terms, and we are left with a vector times a diagonal matrix times a vector, which we can calculate efficiently through elementwise vector multiplication. Thus, the overall complexity is the same as for Eq. (7.12). For the variance term we have:

$$K^{**}_{\boldsymbol{\theta}} - K^*_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \sigma^2 I)^{-1}K^{*\top}_{\boldsymbol{\theta}} = K^{**}_{\boldsymbol{\theta}} - K^*_{\boldsymbol{\theta}}(Q_1^\top \otimes Q_2^\top)(\Lambda_1 \otimes \Lambda_2 + \sigma^2 I)^{-1}(Q_1 \otimes Q_2)K^{*\top}_{\boldsymbol{\theta}} \tag{7.15}$$

We use Kronecker matrix-vector multiplication twice to efficiently calculate the variance.

7.3 Modeling approaches

In this section, I demonstrate the advantages and flexibility of a fully Bayesian approach to GP modeling. I explore priors for categorical data and a low-rank factor analysis style model for clustering, demonstrate novel priors over hyperparameters for structured kernels, show how to infer missing data, and close with extensions for non-Gaussian observation models.

7.3.1 Kernel choice and priors over hyperparameters

The spatial statistics and machine learning literature provides a rich palette of scientifically and statistically motivated kernels from which to choose. As with any modeling choice, this could be informed by prior information, continuous model expansion Gelman and Shalizi (2013), or a nonparametric Wilson and Adams (2013b) approach could be taken. Because the problem of kernel learning is so difficult, including priors over the kernel hyperparameters to accurately characterize posterior uncertainty is very important.

Another advantage of placing priors over kernel hyperparameters is that the posterior distribution, which integrates out these parameters, is a *heavy tailed non-Gaussian process*. Following Agarwal and Gelfand (2005), we adopt weakly informative priors for inverse length-scales and variances which concentrate their support on reasonable values. We face an issue similar to the problem of applying HMC to neural networks Choo (2000): small changes in the inverse length-scales can result in orders-of-magnitude changes in the posterior. Experience suggests that more informative priors can contribute to sampling efficiency with HMC. My provisional suggestions for priors are listed below.

For a stationary kernel $k(x,x') = v^2 \kappa(|x-x'|_2 \lambda) + \sigma^2 I(x=x')$ where, e.g. $\kappa(d) = \exp(-d^2)$ is the RBF kernel or $\kappa(d) = \left(1 + d\sqrt{3}\right) \exp\left(-d\sqrt{3}\right)$ is the Matérn-$\frac{3}{2}$ kernel I suggest priors as follows:

- The inverse length-scale λ should have a weakly informative prior, like $\text{Cauchy}(0, 2.5)$ constrained to be positive (also known as the half-Cauchy).[2] I usually standardize our input locations to have standard deviation 1, but it is very important to untransform the learned length-scale to check that it is a reasonable value.

- The signal-variance v^2 should be about the scale of the data. In practice, I standardize our observed y to have standard deviation 1 so that I can place a $\text{log-normal}(0,1)$ prior on v^2.

- For computational reasons (equivalent to the jitter that is often added to the main diagonal) I constrain the nugget σ^2 to be greater than $\varepsilon = 10^{-6}$ and use a $\text{log-normal}(0,1)$ prior.

[2] An alternative worth considering is placing a Student-t with $\nu = 4$ centered at 0 with scale parameter 1 constrained to be positive on the *length-scale*. This is a more informative choice than the half-Cauchy.

7.3.2 Categorical data

Consider a multitask / co-kriging model where our dataset is structured as a real-valued observation y_i from category (task) c_i occuring at time t_i. I propose the following GP model:

$$y_i \sim \mathcal{N}(f(t_i, c_i), \sigma^2) \tag{7.16}$$

$$f(t, c) \sim \mathcal{GP}(0, K_t \otimes K_c) \tag{7.17}$$

We can immediately apply the methods developed in the previous sections to this setting, which is similar to the multitask approach in Bonilla et al. (2007). It remains to choose the kernel over categories. Let us express c_i as an indicator vector, where, e.g. for 3 categories we have the vectors $[1\ 0\ 0]^\top$, $[0\ 1\ 0]^\top$ and $[0\ 0\ 1]^\top$. Then we can use the covariance matrix K_c as the kernel:

$$k(c, c') = c^\top K_c c \tag{7.18}$$

The default choice of prior for a covariance matrix is the inverse Wishart distribution, which is conjugate, but has well-known drawbacks; various alternatives have been proposed and analyzed Tokuda et al. (2011). As an alternative, and because I only care about learning correlations, not covariances, between tasks, I use a recently proposed prior Lewandowski et al. (2009) which was termed the LKJ prior in Stan Development Team (2014). This prior has the advantage that for precision parameter $\alpha = 2$, each partial correlation has distribution $\text{Uniform}(-1, 1)$.

7.3.3 Factor analysis

Another natural extension if we want to cluster our m categories into p clusters is to use a low-rank factorization where $K_s = LL^\top + \sigma^2 I$ with $K_s \in \mathbb{R}^{m \times m}$ and $L \in \mathbb{R}^{m \times p}$ for $p \ll m$. This kernel has been called a "factor analysis" kernel Rasmussen and Williams (2006). I propose constraining each row of L to sum to 1 to represent a soft clustering model, where L_{ij} gives the degree of membership of x_i in group j. A natural choice of prior is the Dirichlet distribution. Denoting row i of L as L_i we have:

$$L_i \overset{iid}{\sim} \text{Dirichlet}(\alpha, \ldots, \alpha) \tag{7.19}$$

for a concentration parameter α which may be fixed or have its own prior.

If we have, for example, time series observations for each category and we want to cluster categories then we can use the Kronecker formulation: $K = K_t \otimes K_s$. The factorization above is convenient because we can readily obtain the eigendecomposition of Eq. (7.12) through singular value decomposition (SVD) of L. We can efficiently find the p singular values of L, which we denote $e_1, \ldots, e_p$. Then the m eigenvalues of K_s are $e_1^2 + \sigma^2, \ldots, e_p^2 + \sigma^2, \sigma^2, \ldots, \sigma^2$. Similarly, we can use the left-singular vectors from SVD to obtain the eigenvectors of K_s.

7.3.4 Additive models

Another structured model worth considering is an additive covariance model $K_s + K_t$, which is equivalent to $f \sim \mathcal{GP}(0, K_s) + \mathcal{GP}(0, K_t)$. We propose a very convenient way of emulating this model using a mixture modeling approach. We introduce a latent variable $z \sim \text{Bernoulli}(\pi)$. Then we have:

$$f \mid z = 0 \sim \mathcal{GP}(0, K_s) \tag{7.20}$$

$$f \mid z = 1 \sim \mathcal{GP}(0, K_t) \tag{7.21}$$

The goal is to obtain a very flexible model, without adding much computational burden. As shown in Section 7.6, the implementation is very straightforward after integrating out z.

7.3.5 Missing observations

Incomplete grids due to missing observations can be straightforwardly handled in the fully Bayesian framework (especially when using a probabilistic programming language): for any observation location x_i where we do not observe y_i, we treat y_i as a latent parameter. The key likelihood calculation in Eq. (7.6) remains the same, only now we mix together observed (and thus for the purposes of our sampler fixed) y_i's with missing y_i's which we must learn by sampling. Code is in Section 7.6.

7.3.6 Extensions to non-Gaussian observation models

Non-Gaussian observation models are very useful for generalized regression and classification. For example, classification uses the Bernoulli likelihood with logistic link function:

$$y \sim \text{Bernoulli}(\text{logit}(f(x))) \tag{7.22}$$

$$f \sim \mathcal{GP}(0, K_{\boldsymbol{\theta}}) \tag{7.23}$$

This model is no longer conjugate, so we cannot integrate out $\mathbf{f}$, but we can handle it in an MCMC framework. This has the effect of increasing the number of parameters in our model, and these parameters have strong correlations. To attempt to mitigate these correlations, I adopt the formulation of Christensen et al. (2006) which has been used in HMC in Vanhatalo and Vehtari (2007), introducing latent variables $z_1, \ldots, z_n$ based on the weight-space view of GPs:

$$z_1, \ldots, z_n \overset{iid}{\sim} \mathcal{N}(0,1) \tag{7.24}$$

Now we calculate $K_{\boldsymbol{\theta}}$ and its Cholesky decomposition L where $LL^T = K_{\boldsymbol{\theta}}$. Then we have:

$$\mathbf{f} := Lz \tag{7.25}$$

$$y_i \sim \text{Bernoulli}(\text{logit}(f(x_i))) \tag{7.26}$$

By introducing z we have avoided some computational difficulty as we no longer need to calculate the determinant of $K_{\boldsymbol{\theta}}$. But we still need to calculate K itself and its Cholesky decomposition, which is $O(n^3)$ time and $O(n^2)$ memory. Once again we can exploit two Kronecker algebra results. We calculate the $N \times N$ Kronecker matrices K_s and K_t and find their Cholesky decompositions L_s and L_t in $O(N^3)$ time. Since $K = LL^\top$ we have:

$$K = K_s \otimes K_t = (L_s \otimes L_t)(L_s^\top \otimes L_t^\top) \tag{7.27}$$

Now we need to calculate $\mathbf{f} = Lz = (L_s \otimes L_t)z$ for Eq. (7.25). Once again, we use efficient Kronecker matrix-vector multiplication. Now, rather than a costly multivariate Gaussian pdf, we only have to evaluate the much cheaper univariate Gaussian distribution in Eq. (7.24).

In practice, I have had difficulties using HMC to update both $z_1, \ldots, z_n$ and the hyperparameters simultaneously. A reasonable solution might be to follow the blocked approach of Agarwal and Gelfand (2005); Vanhatalo et al. (2013), wherein the hyperparameters are sampled with, e.g. HMC and then conditional on these hyperparameters the z_i are sampled, with HMC or another algorithm.

7.4 Experiments

I implemented our models using HMC in the probabilistic programming language Stan Stan Development Team (2014). This allowed me to easily try out different choices of priors and different modeling approaches. All source code is provided Section 7.6.

7.4.1 Synthetic data

I simulate from a Gaussian process on an $n \times n$ regular grid using a product of RBF kernels: $k((s,t),(s',t')) = e^{-4|s-s'|^2} e^{-|t-t'|^2}$ with spherical Gaussian noise $\sigma = 0.1$. A sample is shown

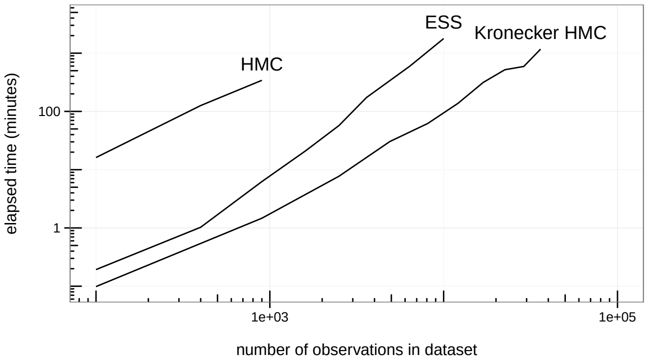


Fig. 7.1 My method (“Kronecker HMC”) was implemented in Stan, standard HMC and elliptical slice sampling were implemented in GPstuff. HMC was run for 200 iterations, with 100 iterations of warm-up, and elliptical slice sampling (ESS) for 30 iterations. Each method was compared on the same simulated datasets (Section 7.6.1).

in Figure 7.4. I compare our proposed Kronecker-based inference to non-Kronecker inference, both with HMC, and to elliptical slice sampling. As shown in Figure 7.1 our approach is much more efficient than the alternatives. We are not in a regime in which the $\mathcal{O}(n^3)$ asymptoptic time of the Cholesky decomposition dominates the computation, and there are many other factors which come into play in determining how long MCMC takes to run, but it is clear that our HMC approach is much faster, especially considering that I implemented it in a general purpose probabilistic programming language (Stan) rather than relying on custom code. Furthermore, the memory requirements for the non-Kronecker methods became prohibitively large in practice. As another comparison, I calculated the effective sample size Kass et al. (1998) for a dataset of size $n = 2{,}500$. My model generated an effective sample of 203 draws in 296 seconds or 0.69 samples per second. Elliptical slice sampling generated an effective sample of 221 draws in 3,438 seconds or 0.06 samples per second. Standard HMC generated an effective sample size of 51 samples in 31,364 seconds or 0.002 samples per second.

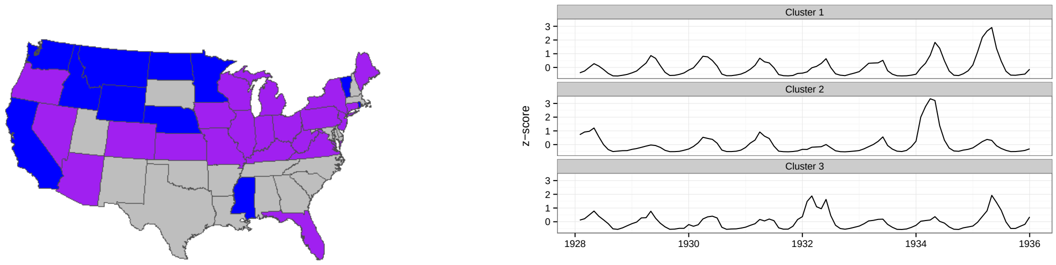
7.4.2 Real data

I obtained time series of monthly population-adjusted incidence of hepatitis A, measles, mumps, pertussis, and rubella for the 48 lower United States plus DC from Project Tycho[3].

I used our factor analysis formulation to cluster US states based on the time series of measles incidence. The separable covariance function was $K_t \otimes K_s$ where K_t was an RBF kernel and $K_s = \Lambda\Lambda^\top + \sigma^2 I$, with a Dirichlet$(0.1, 0.1, 0.1)$ on each row of $\Lambda \in \mathcal{R}^{49\times 3}$. The clustering of states is shown in Figure 7.2 (left): a geographic pattern is recovered, despite the model

[3] www.tycho.pitt.edu

not using geographic information. In Figure 7.2 (right) I show the posterior mean time series averaged for each cluster. Different dynamics are evident.



(a) (b)

Fig. 7.2 Left: clustering US states based on their time series of measles incidence. For each state I learned a 3-dimensional vector giving cluster assignment probabilities. I assign each state to its most probable cluster of the three, and shade it accordingly. Despite not using any geographic information in our model, I find a clear geographic pattern based on the similarities in time series. Right: mean posterior time series are shown for each cluster with evident differences in dynamics.

Finally, I considered the national time series for 4 different diseases from the Project Tycho dataset with a separable covariance $K_t \otimes K_c$ where K_t is as above and K_c is the categorical kernel in Eq. (7.18). I assign an LKJ prior with $\alpha = 2$ over the cross-type correlation matrix K_c. In Table 7.1 I show the posterior cross-type correlation matrix K_c. The lengthscale for K_t was 2 months $(1.9, 2.3)$ which corresponds to short-scale variation. Posterior plots are shown in Figure 7.3.

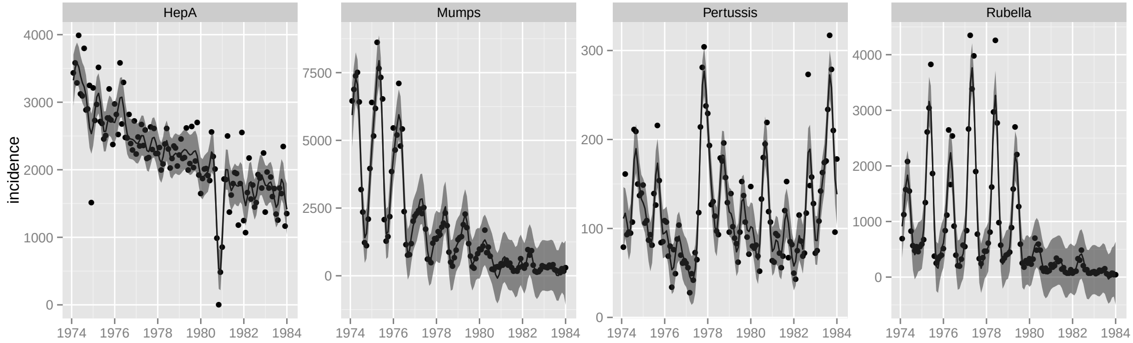


Fig. 7.3 Raw incidence across the United States of 4 types of infectious disease are shown as points, along with our model's estimates and 95% uncertainty intervals.

	Hepatitis A	Mumps	Pertussis	Rubella
Hepatitis A	1	0.6 (0.4,0.8)	-0.3 (-0.6,-0.1)	0.4 (0.1,0.6)
Mumps	0.6 (0.4,0.8)	1	-0.2 (-0.4,0.0)	0.6 (0.4,0.7)
Pertussis	-0.3 (-0.6,-0.1)	-0.2 (-0.4,0.0)	1	-0.2 (-0.5,-0.0)
Rubella	0.4 (0.1,0.6)	0.6 (0.4,0.7)	-0.2 (-0.5,-0.0)	1

Table 7.1 For the multitask model with covariance $K_t \otimes K_c$, I learned the posterior over a cross-task correlation matrix K_c. Medians and 95% UI intervals are stated. The corresponding lengthscale for K_t was 2 months $(1.9, 2.4)$ which corresponds to short-scale variation.

7.5 Conclusion

In this chapter I presented efficient inference methods for a Bayesian hierarchical modeling framework based on GPs, demonstrating the efficiency gains possible in the case of structured kernels as compared to standard approaches. This modeling approach enabled a variety of interesting and novel models, including learning cross-correlations between categories and factor analysis for spatiotemporal data. As my methods were implemented in a probabilistic programming language rather than in custom code, speed gains will no doubt result from dedicated code for some of the key Kronecker algebra, eigendecomposition, and gradient calculations. Further work is needed on the challenges of non-Gaussian observation models and on investigating other approximate methods for GPs like inducing points.

7.6 Source code

R code for generating large synthetic datasets and Stan code for the models in this chapter are given in this section.

7.6.1 Synthetic Data

To generate large synthetic datasets, I use the following R code:

```
library(kernlab)

eps = 1e-8
n = 200
space = seq(-2,2,length.out=n)
time = space

K1 = kernelMatrix(rbfdot(4), as.matrix(space))
```

```
K2 = kernelMatrix(rbfdot(1), as.matrix(time))
L1 = chol(K1 + eps * diag(n))
L2 = chol(K2 + eps * diag(n))

v = rnorm(n*n)
y = as.numeric(matrix(t(t(L2) %*% matrix(t(t(L1) %*% matrix(v,n,n)),n,n)),n*n,1))
y = y + rnorm(n*n,sd=.1)       # Add spherical noise

data = list(n1=length(space),n2=length(time), x1=space, x2=time, y=as.numeric(y))
```

A sample dataset is shown in Figure 7.4.

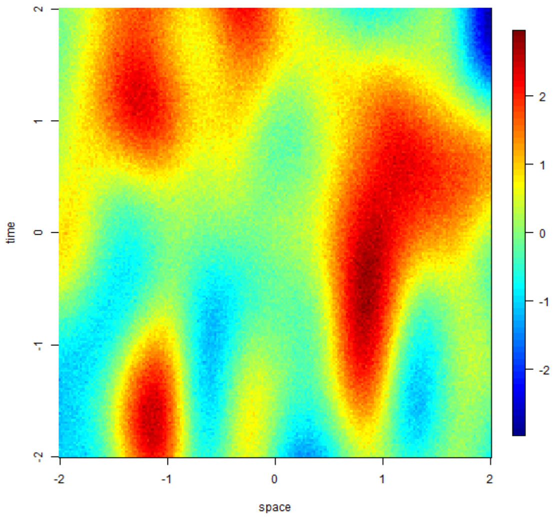


Fig. 7.4 A synthetic dataset with $n = 40,000$ observations generated by a GP with a separable covariance function $k((s,t),(s',t')) = e^{-4|s-s'|^2}e^{-|t-t'|^2}$.

7.6.2 Categorical data model

```
functions {
  // return (A \otimes B) v where:
  // A is n1 x n1, B = n2 x n2, V = n2 x n1 = reshape(v,n2,n1)
  matrix kron_mvprod(matrix A, matrix B, matrix V) {
    return transpose(A * transpose(B * V));
  }

  // A is a length n1 vector, B is a length n2 vector.
  // Treating them as diagonal matrices, this calculates:
  // v = (A \otimes B + sigma2)^{-1}
  // and returns the n1 x n2 matrix V = reshape(v,n1,n2)
  matrix calculate_eigenvalues(vector A, vector B, int n1, int n2, real sigma2) {
    matrix[n1,n2] e;
    for(i in 1:n1) {
      for(j in 1:n2) {
        e[i,j] <- (A[i] * B[j] + sigma2);
      }
    }
    return(e);
  }
}

data {
  int<lower=1> n1;
  int<lower=1> n2; // categories for learning cross-type correlations
  vector[n2] x1; // observation locations (e.g. timestamps)
  matrix[n2,n1] y; // NB: this should be reshape(y, n2, n1),
                 //   where y corresponds to expand.grid(x2,x1).
                 //   To double-check, make sure that y[i,j] is
                 //   the observation from category x2[i]
                 //   at location x1[j]
}

transformed data {
  matrix[n1, n1] xd;
  vector[2] one;
  one[1] <- 1;
  one[2] <- 1;

  for (i in 1:n1) {
    xd[i, i] <- 0;
    for (j in (i+1):n1) {
      xd[i, j] <- -(x1[i]-x1[j])^2;
      xd[j, i] <- xd[i, j];
    }
  }
}
```

```
parameters {
  real<lower=0> var1; // signal variance
  real<lower=0> bw1; // this is equivalent to 1/sqrt(length-scale)
  corr_matrix[n2] L;
  real<lower=0.00001> sigma1;
}

model {
  matrix[n1, n1] Sigma1;
  matrix[n1, n1] Q1;
  vector[n1] R1;
  matrix[n2, n2] Q2;
  vector[n2] R2;
  matrix[n2,n1] eigenvalues;

  Sigma1 <- var1 * exp(xd * bw1);
  for(i in 1:n1)
    Sigma1[i,i] <- Sigma1[i,i] + .00001;

  L ~ lkj_corr(2.0);

  Q1 <- eigenvectors_sym(Sigma1);
  R1 <- eigenvalues_sym(Sigma1);

  Q2 <- eigenvectors_sym(L);
  R2 <- eigenvalues_sym(L);

  eigenvalues <- calculate_eigenvalues(R2,R1,n2,n1,sigma1);

  var1 ~ lognormal(0,1);
  bw1 ~ cauchy(0,2.5);
  sigma1 ~ lognormal(0,1);
  increment_log_prob(
      -0.5 * sum(y .* kron_mvprod(Q1,Q2, // calculates -0.5 * y' (K1 \otimes K2) y
            kron_mvprod(transpose(Q1),transpose(Q2),y) ./ eigenvalues))
      -0.5 * sum(log(eigenvalues))); // calculates logdet(K1 \otimes K2)
}
```

7.6.3 Low-rank factorization model

```
functions {
  ... see first model above ...
}
data {
  int<lower=1> n1; // categories for clustering

  int<lower=1> n2;
  vector[n2] x2; // observation locations (e.g. timestamps)

  matrix[n2,n1] y; // NB: this should be reshape(y, n2, n1),
                   //   where y corresponds to expand.grid(x2,x1).
                   //   To double-check, make sure that y[i,j] is
                   //   the observation from category x1[j] at location
                   //   x2[i]
  int K;
}
transformed data {
  vector[K] alpha;
  matrix[n2,n2] xd;
  for(i in 1:n2) {
    xd[i,i] <- 0;
    for (j in (i+1):n2) {
      xd[i, j] <- -(x2[i]-x2[j])^2;
      xd[j, i] <- xd[i, j];
    }
  }

  for(i in 1:K)
    alpha[i] <- .1;
}
parameters {
  real<lower=0> var1;
  real<lower=0> bw2;
  real<lower=0.0001> sigma1;
  real<lower=0.0001> sigma2;
  simplex[K] Lambda1[n1];
}
transformed parameters {
  matrix[n1,K] Lambda1m;
  for(i in 1:n1) {
    Lambda1m[i] <- to_row_vector(Lambda1[i]);
  }
}

model {
  matrix[n1, n1] Sigma1;
  matrix[n2, n2] Sigma2;
  matrix[n1, n1] Q1;
  matrix[n2, n2] Q2;
  vector[n1] L1;
  vector[n2] L2;
  matrix[n2,n1] eigenvalues;

  for(i in 1:n1)
    to_vector(Lambda1[i]) ~ dirichlet(alpha);
  Sigma1 <- var1 * Lambda1m * transpose(Lambda1m);
  for (i in 1:n1)
    Sigma1[i] <- Sigma1[i] + sigma1;

  Sigma2 <- exp(xd * bw2);
  for (i in 1:n2) {
    Sigma2[i, i] <- Sigma2[i,i] + 0.000001;
  }

  Q1 <- eigenvectors_sym(Sigma1);
  Q2 <- eigenvectors_sym(Sigma2);
  L1 <- eigenvalues_sym(Sigma1);
  L2 <- eigenvalues_sym(Sigma2);

  eigenvalues <- calculate_eigenvalues(L2,L1,n2,n1,sigma2);
  bw2 ~ cauchy(0,2.5);
  var1 ~ lognormal(0,1);
  sigma1 ~ lognormal(0,1);
  sigma2 ~ lognormal(0,1);
  increment_log_prob(
      -0.5 * sum(y .* kron_mvprod(Q1,Q2, // calculates -0.5 * y' (K1 \otimes K2) y
            kron_mvprod(transpose(Q1),transpose(Q2),y) ./ eigenvalues))
      -0.5 * sum(log(eigenvalues))); // calculates logdet(K1 \otimes K2)
```

```
}
```

7.6.4 Synthetic data

```
functions {
  ... see first model above ...
}

data {
  int<lower=1> n1;
  int<lower=1> n2;
  vector[n1] x1;
  vector[n2] x2;
  matrix[n1,n2] y;
  real sigma2;
}

parameters {
  real<lower=0> bw1;
  real<lower=0> bw2;
  real<lower=0> var1;
}

model {
  matrix[n1, n1] Sigma1;
  matrix[n2, n2] Sigma2;
  matrix[n1, n1] Q1;
  matrix[n2, n2] Q2;
  vector[n1] L1;
  vector[n2] L2;
  matrix[n1,n2] eigenvalues;

  // these loops can be moved to the transformed data
  // block for efficiency, as in the source code in
  // the next section
  for (i in 1:n1) {
    Sigma1[i, i] <- var1;
    for (j in (i+1):n1) {
      Sigma1[i, j] <- var1 * exp(-(x1[i]-x1[j])^2*bw1);
      Sigma1[j, i] <- Sigma1[i, j];
    }
  }
  for (i in 1:n2) {
    Sigma2[i, i] <- 1;
    for (j in (i+1):n2) {
      Sigma2[i, j] <- exp(-(x2[i]-x2[j])^2*bw2);
      Sigma2[j, i] <- Sigma2[i, j];
    }
  }

  Q1 <- eigenvectors_sym(Sigma1);
  Q2 <- eigenvectors_sym(Sigma2);
  L1 <- eigenvalues_sym(Sigma1);
  L2 <- eigenvalues_sym(Sigma2);

  eigenvalues <- calculate_eigenvalues(L1,L2,n1,n2,sigma2);
  var1 ~ lognormal(0,1);
  bw1 ~ cauchy(0,2.5);
  bw2 ~ cauchy(0,2.5);
  sigma2 ~ lognormal(0,1);
  increment_log_prob( -0.5 * sum(y .* kron_mvprod(Q1,Q2,
       kron_mvprod(transpose(Q1),transpose(Q2),y) ./ eigenvalues))
       - .5 * sum(log(eigenvalues)));
}
```

7.6.5 Incomplete grids / missing observations

```
functions {
   ... see first model above ...
}
data {
  int<lower=1> n1;
  int<lower=1> n2;
  int<lower=0> nmissing;
  vector[n1] x1;
  vector[n2] x2;
  matrix[n2,n1] y;
  int x1missing[nmissing];
  int x2missing[nmissing];
}

transformed data {
  matrix[n1, n1] xd1;
  matrix[n2, n2] xd2;

  for(i in 1:n1) {
    xd1[i, i] <- 0;
    for (j in (i+1):n1) {
      xd1[i, j] <- -(x1[i]-x1[j])^2;
      xd1[j, i] <- xd1[i, j];
    }
  }
  for (i in 1:n2) {
    xd2[i, i] <- 0;
    for (j in (i+1):n2) {
      xd2[i, j] <- -(x2[i]-x2[j])^2;
      xd2[j, i] <- xd2[i, j];
    }
  }
}
parameters {
  real<lower=0> var1; // signal variance
  real<lower=0> bw1; // bandwidth
  real<lower=0> bw2;
  real<lower=0.00001> sigma1;
  vector[nmissing] ymissing;
}
transformed parameters {
  matrix[n2,n1] ystar;
  ystar <- y;
  for(i in 1:nmissing) {
    ystar[x2missing[i],x1missing[i]] <- ymissing[i];
  }
}

model {
  matrix[n1, n1] Sigma1;
  matrix[n2, n2] Sigma2;
  matrix[n1, n1] Q1;
  vector[n1] R1;
  matrix[n2, n2] Q2;
  vector[n2] R2;
  matrix[n2,n1] eigenvalues;

  Sigma1 <- var1 * exp(xd1 * bw1);
  for(i in 1:n1)
    Sigma1[i,i] <- Sigma1[i,i] + .00001;
  Sigma2 <- exp(xd2 * bw2);
  for(i in 1:n2)
    Sigma2[i,i] <- Sigma2[i,i] + .00001;

  Q1 <- eigenvectors_sym(Sigma1);
  R1 <- eigenvalues_sym(Sigma1);

  Q2 <- eigenvectors_sym(Sigma2);
  R2 <- eigenvalues_sym(Sigma2);

  eigenvalues <- calculate_eigenvalues(R2,R1,n2,n1,sigma1);

  var1 ~ lognormal(0,1);
  bw1 ~ cauchy(0,2.5);
  bw2 ~ cauchy(0,2.5);
  sigma1 ~ lognormal(0,1);
  increment_log_prob(
      -0.5 * sum(ystar .* kron_mvprod(Q1,Q2,
        kron_mvprod(transpose(Q1),transpose(Q2),ystar) ./ eigenvalues)) // calculates -0.5 *
      -0.5 * sum(log(eigenvalues))); // calculates logdet(K1 \otimes K2)
}
```

7.6.6 Mixture model

```
data {
  int<lower=1> n1;
  int<lower=1> n2;
  vector[2] x1[n1];
  vector[n2] x2;
  matrix[n2,n1] y;
}
transformed data{
  vector[2] one;
  matrix[n1,n2] yt;
  vector[n1] zero1;
  vector[n2] zero2;
  one[1] <- 1;
  one[2] <- 1;
  for(i in 1:n1)
    zero1[i] <- 0;
  for(i in 1:n2)
    zero2[i] <- 0;
  yt <- transpose(y);
}
parameters {
  real<lower=0> var1;
  real<lower=0> var2;
  vector<lower=0>[2] bw;
  real<lower=0.00001> sigma1;
  real<lower=0.00001> sigma2;
  real<lower=0> spacetime;
}

model {
  matrix[n1, n1] Sigma1;
  matrix[n2, n2] Sigma2;
  matrix[2,2] SpaceTime;
  real lp1;
  real lp2;

  SpaceTime[1,1] <- 10;
  SpaceTime[1,2] <- spacetime;
  SpaceTime[2,1] <- spacetime;
  SpaceTime[2,2] <- 10;

  for (i in 1:n1) {
    Sigma1[i, i] <- var1 + sigma1;
    for (j in (i+1):n1) {
      Sigma1[i, j] <- var1 * exp(-dot_self(x1[i]-x1[j])*bw[1]);
      Sigma1[j, i] <- Sigma1[i, j];
    }
  }
  for (i in 1:n2) {
    Sigma2[i, i] <- var2 + sigma2;
    for (j in (i+1):n2) {
      Sigma2[i, j] <- var2 * exp(-(x2[i]-x2[j])^2*bw[2]);
      Sigma2[j, i] <- Sigma2[i, j];
    }
  }

  bw ~ multi_normal(one,SpaceTime);
  spacetime ~ uniform(-1,1);
  var1 ~ lognormal(0,1);
  var2 ~ lognormal(0,1);
  sigma1 ~ lognormal(0,1);
  sigma2 ~ lognormal(0,1);
  pi ~ uniform(0,1);
  lp1 <- 0;
  for(i in 1:n2)
    lp1 <- lp1 + multi_normal_log(y[i],zero1, Sigma1);
  lp2 <- 0;
  for(j in 1:n1)
    lp2 <- lp2 + multi_normal_log(yt[j],zero2, Sigma2);
  increment_log_prob(log_mix(pi,lp1,lp2));
}
```

7.6.7 Trace plots and summary statistics

For the multitask model, we ran 4 chains for 600 iterations with 200 iterations of warm-up. The effective sample size was above 1000 for each parameter, with the Gelman-Rubin potential scale reduction statistic $\hat{R} \leq 1.01$. Trace plots are shown in Figure 7.5.

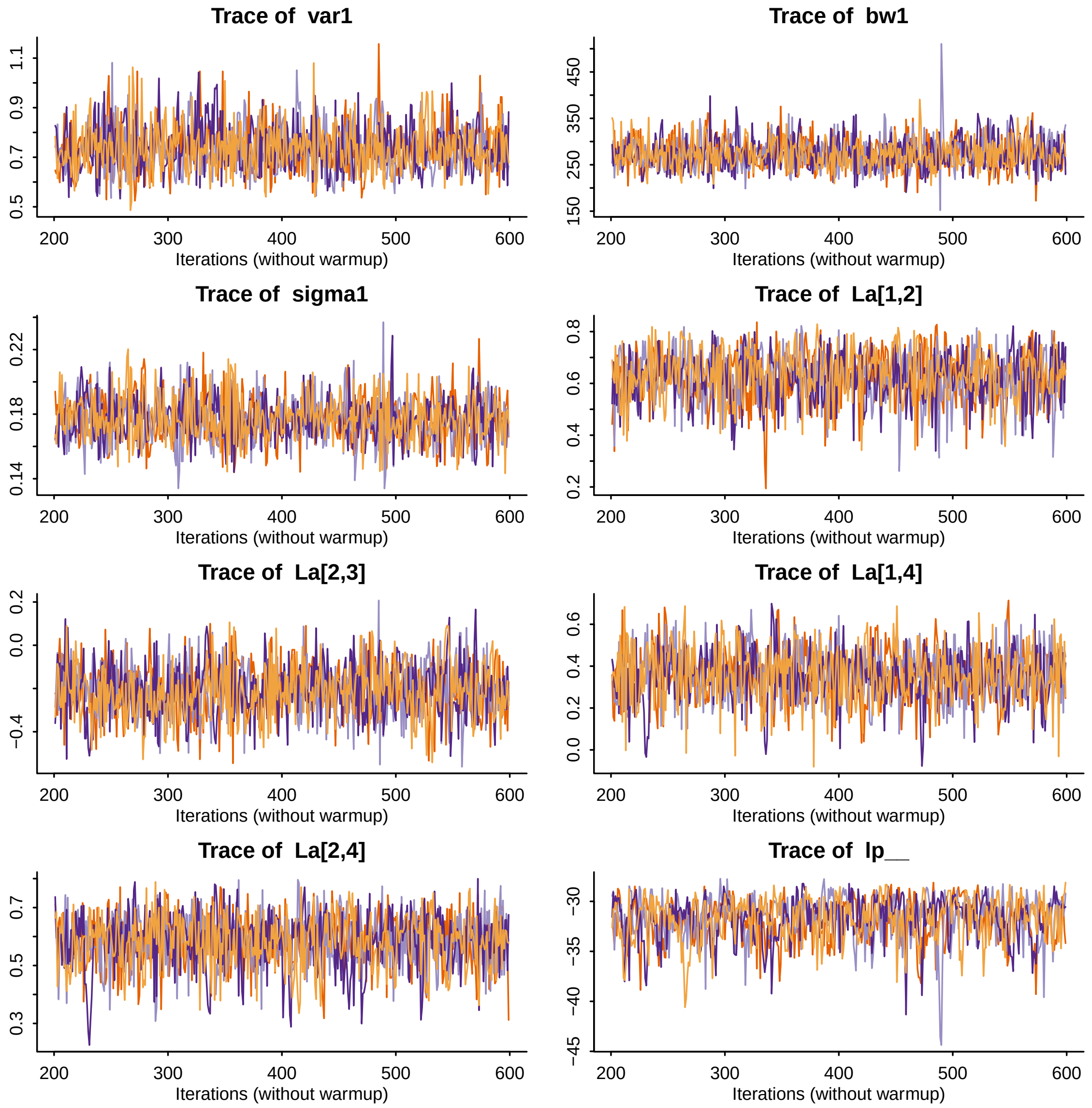


Fig. 7.5 Traces show good convergence for the 4 chains, each of which was run for 600 iterations after 200 iterations of warm-up.

Chapter 8

Conclusion

This thesis was motivated by the pressing public policy and social science questions my collaborators ask everyday, and it was driven by my desire to develop statistical machine learning methods to help them find answers to these questions. Machine learning has excelled in data rich domains on prediction tasks, but when I told our collaborators in the Chicago government that I had discovered novel associations in their crime data (that is, I could make good predictions) they immediately wanted more: how sure was I? Were the associations causal? Would the predictions hold in the future?

Answering these and related questions led me to work on difficult problems, some long-established within statistics but with only small sparks of attention within machine learning: spatiotemporal learning and inference, causal inference, spatiotemporal forecasting, and ecological inference. I hope that this thesis can contribute to more firmly establishing a field of spatiotemporal statistical machine learning. My approach emphasizes nonparametric Bayesian modeling based on Gaussian processes and kernel methods, but entirely different and fascinating approaches exist as well, e.g. Montanez and Shalizi (2015). I also hope that this thesis has asked novel questions which will inspire future research on relatively unexplored topics such as spatiotemporal causal inference.

The central body of work of this thesis was methodological: I presented my Kernel Space-Time interaction test in Chapter 3, new ways of scaling up inference for large-scale log-Gaussian Cox Processes in Chapter 4, a new ecological inference through distribution regression method in Chapter 5, a new conditional independence test, appropriate for spatiotemporal data in Chapter 6, and new hierarchical modeling and inference methods for fully Bayesian GP models in Chapter 7. Nevertheless, I have tried not to lose sight of the real-world applications driving these methods, and although I have relied on toy and synthetic datasets, I have tried whenever possible to use real datasets as well. Much more work lies ahead in testing these and related methods in the field. Machine learning's successes on prediction tasks are by now unmatched and the standard across a range of applications. But the future of statistical machine learning driven scientific discovery is wide open. I hope my methods will inspire practitioners and methodologists alike towards new discoveries.

References

Deepak K Agarwal and Alan E Gelfand. Slice sampling for simulation based fitting of spatial data models. *Statistics and Computing*, 15(1):61–69, 2005.

FE Alexander, P Boyle, PM Carli, JW Coebergh, GJ Draper, A Ekbom, F Levi, PA McKinney, W McWhirter, C Magnani, et al. Spatial temporal patterns in childhood leukaemia: further evidence for an infectious origin. euroclus project. *British journal of cancer*, 77(5):812, 1998.

Francis Bach. On the equivalence between quadrature rules and random features. *arXiv preprint arXiv:1502.06800*, 2015.

Francis R Bach and Michael I Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2003.

Rose D Baker. Testing for space-time clusters of unknown size. *Journal of Applied Statistics*, 23(5):543–554, 1996.

Matt A. Barreto, Fernando Guerra, Mara Marks, Stephen A. Nuño, and Nathan D. Woods. Controversies in exit polling: Implementing a racially stratified homogenous precinct approach. *PS: Political Science and Politics*, 39(3):pp. 477–483, 2006. ISSN 10490965. URL http://www.jstor.org/stable/20451787.

M. S. Bartlett. The detection of space-time interactions (commentary). *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 13(1):pp. 30, 1964. ISSN 00359254. URL http://www.jstor.org/stable/2985220.

Michel Besserve, Nikos K Logothetis, and Bernhard Schölkopf. Statistical analysis of coupled time series with kernel cross-spectral density operators. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2535–2543. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5079-statistical-analysis-of-coupled-time-series-with-kernel-cross-spectral-density-operators.pdf.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007.

George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis*. John Wiley & Sons, Inc., 2008. ISBN 9781118619193. doi: 10.1002/9781118619193.ch1. URL http://dx.doi.org/10.1002/9781118619193.ch1.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, 1985.

Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

Taeryon Choi and Mark J Schervish. On posterior consistency in nonparametric regression problems. *Journal of Multivariate Analysis*, 98(10):1969–1987, 2007.

Kiam Choo. *Learning hyperparameters for neural network models using Hamiltonian dynamics*. PhD thesis, Citeseer, 2000.

Ole F Christensen, Gareth O Roberts, and Martin Sköld. Robust markov chain monte carlo methods for spatial generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 15(1):1–17, 2006.

Kacper Chwialkowski and Arthur Gretton. A kernel independence test for random processes. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1422–1430, 2014.

N Cressie. Statistics for spatial data: Wiley series in probability and statistics, 1993.

N. Cressie and C.K. Wikle. *Statistics for spatio-temporal data*, volume 465. Wiley, 2011.

Noel Cressie and Hsin-Cheng Huang. Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association*, 94(448):1330–1339, 1999.

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

John P Cunningham, Krishna V Shenoy, and Maneesh Sahani. Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th international conference on Machine learning*, pages 192–199. ACM, 2008.

Peter J Diggle, Paula Moraga, Barry Rowlingson, Benjamin M Taylor, et al. Spatial and spatio-temporal log-gaussian cox processes: extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563, 2013.

P.J. Diggle, A.G. Chetwynd, R. Häggkvist, and SE Morris. Second-order analysis of space-time clustering. *Statistical methods in medical research*, 4(2):124–136, 1995.

AJ Dobson. An introduction to generalized linear models. *Chapman & Hall texts in statistical science*, 2002.

Gary Doran, Krikamol Muandet, Kun Zhang, and Bernhard Schölkopf. A permutation-based kernel conditional independence test. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI2014)*, pages 132–141, 2014.

O B Duncan and Davis B. An alternative to ecological correlation. *American Sociological Review*, 16:665–666, 1953.

David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1166–1174, 2013.

Miroslav Fiedler. Bounds for the determinant of the sum of hermitian matrices. *Proceedings of the American Mathematical Society*, pages 27–31, 1971.

Barbel Finkenstadt and Leonhard Held. *Statistical methods for spatio-temporal systems*. CRC Press, 2006.

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

Seth R Flaxman, Daniel B Neill, and Alex J Smola. Correlates of homicide: new space/time interaction tests for spatiotemporal point processes. *Heinz College working paper*, 2013. URL http://sethrf.com/files/space-time.pdf.

Seth R Flaxman, Andrew Gelman, Daniel B Neill, Alexander J Smola, Aki Vehtari, and Andrew Gordon Wilson. Fast hierarchical Gaussian processes. *Manuscript in preparation*, 2015a.

Seth R Flaxman, Daniel B Neill, and Alexander J Smola. Gaussian processes for independence tests with non-iid data in causal inference. *Provisional acceptance at ACM Transactions on Intelligent Systems and Technology (TIST)*, 2015b. URL http://www.sethrf.com/files/gp-depend.pdf.

Seth R Flaxman, Yu-Xiang Wang, and Alexander J Smola. Who Supported Obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–298. ACM, 2015c.

Seth R Flaxman, Andrew G Wilson, Daniel B Neill, Hannes Nickisch, and Alexander J Smola. Fast Kronecker inference in Gaussian processes with non-Gaussian likelihoods. *International Conference on Machine Learning 2015*, 2015d. URL http://www.sethrf.com/files/fast-kronecker-inference.pdf.

Ragnar Frisch and Frederick V Waugh. Partial time regressions as compared with individual trends. *Econometrica: Journal of the Econometric Society*, pages 387–401, 1933.

Montserrat Fuentes. Testing for separability of spatial–temporal covariance functions. *Journal of Statistical Planning and Inference*, 136(2):447–466, 2006.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Kernel dimensionality reduction for supervised learning. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, volume 16, page 81. MIT Press, 2004.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *NIPS*, volume 20, pages 489–496, 2007.

Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alex J Smola. Multi-instance kernels. In *ICML*, volume 2, pages 179–186, 2002.

Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.

Andrew Gelman, David Park, Boris Shor, Joseph Bafumi, and Jeronimo Cortina. *Red State, Blue State, Rich State, Poor State: Why Americans Vote the Way They Do*. Princeton University Press, August 2008. ISBN 069113927X.

Marc G Genton. Classes of kernels for machine learning: a statistics perspective. *The Journal of Machine Learning Research*, 2:299–312, 2002.

E. Gilboa, Y. Saatci, and J. Cunningham. Scaling multidimensional inference for structured gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP (99):1–1, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.192.

Elad Gilboa, John P Cunningham, Arye Nehorai, and Viktor Gruev. Image interpolation and denoising for division of focal plane sensors using gaussian processes. *Optics express*, 22 (12):15277–15291, 2014.

Tilmann Gneiting. Nonseparable, stationary covariance functions for space–time data. *Journal of the American Statistical Association*, 97(458):590–600, 2002.

Tilmann Gneiting, M Genton, and Peter Guttorp. Geostatistical space-time models, stationarity, separability and full symmetry. *Statistical Methods for Spatio-Temporal Systems*, pages 151–175, 2007.

Leo A Goodman. Some alternatives to ecological correlation. *American Journal of Sociology*, pages 610–625, 1959.

A. Gretton, K. Fukumizu, C.H. Teo, L. Song, B. Schoelkopf, and A. Smola. A kernel statistical test of independence. 2008. URL http://books.nips.cc/papers/files/nips20/NIPS2007_0730.pdf.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13:723–773, 2012.

Perry Groot, Markus Peters, Tom Heskes, and Wolfgang Ketter. Fast laplace approximation for gaussian processes with a tensor product kernel. In *Proceedings of 22th Benelux Conference on Artificial Intelligence (BNAIC 2014)*, 2014.

Mark S Handcock and Michael L Stein. A bayesian analysis of kriging. *Technometrics*, 35(4): 403–410, 1993.

David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

Larry D Haugh. Checking the independence of two covariance-stationary time series: a univariate residual cross-correlation approach. *Journal of the American Statistical Association*, 71(354):378–385, 1976.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. 2013.

Joseph M Hilbe. *Negative binomial regression*. Cambridge University Press, 2011.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *NIPS*, volume 21, pages 689–696, 2008.

Christopher Jackson, Nicky Best, and Sylvia Richardson. Improving ecological inference using individual-level data. *Statistics in medicine*, 25(12):2136–2159, 2006.

GEOFFREY M Jacquez. A k nearest neighbour test for space–time interaction. *Statistics in medicine*, 15(18):1935–1949, 1996.

Alfredo Kalaitzis, Antti Honkela, Pei Gao, and Neil D. Lawrence. *gptk: Gaussian Processes Tool-Kit*, 2013. URL http://CRAN.R-project.org/package=gptk. R package version 1.07.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. URL http://www.jstatsoft.org/v47/i11/.

Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL http://www.jstatsoft.org/v11/i09/.

Robert E Kass, Bradley P Carlin, Andrew Gelman, and Radford M Neal. Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

Gary King. *A Solution to the Ecological Inference Problem*. Princeton University Press, March 1997. ISBN 0691012407.

Gary King, Martin A Tanner, and Ori Rosen. *Ecological inference: New methodological strategies*. Cambridge University Press, 2004.

Melville R Klauber. Two-sample randomization tests for space-time clustering. *Biometrics*, pages 129–142, 1971.

E. G. Knox. The detection of space-time interactions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 13(1):pp. 25–29, 1964. ISSN 00359254. URL http://www.jstor.org/stable/2985220.

Samory Kpotufe, Eleni Sgouritsa, Dominik Janzing, and Bernhard Schölkopf. Consistency of causal inference under the additive noise model. 2014.

H. Kueck and N. de Freitas. Learning about individuals from group statistics. In *21st Uncertainty in Artificial Intelligence (UAI)*, pages 332–339, 2005.

M. Kulldorff. A spatial scan statistic. *Communications in statistics-theory and methods*, 26(6): 1481–1496, 1997.

Martin Kulldorff and Ulf Hjalmars. The knox method and other tests for space-time interaction. *Biometrics*, 55(2):544–552, 1999.

Hans R Kunsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, pages 1217–1241, 1989.

Imre Lakatos. The methodology of scientific research programmes. 1978.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.

Quoc Le, Tamas Sarlos, and Alex Smola. Fastfood: approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, 2013.

Pierre Legendre and Louis Legendre. *Numerical ecology*. Elsevier, 2012.

Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100 (9):1989–2001, 2009.

Colin Loftin. Assaultive violence as a contagious social process. *Bulletin of the New York Academy of Medicine*, 62(5):550, 1986.

H.J. Lynch and P.R. Moorcroft. A spatiotemporal ripley's k-function to analyze interactions between spruce budworm and fire in british columbia, canada. *Canadian Journal of Forest Research*, 38(12):3112–3119, 2008.

Nathan Mantel. The detection of disease clustering and a generalized regression approach. *Cancer research*, 27(2 Part 1):209–220, 1967.

Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

CharlesA. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986. ISSN 0176-4276. doi: 10.1007/BF01893414. URL http://dx.doi.org/10.1007/BF01893414.

Matthew W Mitchell, Marc G Genton, and Marcia L Gumpertz. Testing for separability of space–time covariances. *Environmetrics*, 16(8):819–831, 2005.

Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45, 1997.

J. Møller, A.R. Syversveen, and R.P. Waagepetersen. Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

Jesper Møller and Mohammad Ghorbani. Aspects of second-order analysis of structured inhomogeneous spatio-temporal point processes. *Statistica Neerlandica*, 66(4):472–491, 2012.

Alessio Moneta, Nadine Chlaß, Doris Entner, and Patrik O Hoyer. Causal search in structural vector autoregressive models. *Journal of Machine Learning Research-Proceedings Track*, 12:95–114, 2011.

George D Montanez and Cosma Rohilla Shalizi. The licors cabinet: Nonparametric algorithms for spatio-temporal prediction. *arXiv preprint arXiv:1506.02686*, 2015.

Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, pages 17–23, 1950.

Iain Murray and Ryan P Adams. Slice sampling covariance hyperparameters of latent gaussian models. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1732–1740. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4114-slice-sampling-covariance-hyperparameters-of-latent-gaussian-models.pdf.

Iain Murray, Ryan Prescott Adams, and David J.C. MacKay. Elliptical slice sampling. *JMLR: W&CP*, 9:541–548, 2010.

Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996. ISBN 0387947248.

New York Times. President exit polls - election 2012. http://elections.nytimes.com/2012/results/president/exit-polls, 2012. Accessed: 16 February 2015.

Junier B. Oliva, Willie Neiswanger, Barnabás Póczos, Jeff G. Schneider, and Eric P. Xing. Fast distribution to real regression. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Proceedings*, pages 706–714. JMLR.org, 2014.

Marek Omelka and Šárka Hudecová. A comparison of the mantel test with a generalised distance covariance test. *Environmetrics*, 24(7):449–460, 2013.

Stan Openshaw. Ecological fallacies and the analysis of areal census data. *Environment and Planning A*, 16(1):17–31, 1984.

R Kelley Pace and Otis W Gilley. Using the spatial configuration of the data to improve estimation. *The Journal of Real Estate Finance and Economics*, 14(3):333–340, 1997.

Christopher J Paciorek and Mark J Schervish. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5):483–506, 2006.

Giorgio Patrini, Richard Nock, Tiberio Caetano, and Paul Rivera. (almost) no label no cry. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 190–198. Curran Associates, Inc., 2014.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. ISBN 052189560X, 9780521895606.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1): 2009–2053, 2014.

Barnabas Poczos, Aarti Singh, Alessandro Rinaldo, and Larry Wasserman. Distribution-free distribution regression. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 507–515, 2013.

Ross L Prentice and Lianne Sheppard. Aggregate data studies of disease risk factors. *Biometrika*, 82(1):113–125, 1995.

Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *The Journal of Machine Learning Research*, 10:2349–2374, 2009.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2008.

Joseph D Ramsey. A scalable conditional independence test for nonlinear, non-gaussian data. *arXiv preprint arXiv:1401.5031*, 2014.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning, 2006.

Sashank Reddi and Barnabás Póczos. Scale invariant conditional dependence measures. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1355–1363, 2013.

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

B.D. Ripley. Edge effects in spatial stochastic processes. *Statistics in theory and practice*, pages 247–262, 1982.

Brian D Ripley. The second-order analysis of stationary point processes. *Journal of applied probability*, pages 255–266, 1976.

Brian D Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

W S Robinson. Ecological correlations and the behavior of individuals. *American Sociological Review*, 15:351–57, 1950.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

Yunus Saatçi. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2011.

K. Salkauskas. Some relationships between surface splines and kriging. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory II*, pages 313–325. Birkhauser, Basel, 1982.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. the MIT Press, 2002.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks ICANN'97*, pages 583–588. Springer, 1997.

César R Souza. Kernel functions for machine learning applications, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2001.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 99:1517–1561, 2010.

Stan Development Team. Stan: A c++ library for probability and sampling, version 2.5.0, 2014. URL http://mc-stan.org/.

W-H Steeb and Yorick Hardy. *Matrix calculus and Kronecker product: a practical approach to linear and multilinear algebra*. World Scientific, 2011.

Oliver Stegle, Christoph Lippert, Joris M Mooij, Neil D Lawrence, and Karsten M Borgwardt. Efficient inference in matrix-variate gaussian models with iid observation noise. In *Advances in neural information processing systems*, pages 630–638, 2011.

Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

Liangjun Su and Halbert White. A consistent characteristic function-based test for conditional independence. *Journal of Econometrics*, 141(2):807–834, 2007.

Z. Szabo, A. Gretton, B. Poczos, and B. Sriperumbudur. Two-stage Sampled Learning Theory on Distributions. *Artificial Intelligence and Statistics (AISTATS)*, February 2015.

Gábor J Székely, Maria L Rizzo, et al. Brownian distance covariance. *The annals of applied statistics*, 3(4):1236–1265, 2009.

Gabor J Szekely, Maria L Rizzo, et al. Partial distance correlation with methods for dissimilarities. *The Annals of Statistics*, 42(6):2382–2412, 2014.

Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.

Robert E Tillman, Arthur Gretton, and Peter Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *NIPS*, pages 1847–1855, 2009.

Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, pages 234–240, 1970.

Tomoki Tokuda, Ben Goodrich, I Van Mechelen, Andrew Gelman, and F Tuerlinckx. Visualizing distributions of covariance matrices. *Columbia Univ., New York, NY, USA, Tech. Rep*, 2011.

Aad Van Der Vaart and Harry Van Zanten. Information rates of nonparametric gaussian process methods. *The Journal of Machine Learning Research*, 12:2095–2119, 2011.

Jarno Vanhatalo and Aki Vehtari. Sparse log gaussian processes via mcmc for spatial epidemiology. In *Gaussian Processes in Practice*, pages 73–89, 2007.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *The Journal of Machine Learning Research*, 14(1):1175–1179, 2013.

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer Academic, 1998.

Christopher KI Williams and David Barber. Bayesian classification with gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

A. G. Wilson and R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning*, 2013a.

Andrew G Wilson and Ryan P Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1067–1075, 2013b.

Andrew Gordon Wilson, Elad Gilboa, Arye Nehorai, and John P. Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*. MIT Press, 2014. URL http://www.cs.cmu.edu/~andrewgw/manet.pdf.

Z. Yang, A.J. Smola, L. Song, and A.G. Wilson. A la carte - learning fast kernels. *Artificial Intelligence and Statistics*, 2015.

K. Zhang, J. Peters, D. Janzing, B., and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *27th Conference on Uncertainty in Artificial Intelligence*, pages 804–813, 2011.

Xinhua Zhang, Le Song, Arthur Gretton, and Alex J Smola. Kernel measures of independence for non-iid data. In *NIPS*, volume 22, 2008.