Applying Gaussian Processes to Air Pollution Data

Charles Li

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2018

(Graduation date: November 2018)

Abstract

This project explores the possibility of applying Gaussian processes to model and forecast NO_2 concentration in the U.K., utilising the hourly pollutant concentration data collected by the nationwide Automatic Urban and Rural Networks (AURN). In most experiments, models are trained and evaluated on a dataset of approximately 74,000 data points, except for one set of experiments where the size of the dataset is roughly 125,000. We address the scalability problem of conventional Gaussian processes by using scalable methods based on *inducing inputs*, covariance matrix structure exploitation or variational inference. The experiments are not all successful, but even the failed ones demonstrate the pattern recognition capabilities of Gaussian processes as well as their limitations, providing direction for future work. Our experiment with the task of forecasting exceedance shows that Gaussian processes' trend discovering ability can be used to alert the public to potentially bad air quality that may pose a threat to their health.

Acknowledgements

I am grateful to my supervisor, Dr. Chris Lucas, for his support, advice and guidance throughout the project. I would also like to thank all my family and friends for their support throughout my MSc.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Charles Li*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In classical dynamics, motions of micro-particles are governed by the diffusion equation:

$$\frac{\partial \phi(\boldsymbol{r},t)}{\partial t} = \nabla \cdot [D(\phi,\boldsymbol{r},t)\nabla\phi(\boldsymbol{r},t)], \tag{1.1}$$

where $\phi(\boldsymbol{r},t)$ is the density of the diffusing material at location $\boldsymbol{r}$ and time t, and $D(\phi,\boldsymbol{r},t)$ is the collective diffusion coefficient for density ϕ at location $\boldsymbol{r}$ and time t. We can further modify the equation to include a *source* term $s(\boldsymbol{r},t,\boldsymbol{x})$ and a *decay* term $\varepsilon(\boldsymbol{r},t,\boldsymbol{x})$ that represent the creation and decay rate of the material at location $\boldsymbol{r}$ and time t such that

$$\frac{\partial \phi(\boldsymbol{r},t)}{\partial t} = \nabla \cdot [D(\phi,\boldsymbol{r},t)\nabla\phi(\boldsymbol{r},t)] + s(\boldsymbol{r},t,\boldsymbol{x}) - \varepsilon(\boldsymbol{r},t,\boldsymbol{x}). \tag{1.2}$$

We use $\boldsymbol{x}$ to indicate all variables other than location and time that may have an influence.

In theory, the dispersion of air pollutants, such as nitrogen dioxide (NO_2), are dictated by Equation 1.2. Solutions to partial differential equations like 1.2 exist under certain smoothness and rectifiability assumptions of initial states and boundary. If we assume such a solution exists, we can write the density of NO_2 as a function of these variables:

$$\phi_{NO_2}(\boldsymbol{r},t) = f(\boldsymbol{r},t,\boldsymbol{x}) \tag{1.3}$$

where $\boldsymbol{x}$ can be many factors: temperature, pressure, traffic density, etc. In principle, if we can solve Equation 1.2 we will know the exact distribution of NO_2. Unfortunately, the exact solution is unattainable due to the sheer complexity of the atmospheric system and human activities (there are still numerical models based on particle diffusion, transport and other physical-chemical processes, which we discuss in Section 3.1).

Nevertheless, we can employ a machine learning algorithm to approximate such function by training on a collection of NO_2 concentration records.

Gaussian processes (GP) are one kind of *universal approximators*, in the sense that it can approximate a function arbitrarily well, if trained with enough data [Ghosal and Roy, 2006]. However, the 'enough data' condition has long been difficult to satisfy, as Gaussian processes are inherently plagued by computational complexity that scales cubically with the dataset size. This problem has been gradually addressed in recent years, thanks to advancements in scalable Gaussian processes that reduce the computational cost considerably. In some cases, Gaussian process models are applied to datasets containing a billion data points [Salimbeni and Deisenroth, 2017].

Readers familiar with other statistical/machine learning methods such as random forest, ARIMA (autoregressive integrated moving average), neural networks etc. may wonder what makes Gaussian processes worth considering. First and foremost, Gaussian processes are a class of Bayesian methods; apart from the predictions (in the form of predictive mean), they also quantify the uncertainty in their predictions, in the form of predictive variance. As we will see in Chapter 6, even if the predictions are not exact, the true target values sit in the confidence region inferred from predictive variance most of the time; we also make creative use of the predictive variance for a classification task. Secondly, properties of a Gaussian process is completely specified by its covariance function (Chapter 2), and we only need to find expressive covariance functions that induce inductive biases consistent with the dataset when designing Gaussian process models. Thirdly, as a Bayesian method, Gaussian processes require no cross-validation or regularisation thanks to the *complexity penalty* term (Section 2.3.2) in its objective function that arises naturally when performing optimisation.

In this project, we apply Gaussian process models to the air pollution data collected by the Automatic Urban and Rural Networks (AURN) in the U.K. We aim to discover the functional described by Equation 1.3, and to make sensible predictions about future NO_2 concentration in the air.

The outline of this dissertation is as follows:

- Chapter 2 gives a brief review of the air quality status in the U.K. and our data source, the Automatic Urban and Rural Networks, before introducing to readers

basic concepts in Gaussian processes;

- In Chapter 3, we discuss related work, both in air pollution modelling and in scalable Gaussian processes, and how they relate to or motivate our work;
- Chapter 4 covers details of each scalable Gaussian process models we use in our experiments;
- Chapter 5 contains information about our experiment setup, such as data preprocessing pipeline, hyperparameters and evaluation methods;
- We report and discuss our results in Chapter 6;
- We conclude our project and suggest potential future work in Chapter 7.

Chapter 2

Background

2.1 Air Quality in the U.K.

Clean air is vital to people's well-being. A range of air pollutants are known to have harmful impact on health as well as the environment. For example, nitrogen dioxide, or NO_2, the principal chemical compound of interest in this project, is shown to irritate the airways of the lungs and could worsen the symptoms of those already suffering from respiratory diseases, especially the elderly and infants. A research project [Walton et al., 2015] carried out by King's College London concluded that in 2010 alone, long-term exposure to nitrogen dioxide was estimated to be responsible for 5879 premature deaths in London.

In the U.K., air pollutants are mainly the products of combustion from space heating, power generation or from motor vehicle traffic [DEFRA, 2017]. Pollutants from these sources may not only prove a problem in the immediate vicinity of these sources but can travel long distances. Since the promulgation of the Environment Act in 1995, a National Air Quality Strategy [DEFRA, 2011] has been proposed and published that outlined policies for assessment and management of air quality. The Strategy has established objectives for eight key pollutants. Moreover, relevant EU legislations, such as the EU Ambient Air Quality Directive (2008/50/EC) [EU, 2008] and the 4th Air Quality Daughter Directive (2004/107/EC) [EU, 2004], set limit and target values for numerous pollutants in ambient air and require member states to report compliance and take action to correct any exceedance.

Limit values for NO_2 is set to:

- Hourly objective: 200 μg m^{-3} with no more than 18 exceedances per year; and
- Annual objective: 40 μg m^{-3}, not to be exceeded anywhere.

There are 43 zones in the U.K. for the purpose of ambient air quality reporting. By 2016, the 1-hour target had been met in all but two zones: Greater London Urban Are and South Wales. However, only six zones met the annual mean limit value for NO_2. One set of our experiments is about forecasting incidence of hourly exceedances of NO_2 in London; see Chapter 5 and 6 for details.

2.2 Automatic Urban and Rural Networks

After the introduction of the Clean Air Act in 1956, the U.K. became the first country in the world to establish a coordinated national air pollution monitoring network, the National Survey. At that time, the monitoring focus was on black smoke and sulphur dioxide. Over 60 years of observation has witnessed a steady decline in both black smoke and SO_2 concentration, and the focus has gradually shifted towards pollutants generated by vehicles. In 1987, an automatic urban monitoring network was set up to monitor compliance with the upcoming EC Directive limit values on air quality, and the Automatic Urban and Rural Network (AURN) was eventually formed in 1998 as a result of the combination of previously separate UK urban and rural automatic networks.

One hundred and sixty four monitoring sites are currently in operation, and they provide high resolution hourly measurement of a variety of air pollutants, including oxides of nitrogen (NO_x), sulphur dioxide (SO_2), ozone (O_3), carbon monoxide (CO) and particles (PM_{10}, $PM_{2.5}$). A typical entry of data collected by an AURN site can be found in Table 2.1.

As can be seen from Table 2.1, each monitoring site is characterised by its 'environment type' in addition to its coordinate. Depending on its proximity to built-up areas and to emission source of air pollutants, those sites can be classified into six categories:

- Urban area: continuously built-up urban areas populated completely (or at least highly predominantly) by street front side buildings of at least two floors or detached buildings with at least two floors; not mixed-up with non-urbanised

Table 2.1: Excerpted Data at Aberdeen Union Street Roadside Site. Unit is μg m^{-3}

Site Name	Aberdeen Union Street Roadside
Altitude (metres)	26
EU Site ID	GB0923A
Easting	393656
Environment Type	Urban Traffic
Government Region	North East Scotland
Latitude	57.1446
Longitude	-2.10647
Northing	805968
Site Address	Aberdeen
Site Code	ABD7
UK-AIR ID	UKA00513

Date	time	Nitric oxide	Nitrogen dioxide	Nitrogen oxides as nitrogen dioxide
01-01-2017	01:00	40.42998	45.52324	107.51501
01-01-2017	02:00	17.05059	24.56670	50.71057
01-01-2017	03:00	23.80441	37.98847	74.48805
01-01-2017	04:00	23.20228	25.25456	60.83089
01-01-2017	05:00	21.18321	29.54765	62.02811

areas with the exception of city parks. Measurement is representative of air quality within a few km^2;

- Suburban area: largely built-up urban area with contiguous settlement of detached buildings of any size; mixed with non-urbanised areas such as agricultural, lake, woods etc.. Measurement is representative of air quality within some tens of km^2;

- Rural area: sampling points targeted at the protection of vegetation and natural ecosystems that are far away from built-up areas, with unrestricted air flow. Measurement is representative of air quality in a surrounding area of at least 1000 km^2;

- Traffic station: location where the pollution level is mainly due to emissions from nearby traffic. Measurement is representative of air quality for a street segment at least 100m long;

- Industrial station: location where the pollution level is influenced principally by emissions from nearby industrial sources. Measurement is representative of air quality for an area of at least 250m $\times$ 250m;

- Background station: location where the pollution level is not due to any single source, but rather as a result of joint contributions from various sources upwind of the station. Representative for several km^2.

The main focus of this project is to model and predict NO_2 concentration data measured by AURN sites. Methodology for the selection of data points, as well as descriptive statistics, are detailed in Chapter 5. Our second and third sets of experiments assess if our GP models can predict the readings at any AURN site and any time; details are in Chapter 5 and 6.

2.3 Gaussian Processes

Materials in this section are mainly based on *Gaussian Processes for Machine Learning* by Carl Rasmussen and Chris Williams [Rasmussen and Williams, 2006].

2.3.1 Introduction

Gaussian processes (GP) provide a simple, unified yet powerful Bayesian non-parametric framework to learning in kernel machines. Originated from neural network research [Neal, 1996], Gaussian processes can be regarded as universal function approximators just like neural networks, in the sense that they define a distribution over functions. Concretely, it we want to model a function f, instead of guessing its functional form, we can model it as a collection of values $\{f(x_i)\}_{i=1}^N$ where x_i are some (input) values in the domain of f, and the correlation between any two $f(x_i)$ and $f(x_j)$ modelled by a *covariance function* (or *covariance kernel*, *kernel*) $k(x_i, x_j)$. Formally (as is defined in [Rasmussen and Williams, 2006]):

Definition 1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

The mean function and covariance function completely determine a Gaussian process. The mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ of $f(\mathbf{x})$ are defined as:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{2.1}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{2.2}$$

With these notation, we write the Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2.3}$$

The definition of Gaussian processes gives rise to the marginalisation property: if the GP specifies $(y_1, y_2) \sim N(\boldsymbol{\mu}, \Sigma)$, then $y_1 \sim N(\mu_1, \Sigma_{11})$ where Σ_{11} is a block submatrix of Σ. The marginalisation property is satisfied if entries of the covariance matrix Σ are given by the covariance function $k(\mathbf{x}, \mathbf{x}')$. Hence, we can represent the process f as

$$f \sim N(\mathbf{0}, K(X, X)) \tag{2.4}$$

where for simplicity, we assume the mean function to be zero everywhere, and $K(X, X)$ generated by $K_{ij} = k(x_i, x_j)$.

2.3.2 Inference and Learning

Suppose we are interested in modelling some noisy target y using a GP prior f defined over a set of observations X = $\{x_i\}$, and predicting the corresponding target y_* at a set

of inputs $X_* = \{x_{*i}\}$. The Bayesian approach

$$p(y_*|y) = \int p(y_*|f)p(f|y)df, \tag{2.5}$$

$$p(f|y) \propto p(y|f)p(f) \tag{2.6}$$

requires us to solve the integral which may not be analytically tractable. Fortunately, since we assume that each target value is modelled by a GP (and a Gaussian noise with variance σ^2), i.e.

$$y(x) = f(x) + \varepsilon(x), \tag{2.7}$$

$$\varepsilon(x) \sim N(0, \sigma^2) \tag{2.8}$$

all terms in equation are Gaussian and the integral can be solved analytically. We can show that y(x) is also a Gaussian process with covariance function $k(x_i, x_j) + \sigma^2\delta_{ij}$ where δ_{ij} is the Kronecker delta. The joint distribution over targets y, and the Gaussian process f_* evaluated at test locations X_*, is given by

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.9}$$

Having obtained the joint distribution, we can then condition on the observations X and y to obtain the predictive distribution

$$f_*|X, y, X_* \propto N(\bar{f}_*, \mathrm{cov}(f_*)), \tag{2.10}$$

where

$$\bar{f}_* = \mathbb{E}[f_*|X, y, X_*] = K(X_*,X)[K(X,X) + \sigma^2 I]^{-1} y, \tag{2.11}$$

$$\mathrm{cov}(f_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma^2 I]^{-1} K(X,X_*). \tag{2.12}$$

We use the compact notation $K = K(X,X)$, $K_* = K(X,X_*)$ and use $k(x_*)$ to represent the covariance vector in the case that there is only one test input. The predictive mean and variance become

$$\bar{f}_* = k_*^{\mathrm{T}}(K + \sigma^2 I)^{-1} y \tag{2.13}$$

which is merely a linear combination of observations y, with predictive variance

$$\mathrm{var}[f_*] = k(x_*,x_*) - k_*^{\mathrm{T}}(K + \sigma^2 I)^{-1} k_* \tag{2.14}$$

In this sense, Gaussian processes are a *linear predictor*. Despite the fact that we are modelling in the functional space (which implies GP can be represented in terms of a

possibly infinite number of basis functions), the predictive mean can nevertheless be written in the form of Equation 2.13, where the coefficients in the linear combination are completely determined by the covariance function $k(x,x')$. Thus, with purposely chosen covariance functions, a GP is fully capable of discovering the underlying structures of data and interpolating to a test set generated by the same function as the set of observations.

Training a GP model on dataset (X, y) is equivalent to finding the most suitable hyperparameters of the kernel function (if the kernel has already been chosen). In frequentist models such as linear regression, we often use maximal likelihood estimation (MLE) to find hyperparameters that fit the training dataset. The likelihood function $p(y|\theta)$ is a function of model (hyper)parameters θ that indicates how likely the observations y are conditioned on a certain set of θ. Likewise, we can define the *marginal likelihood* or probability density of the data y in the context of Gaussian process as

$$p(y|\theta,X) = \int p(y|f,X)p(f|\theta,X)df \tag{2.15}$$

where f is the GP, and θ represent hyperparameters of the covariance function. It is 'marginal' in the sense that we perform marginalisation on the likelihood function $p(y|f,X)$ over the GP f. With a Gaussian likelihood, the integral can be computed analytically such that

$$\log p(y|\theta,X) = \underbrace{-\frac{1}{2}y^{\mathrm{T}}(K+\sigma^2 I)^{-1}y}_{\text{model fit}} \underbrace{-\frac{1}{2}\log|K+\sigma^2 I| - \frac{N}{2}\log(2\pi)}_{\text{complexity penalty}}, \tag{2.16}$$

where N is the size of the training set, and K is an N $\times$ N covariance matrix. The marginal log likelihood (MLL) can be decomposed into two components: the first term, 'model fit', that resembles the form of the predictive mean describes how well the model fits the training data, and the remaining terms can be regarded as the 'complexity penalty'. Notice that only the model fit term depends on the training data, and the complexity penalty term decreases as the size of the training data increases.

We can see from equation that maximising MLL (sometimes called type-II maximum likelihood) is very different from MLE. In MLE, regularisation techniques are applied and a penalty term is often added to the likelihood function so that the model will not overfit, whereas a complexity penalty term is automatically included in the expression of MLL. Thus (in theory) there is no need to introduce additional regularisation terms,

or utilise regularisation techniques such as cross validation. However, we will see later that certain GP models can still overfit the training data and require careful calibration.

To maximise the MLL, we take partial derivative with respect to the hyperparameters θ:

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(y|X,\theta) &= \frac{1}{2}y^{\mathrm{T}}K^{-1}\frac{\partial K}{\partial\theta_j}K^{-1}y - \frac{1}{2}\mathrm{tr}(K^{-1}\frac{\partial K}{\partial\theta_j}) \\ &= \frac{1}{2}\mathrm{tr}((\alpha\alpha^{\mathrm{T}} - K^{-1})\frac{\partial K}{\partial\theta_j})\end{aligned} \tag{2.17}$$

where $\alpha = K^{-1}y$. We can then make use of gradient-based optimiser to reach maxima. The time complexity of the optimisation step is greatly influenced by the size of the covariance matrix, since its inversion normally requires time $O(n^3)$, and the computation of partial derivative only takes $O(n^2)$ per hyperparameter once K^{-1} is evaluated.

It is important to note that gradient-based optimisation does not always lead to global optima. Nevertheless, if we view different local optima as different possible interpretations of the dataset, some interpretations are bound to be more prescriptive than others. In practice, we find that we almost always reach the correct maxima after sufficiently many training iterations for large dataset.

2.3.3 Choice of Kernels

As we have already seen, covariance kernels are central to the representation power of Gaussian process models. To make the inversion of the covariance matrix K possible, valid kernel functions must map any set of inputs $\{x_i\}$ to a positive semi-definite covariance matrix such that $x^{\mathrm{T}}Kx \geq 0$ for all $x \in \mathbb{R}^N$. There are several ways of constructing new valid kernel functions based on existing ones. Here we list those pertaining to kernels used in this project:

$$k(x,x') = \alpha k_1(x,x') + \beta k_2(x,x') \tag{2.18}$$

$$k(x,x') = k_1(x,x')k_2(x,x') \tag{2.19}$$

$$k(x,x') = \alpha k_a(x_a,x'_a) + \beta k_b(x_b,x'_b) \tag{2.20}$$

$$k(x,x') = k_a(x_a,x'_a)k_b(x_b,x'_b) \tag{2.21}$$

where $\alpha,\beta > 0$, x_a and x_b are not necessarily disjoint variables with $x = (x_a,x_b)^{\mathrm{T}}$, and k_a and k_b are valid kernels in their respective spaces.

The *squared exponential* (SE) covariance function is perhaps the most widely used kernel in GP modelling. It has the form

$$k_{\mathrm{SE}}(x,x') = \sigma_f^2 \exp\left(-0.5||x-x'||^2/l^2\right) \tag{2.22}$$

where σ_f is the signal variance parameter, and l is the length-scale hyperparameter. σ_f controls the variability of sample functions, i.e. how deviated f is from the mean function, and l represents how fast GP functions change in input space.

The squared exponential kernel is sometimes called radial basis function (RBF) kernel, because a GP with an SE kernel is in fact equivalent to a RBF model with infinitely many basis functions. Just like multilayer perceptrons, GPs with SE kernel are also *universal approximators*: a GP with an SE kernel can approximate *any* function arbitrarily well if trained on sufficiently large amount of data. [Ghosal and Roy, 2006]

The SE kernel defined in Equation 2.22 is controlled by one length-scale hyperparameter only. Sometimes the dataset we deal with is multi-dimensional, and the scale of each dimension can be very different, in which case a single length-scale may become insufficient to properly model the dataset. To resolve this problem, we can make use of *automatic relevance determination*, or ARD, [MacKay, 1994], such that the SE kernel takes the form

$$k_{\mathrm{SE}}(x,x') = \sigma_f^2 \exp\left(-0.5 \sum_{i=1}^{n} |x_i - x_i'|^2/l_i^2\right) \tag{2.23}$$

where we have one characteristic length-scale hyperparameter per input dimension.

Time series data often comes with certain intrinsic periodicity, and we can combine kernels with the *periodic* kernel k_{PER} to better capture the structure. First derived in [MacKay, 1998], the periodic kernel is obtained by first using the transformation $x \mapsto (\cos(x), \sin(x))$, and then applying the SE kernel in the transformed space to get

$$k_{\mathrm{PER}}(x,x') = \exp(-2\sin^2(\frac{x-x'}{2})/l^2) \tag{2.24}$$

For example, Rasmussen and Williams [Rasmussen and Williams, 2006] used the product kernel of the period kernel and the squared exponential kernel to account for the periodicity of oscillations in the atmospheric CO_2 data. Our experiments in Chapter 6 also demonstrate the huge improvement that can be made by replacing SE kernel with more expressive product kernel.

2.3.4 The Role of the Mean Function

In all previous discussions, we assume the mean function to be zero. The mean function represents our prior knowledge about the dataset we model. In most cases, it is safer to assume that we know very little about the dataset and let the GP models do the work of pattern discovery. Still, if we can make educated guesses about the underlying mechanisms that generate the dataset, we can incorporate a parametrised mean function into our GP model, accompanied by a signal variance parameter like the σ_f^2 term in the SE kernel. Training then enables us to find parameters not accounted for by the mean function as well as the variability of the sample functions.

There are meticulously crafted models based on physical and chemical principles governing the creation and dispersion of air pollutants, which we describe in Chapter 3. Throughout this project, we do not specify any particular mean function for our Gaussian process models, as we discover that Gaussian processes can automatically detect the global empirical mean and site-type-specific means. See Chapter 6 for more details.

2.3.5 Flaws of Gaussian Processes

Despite their easily understood framework and strong representation power, Gaussian processes are not without shortcomings. Firstly, we cannot expect the same kernel function to be good at modelling any dataset. This means a considerable amount of time will be spent on observing the dataset and hand-crafting specific kernel functions appropriate to the task. Secondly, training a GP model necessitates the computation of $(K+\sigma^2 I)^{-1}$ for inference and $\log|K+\sigma^2 I|$ for hyperparameter optimisation. The standard Cholesky decomposition procedure for matrix inversion requires $O(n^3)$ operations and $O(n^2)$ storage, and computations of the predictive mean and variance require $O(n)$ and $O(n^2)$ for a single test input.

Substantial progress has been made to address these problems, and we will give a brief review on the most relevant approaches in the next chapter.

Chapter 3

Related Work and Motivation

3.1 Air Pollution Modelling

Due to the detrimental effect air pollution poses on human health, great efforts have been made to construct models capable of predicting air pollutant concentration and assessing health risks. One class of successful applications is based on land-use regression [Hoek et al., 2008], a method that integrates readings of air pollutant measured at a few monitoring locations (typically between 20 and 100) with probabilistic models that utilise predictor variables retrieved through geographic information system (GIS). Typical predictor variables are factors affecting pollutant emission and dispersion, e.g. traffic and population density, land use, terrain and climate. Land-use regression models are generally good at model long-term (for example, annual) [Gulliver et al., 2011] tendencies, and show good results in modelling annual mean concentrations of NO_2, NO_X and $PM_{2.5}$. However, the method is not suitable for short-term modelling.

Models intended for predicting short-term behaviour of air pollutants fall into two categories. One is statistical models that seek to impose spatiotemporal structure to collection of data gathered by static monitoring sites, often in the form of hierarchical Bayesian models. Successful applications of this statistical approach include modelling daily pollutant data at various sites [Shaddick and Wakefield, 2002], modelling ozone levels in cities [Huerta et al., 2004][Sahu et al., 2007], spatiotemporal modelling of fine particle matters [Cocchi et al., 2007], etc. Problems associated with statistical methods can be related to data quantity or quality; static monitoring sites are thinly distributed and unevenly spaced, and missing data can be a challenging problem for the modelling task.

Another approach is to simulate the process of pollutant emission and dispersion based on principles of underlying physical-chemical processes that take place in the atmosphere. This approach is not constrained by the sparsity of monitoring sites, and can obtain highly accurate predictions that extend over both space and time. However, accuracy of these simulated models relies on the precise characterisation of factors such as emission sources, weather and climate, and geographical features. Examples of air dispersion models include ADMS (Advanced Dispersion Modelling System) [Carruthers et al., 1994][McHugh et al., 1997], CMAQ (Community Multi-Scale Air Quality System) [Byun and Schere, 2006], and FRAME (Fine Resolution Atmospheric Multi-pollutant Exchange) [Singles et al., 1998].

Some researchers [McMillan et al., 2010] [Sahu et al., 2010] also suggest combining the statistical approach with air dispersion models. More specifically, we can use numerical outputs of air dispersion models as the primary contribution (or ambient measurement) to local pollutant concentration, with environment-specific effects described by Bayesian methods [Pirani et al., 2014]. This concerted approach eases the need to incorporate all relevant features into the air dispersion model, while producing more accurate predictions for monitoring sites of different types.

If we confine our interest to one particular area, we can model the air pollution by a temporal model, i.e. treating it as a time series forecast problem. Examples of this approach can be found in [Gocheva-Ilieva et al., 2014][Niska et al., 2004][Freeman et al., 2018]. However, temporal models cannot capture mid-to-long range behaviour of pollutants that contribute to local pollutant concentration; they also do not generalise to other places, whereas a spatiotemporal model can be used to produce predictions at any place and time.

3.1.1 Motivation

We run four sets of experiments in this project. Firstly, we apply Gaussian processes to model the entire air pollution dataset, as a preliminary validation step of our premise that Gaussian processes can perform learning on this particular dataset. Secondly, we run the experiments of spatial extrapolation, to assess if Gaussian process models can generalise to locations not contained in the training set. Thirdly, we investigate if the

spatiotemporal modelling approach has an advantage over a collection of individual, local temporal models. Lastly, we explore if Gaussian process models can be used to predict incidence of terrible outdoor air quality and alert the public.

Our second set of experiments, spatial extrapolation, are strongly motivated by [Pirani et al., 2014], where the authors demonstrated that Bayesian hierarchical models can predict PM_{10} concentration with high accuracy at new locations within Greater London area, when using predictor variables including time, coordinates, site environment type, day in a week, etc. The features we use in this project are largely in line with theirs. Apart from London, we also investigate if spatial extrapolation is possible at a much larger scale, say, the whole Great Britain. The success at a global scale depends on the model's ability to fuse knowledge about global ambience with site-specific local component, as shown by [Sahu et al., 2010][Pirani et al., 2014]. Gaussian processes' trend discovering ability, as shown in [Rasmussen and Williams, 2006], coupled with features like site environment type that reflect the nature of local pollutant emission, will enable GP models to succeed.

Our third set of experiments are inspired by the spatiotemporal versus temporal model discussion. Our spatiotemporal GP model will compete against a collection of individual-site level temporal models. Our last set of experiments stem from the real need to predict such event to protect public health, and are built upon our third set of experiments.

3.2 Scalable Gaussian Processes

It is tempting to find more efficient ways of inverting the matrix $K+\sigma^2 I$ or finding the solution to the linear equations $(K+\sigma^2 I)\mathbf{x}=\mathbf{y}$ for $\mathbf{x}$. If the n×n matrix K is not of full-rank, it decomposes into $K=QQ^{\mathrm{T}}$ where Q is an n×q matrix, and $K+\sigma^2 I$ factorises into

$$(QQ^{\mathrm{T}}+\sigma^2 I)^{-1}=\sigma^{-2}I-\sigma^{-2}Q(\sigma^2 I+Q^{\mathrm{T}}Q)^{-1}Q^{\mathrm{T}} \tag{3.1}$$

by *matrix inversion lemma* (Appendix A), reducing the problem to the inversion of a q×q matrix. Even if K is of rank n, we can still consider replacing K with lower-rank approximations, or designing kernel functions such that the covariance matrix has favourable factorisation property. These two approaches are reviewed in chapter 3.2.1 and 3.2.2 respectively.

3.2.1 Inducing Point-Based Methods

Subset of Regressors (SoR) [Silverman, 1985] is perhaps one of the earliest sparse Gaussian process approximation methods. It originated from the observation that the *mean* GP predictor (not applicable to variance) $m(x)$ is equivalent to a finite-dimensional generalised linear model (GLM)

$$\begin{aligned} f(x_*) &= \sum_{i=1}^{n} \alpha_i k(x_*, x_i) \\ \alpha &\sim N(0, K^{-1}) \text{ (the prior).} \end{aligned} \tag{3.2}$$

We can approximate this GLM by only considering the first m regressors $\{\alpha_i\}_{i=1}^{m}$, so that

$$\begin{aligned} f_{\mathrm{SoR}}(x_*) &= \sum_{i=1}^{m} \alpha_i k(x_*, x_i) \\ \alpha_m &\sim N(0, K_{mm}^{-1}). \end{aligned} \tag{3.3}$$

It can be shown that the SoR approximation is equivalent to replacing the exact kernel $k(x,x')$ with an approximate kernel $\tilde{k}(x,x') = k(x)^{\mathrm{T}} K_{mm}^{-1} k(x')$. Subsequently, the MLL to be maximised becomes

$$\log p_{\mathrm{SoR}}(y|\theta, X) = -\frac{1}{2} y^{\mathrm{T}} (\tilde{K} + \sigma^2 I)^{-1} y - \frac{1}{2} \log |\tilde{K} + \sigma^2 I| - \frac{n}{2} \log (2\pi) \tag{3.4}$$

$\tilde{K}$ is the Nyström approximation [Williams and Seeger, 2001] to K

$$\tilde{K} = K_{nm} K_{mm}^{-1} K_{mn} \tag{3.5}$$

and K_{mn} represents the top m×n block matrix of K and K_{nm} represents its transpose.

Another influential method is the fully independent training conditional (FITC) approximation [Snelson and Ghahramani, 2006], where the kernel function is approximated by

$$\tilde{k}_{\mathrm{FITC}}(x,x') = \tilde{k}_{\mathrm{SoR}} + \delta_{x,x'}(k(x,x') - \tilde{k}_{\mathrm{SoR}}). \tag{3.6}$$

We can regard this method as adding a 'diagonal correction' to the SoR approximation. The estimated covariance matrix is identical to the SoR covariance matrix, except for the diagonal elements which are the same as the diagonal of the original covariance matrix. Therefore, FITC tends to be more accurate than SoR and is preferred in practice.

There are a multitude of other approximation methods:

- The Subset of Datapoints method [Lawrence et al., 2003] which keeps the GP predictor, but only on a smaller subset of size m of the data;
- The Projected Latent Variable method [Seeger, 2003] that relies on likelihood approximation $p(y|f_m) \simeq N(y|K_{nm}K_{mm}^{-1}f_m, \sigma^2 I)$ where f_m represent the values of f that are evaluated explicitly;
- The Bayesian Committee Machine [Tresp, 2000] method that approximates the likelihood function by partitioning the dataset and assuming conditional independence such that $p(y_1,...,y_p|f_*,X) \simeq \prod_{i=1}^{p} p(y_i|f_*,X_i)$, under which we have

$$q(f_*|D_1,...,D_p) \propto p(f_*)\prod_{i=1}^{p} p(y_i|f_*,X_i) \propto \frac{\prod_{i=1}^{p} p(f_*|D_i)}{p^{p-1}(f_*)}$$

Quiñonero-Candela and Rasmussen [Quiñonero-Candela and Rasmussen, 2005] proposed a unifying view for all aforementioned methods by introducing the concept of *inducing variables* $u = [u_1,...,u_m]^{\mathrm{T}}$, latent variables that are values of the Gaussian process as f and f_*, corresponding to a set of *inducing inputs* X_u. They pointed out that due to the marginalisation property (discussed in Chapter 2) of Gaussian processes, we can recover the joint distribution $p(f, f_*)$ by marginalising over u

$$p(f, f_*) = \int p(f, f_*, u)du = \int p(f, f_*|u)p(u)du. \tag{3.7}$$

where $p(u) = N(0, K_{u,u})$. We can also approximate the joint prior $p(f, f_*)$ by assuming that f and f_* are conditionally independent given u such that

$$p(f, f_*) \simeq q(f, f_*) = \int q(f|u)q(f_*|u)p(u)du. \tag{3.8}$$

They explained that the name *inducing* variable was meant to convey the fact that u *induces* dependencies between training and test cases. They also demonstrated how various sparse approximation methods can be recovered by assuming different forms of the conditionals $q(f|u)$ and $q(f_*|u)$.

All inducing point-based methods simplify the task of inverting an n $\times$ n matrix into inverting an m $\times$ m one. They cost $O(m^2n)$ computations and $O(mn)$ storage for learning and inference (assume $m < n$), and evaluating predictive mean and variance take $O(m)$ and $O(m^2)$ per test input.

Inducing point-based methods are in a somewhat awkward position: to observe the

improvement in efficiency, one is restricted to only use a small $m \ll n$ for large dataset, whereas larger datasets clearly require more inducing points to capture their rich structure.

3.2.2 Exact Inference via Structure Exploitation

A different approach is to exploit the existing structure of the covariance matrix, including the Toeplitz structure [Wilson, 2014] and the Kronecker structure [Saatchi, 2011].

Toeplitz covariance matrices are generated by *stationary* kernels satisfying $k(x,x') = k(x-x')$ with inputs x on a regularly spaced one dimensional grid. As a result of the grid structure of inputs, Toeplitz matrices are constant along diagonals: $K_{i,j} = K_{i+1,j+1}$. The linear system $(K+\sigma^2 I)x = y$ can be solved in $O(m \log m)$ operations and $O(m)$ storage, for m grid datapoints.

The *Kronecker product* of two matrices A (of size m $\times$ n) and B (of size p $\times$ q) is a block matrix $A \otimes B$ of size mp $\times$ nq:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{3.9}$$

Suppose that we have multidimensional inputs on a Cartesian grid, $x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_P$ with $m = \prod_{i=1}^{P} n_p$ total grid points (n_p is the number of grid points in dimension p), and a product kernel that factorises across grid dimensions, $k(x,x') = \prod_{p=1}^{P} k(x^{(p)}, x'^{(p)})$, then the covariance matrix can be written as a Kronecker product $K = K_1 \otimes \cdots \otimes K_P$. This enables us to find the eigendecomposition $K = QVQ^{\mathrm{T}}$ where V is a diagonal matrix of eigenvalues, by eigendecomposing $K_1, \dots, K_P$. Once we obtain the eigendecomposition, we can evaluate

$$(K+\sigma^2 I)^{-1} = Q(V+\sigma^2 I)^{-1} Q^{\mathrm{T}}, \tag{3.10}$$

$$\log |K+\sigma^2 I| = \sum_i \log (V_{ii} + \sigma^2) \tag{3.11}$$

This exact learning and inference scheme cost $O(Pm^{1+1/P})$ operations ($P > 1$) and $O(Pm^{2/P})$ storage, which is significantly less than that of inducing point-based approximation methods. However, the structure exploitation methods require the data to sit on a regular grid which severely limits its use in modelling real-world data.

3.2.3 Variational Inference and Deep Gaussian Processes

One key component of inducing-point based method is the set of inducing inputs, and selecting the most appropriate inducing inputs can pose a great challenge when dealing with large dataset. Titsias [Titsias, 2009] proposed that instead of using hand-picked, fixed inducing inputs, one can treat the set of inducing points as hyperparameters of Gaussian process models, and optimise all parameters jointly as part of the variational approximation framework. Variational methods estimate the true posterior with an approximate posterior which is optimised by minimising the Kullback-Leibler divergence between the approximate posterior and true posterior.

Deep Gaussian processes [Damianou and Lawrence, 2013] are another interesting class of Gaussian process models. They are multi-layer generalisation of GP, and correlations between different layers enable DGP to discover complicated patterns in the data. DGP relies on variational inference for posterior approximation.

We will describe the models we use in this project in the next chapter. The choices we make are connected to the discussions we have in this section. In particular, we will use:

- KISS-GP model, a generalisation to covariance matrix structured exploitation that enables exact inference for dataset not placed on a grid, and compare the performance of this exact inference approach with other approximation methods;
- the variational inference version of FITC, known as SGPR (sparse Gaussian process regression), to verify that SGPR performs better than FITC due to the optimised inducing inputs. SGPR is described in Section 4.2;
- SVGP (stochastic variational Gaussian processes), another class of approximate methods leveraging variational inference. We choose SVGP because it should perform better than SGPR on larger dataset (Section 4.3);
- DSDGP (doubly stochastic deep Gaussian processes), an improved version to [Damianou and Lawrence, 2013], discussed in Section 4.5. DGP remind us of deep neural networks that utilise simple activation to represent highly non-linear functions. We would like to examine if DGP employing simple RBF kernel are as effective as ‘shallow’ GP models equipped with sophisticated, task-specific kernel functions.

Chapter 4

Models

This chapter introduces the sparse Gaussian process models we use for pattern discovery and extrapolation in the air pollution data. We mainly make use of the following models:

- The fully independent training conditional (FITC) [Snelson and Ghahramani, 2006] model;
- The sparse Gaussian process regression (SGPR) [Titsias, 2009] model;
- The stochastic variational Gaussian process (SVGP) [Hensman et al., 2013] model;
- The structured kernel interpolation (SKI) framework [Wilson and Nickisch, 2015], and kernel interpolation for scalable structured Gaussian processes (KISS-GP) model;
- The deep kernel learning (DKL) model [Wilson et al., 2015] based on KISS-GP;
- The doubly stochastic deep Gaussian processes (DSDGP) model [Salimbeni and Deisenroth, 20

We prefer FITC over SoR as the only 'pure' inducing point-based method we use, since FITC is a more faithful approximation. The SGPR model is based on FITC but leverages the power of variational inference to optimise the set of inducing inputs.

We will introduce the stochastic variational inference (SVI) method for Gaussian processes. The SVGP model enables us to cope with huge amount of data and work with non-Gaussian likelihood functions.

We will elucidate the SKI framework based on our previous review of matrix structure exploitation methods, as well as how it can be combined with neural network feature extractor to produce a deep model that can work with high-dimensional inputs. We will also describe another promising deep model, DSDGP whose 'depth' comes from layered Gaussian processes instead of an neural network feature extractor. We are looking forward to seeing how well these two deep models fare against each other, and against those 'shallow' models, particularly with multifaceted inputs.

4.1 FITC

Introduced in Chapter 3.

4.2 SGPR

The Kullback-Leibler divergence

$$\mathrm{KL}(p(x)||q(x)) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \tag{4.1}$$

can be understood as measure of how one probability distribution $q(x)$ diverges from the *expected* probability distribution $p(x)$, and plays a central role in variational inference. In variational inference, we are almost always interested in approximating the true posterior distribution with an approximate the posterior, and optimise the approximate posterior so as to minimise the KL divergence between the true posterior and its approximate.

The key idea behind SGPR is to treat the position of inducing inputs as hyperparameters. Suppose we have target vectors $y = \{y_i\}_{i=1}^n$ which is the noisy observation of some function $\{f(x_i)\}$, with noise variance σ^2. We apply a Gaussian process prior f over the function $f(\dot{})$, introduce a set of inducing inputs $\{z_i\}_{i=1}^m$ as we do with FITC in Chapter 3, and denote the collection of $f(z_i)$ by a vector u. Then, using notations introduced in Chapter 2 and 3, we have

$$p(y|f) = N(y|f, \sigma^2 I), \tag{4.2}$$

$$p(f|u) = N(f|K_{nm}K_{mm}^{-1}u, \tilde{K}), \tag{4.3}$$

$$p(u) = N(u|0, K_{mm}) \tag{4.4}$$

where K_{mm} denotes covariance matrix among inducing inputs, K_{nm} denotes covariance matrix between training inputs and inducing inputs, and $\tilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$.

To see where KL divergence arises, we first write

$$\log p(y|u) = \log \mathbb{E}_{p(f|u)}[p(y|f)] \geq \mathbb{E}_{p(f|u)}[\log p(y|f)] \triangleq \mathcal{L} \tag{4.5}$$

where the inequality holds because of the concavity of logarithm and Jensen's inequality, and $\mathcal{L}$ is a lower bound for the conditional probability $p(y|u)$. If $p(y|f)$ factorises across input dimensions such that

$$p(y|f) = \prod_{i=1}^{n} p(y_i|f_i), \tag{4.6}$$

$\mathcal{L}$ can be shown to also factorise:

$$\mathcal{L} = \log(\prod_{i=1}^{n} N(y_i|\mu_i, \sigma^2) \exp(-\frac{1}{2\sigma^2}\tilde{k}_{i,i})) \tag{4.7}$$

where $\mu = K_{nm}K_{mm}^{-1}u$ and $\tilde{k}_{i,i} = \tilde{K}_{i,i}$. It can be shown that the difference between $\log p(y|u)$ and $\mathcal{L}$ is exactly the KL divergence

$$KL(p(f|u)||p(f|u,y)). \tag{4.8}$$

An obvious way to minimise the KL divergence is to set the inducing inputs the same as the training inputs so that the KL divergence is zero. However, this approach recovers the original, non-approximate GP model and provides no computational advantage. We can instead optimise w.r.t. the set of inducing inputs $\{z_i\}$. [Titsias, 2009] proved that the lower bound $\mathcal{L}$ monotonically increases if we employ a greedy selection strategy for the inducing inputs, and thus we need not use gradient-based optimisation to find the optimal locations of the inducing inputs.

4.3 SVGP

If we consider the true log posterior $\log p(y)$, it can be written as

$$\log p(y) = \log \int p(y|u)p(u)du \geq \log \int \exp(\mathcal{L})p(u)du \tag{4.9}$$

We introduce an variational distribution $q(u)$ to approximate the distribution of inducing values $p(u)$, such that

$$\log p(y) \geq \mathbb{E}_{q(u)}[\mathcal{L} + \log p(u) - \log q(u)] \triangleq \mathcal{L}'. \tag{4.10}$$

If we parametrise the distribution $q(u)$ as a Gaussian, $q(u) = N(u|m,S)$, we obtain the expression for the variational bound $\mathcal{L}'$

$$\mathcal{L}' = \sum_{i=1}^{n} \{\log N(y_i|k_i^{\mathrm{T}} K_{mm}^{-1} m, \sigma^2) - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}(S\Gamma_i)\} - \mathrm{KL}(q(u)||p(u)) \tag{4.11}$$

where k_i is a vector of the i-th column of K_{mn} and $\Gamma_i = K_{mm}^{-1} k_i k_i^{\mathrm{T}} K_{mm}^{-1}$. We can then apply stochastic variational inference by using a gradient-based optimiser, but in the direction of the *natural gradient* [Salimbeni et al., 2018] as instructed in [Hensman et al., 2012]. Compared with SGPR, SVGP usually obtains better result when dealing with huge dataset due to the employment of stochastic optimisation.

4.4 SKI and KISS-GP

Following our review in Chapter 2, there appears to be a straightforward way of combining inducing point-based methods with structure exploitation methods: we can simply put all inducing points on a grid, and then exploit either Toeplitz or Kronecker structure. However, this only speeds up terms related to $K_{U,U}$ (or K_{mm}^{-1} using chapter three's notation, the covariance matrix among inducing inputs), leaving the dominant term $O(nm^2)$ unchanged, which is related to $K_{X,U}$, the covariance matrix between training input X and the set of inducing points U.

Nevertheless, we can approximate $K_{X,U}$ by interpolating X to the grid. For simplicity, we assume we are on a one-dimensional grid and would like to evaluate $k(x_i, u_j)$ for training input x_i and inducing point u_j. We find the two nearest inducing points u_a, u_b surrounding x_i such that $u_a \leq x_i \leq u_b$. We can then formulate an interpolation strategy to approximate $k(x_i, u_j)$:

$$\tilde{k}(x_i, u_j) = w_i k(u_a, u_j) + (1 - w_i) k(u_b, u_j) \tag{4.12}$$

where $w_i, (1 - w_i)$ are interpolating weights representing relative distances between u_a, u_b and u_j.

We can easily generalise this method to higher-dimensional grids to exploit the Kronecker structure via forming

$$K_{X,U} \simeq W K_{U,U} \tag{4.13}$$

where W is the interpolating weight matrix of size n $\times$ m. We can either use local cubic interpolation [Keys, 1981] on equi-distanced grids, with 4 non-zero weights per row

of W, or inverse distance weighting [Shepard, 1968] on general non-equally spaced rectangular grids with 2 non-zero weights per row of W.

Substituting the expression of approximate $K_{X,U}$ into the kernel expression of SoR, we have

$$K_{X,X} \simeq K_{X,U} K_{U,U}^{-1} K_{U,X} \simeq W K_{U,U} K_{U,U}^{-1} K_{U,U} W^{\mathrm{T}} = W K_{U,U} W^{\mathrm{T}} = K_{\mathrm{SKI}}. \tag{4.14}$$

We can also add the diagonal correction term to recover the approximate form of the FITC kernel.

Having obtained K_{SKI}, we can easily solve the linear system $(K_{\mathrm{SKI}} + \sigma^2 I)x = y$ for inference using preconditioned conjugate gradient method, and leverage the ability to make fast matrix vector products with K_{SKI} to compute the log determinant $\log|K + \sigma^2 I|$ exactly.

This approach of applying sparse interpolation and Toeplitz/Kronecker algebra to GP models is called KISS-GP. It is highly scalable as it only requires $O(n + m\log m)$ computations and $O(n+m)$ storage for one-dimensional Toeplitz grid, or $O(n + Pm^{1+1/P})$ computations and $O(n + Pm^{2/P})$ storage for P-dimensional Kronecker grid.

One important observation is that established inducing point-based methods such as SoR or FITC can be understood as kernel interpolating methods too. To see this, we notice that, for the predictive mean of a noise-free, zero-mean Gaussian process

$$\bar{f}_* = w_X^{\mathrm{T}}(x_*) y = \alpha^{\mathrm{T}} K_{X,x_*} \tag{4.15}$$

where $w_X(x_*) = K_{X,X}^{-1} K_{X,x_*}$ and $\alpha = K_{X,X}^{-1} y$ can be understood as a weight factor for each observation y and training-test cross-covariance K_{X,x_*} respectively, if we perform regression on training data $\{u_i, k(u_i,x)\}_{i=1}^m$, the predictive mean becomes

$$\bar{f}_* = K_{x_*,U} K_{U,U}^{-1} K_{U,x} \tag{4.16}$$

which is exactly the SoR kernel. Thus, we can view all inducing point-based methods as a *global* interpolation on the true kernel $k(x,x')$, entries of whose weight matrix are all non-zero, in contrast to the *local* cubic or inverse distance weighting interpolation employed by KISS-GP.

It would be interesting to compare the performance of these two strategies. Local interpolation methods compromise on its ability to capture global structure, but can enjoy the benefit of far more interpolating points.

One drawback of KISS-GP is its inability to cope with high-dimensional input, a manifestation of the classic *curse of dimensionality*. If we adopt a cubic grid structure and have fixed number of grid points per dimension, the total number of grid points grows exponentially as the dimensionality increases; meanwhile, the (fixed-sized) training set becomes more sparse in higher-dimensional spaces. A rule of thumb is to only work with $D \leq 4$. Another severe restriction is its inability to perform out-of-domain learning. Interpolation requires all train and test inputs to reside within the grid, which may dent KISS-GP's performance on certain extrapolation tasks such as times series forecast.

4.4.1 Deep Kernel Learning

The DKL model [Wilson et al., 2015] was proposed to address KISS-GP's inadequacy with regard to high-dimensional inputs. The set of inputs X are first processed by a deep neural network to obtain a low-dimensional representation Z; then, a KISS-GP model is applied to Z. Concretely, suppose $z_i = w(x_i) \in Z$ is the neural network output corresponding to $x_i \in X$, and $w(x)$ represents the non-linear map induced by the neural network on x, the DKL model can be viewed as a KISS-GP model with an effective kernel $\tilde{k}(x,x') = k(w(x),w(x'))$. The model has two sets of parameters: $\mathbf{w}$, the weight matrices of the neural network, and $\boldsymbol{\theta}$, the hyperparameters of the kernel function $k(x,x')$. $\{\mathbf{w},\boldsymbol{\theta}\}$ are jointly learnt via maximisation of MLL of the combined model.

4.5 Deep Gaussian Processes and DSDGP

DKL benefits from both the flexibility of the GP non-parametric approach and the hierarchical feature extraction capability of deep neural networks. However, 'depth' need not involve neural networks; we can achieve similar goals, by setting up layers of Gaussian process priors (with the previous layer's prediction being the next layer's input) and maintaining correlation among layers. This resemblance to neural networks should enable DGP models to learn more complicated patterns from data, which is

demonstrated in [Salimbeni and Deisenroth, 2017].

Suppose we have a training set $(X,Y) = (\{x_i\}_{i=1}^N, \{y_i\}_{i=1}^N)$, L layers of Gaussian processes prior $F^1, \ldots, F^L$, and we admit automatic relevance determination for each GP. For each layer we also have a set of inducing points Z^{l-1} and corresponding values $U^l = F^l(Z^{l-1})$. The joint distribution of this deep model can be factorised as

$$p(Y, \{F^l, U^l\}_{l=1}^L) = \underbrace{\prod_{i=1}^N p(y_i|f_i^L)}_{\text{likelihood}} \underbrace{\prod_{l=1}^L p(F^l|U^l; F^{l-1}, Z^{l-1}) p(U^l; Z^{l-1})}_{\text{DGP prior}} \tag{4.17}$$

where f_i^L stands for the distribution obtained by feeding x_i through all layers, and $F^0 = X$.

Inference in this model is done via variational method. We approximate the true posterior p with an approximate posterior q, and minimise the Kullback-Leibler divergence KL$[q|p]$, which is equivalent to maximising the lower bound on the MLL:

$$\mathcal{L} = \int\int q(F,U)[\log \frac{p(Y,F,U)}{q(F,U)}]dFdU = \mathbb{E}_{q(F,U)}[\log \frac{p(Y,F,U)}{q(F,U)}] \tag{4.18}$$

here we omit the index for each layer l and use F, U to represent collections of F^l, U^l for notational simplicity. We assume that the approximate posterior has three desirable properties:

- It includes the exact model, conditioned on U;
- The approximate posterior of U, $q(U)$, factorises across layers;
- Each $q(U^l)$ is a Gaussian with mean m^l and variance S^l.

Under these assumptions, the posterior takes the factorised form

$$q(F,U) = \prod_{l=1}^L p(F^l|U^l; F^{l-1}, Z^{l-1}) q(U^l). \tag{4.19}$$

Since both $p(F^l|U^l)$ and $q(U^l)$ are Gaussian, we can marginalise over U to get

$$q(F) = \prod_{l=1}^L q(F^l|m^l, S^l; F^{l-1}, Z^{l-1}) = \prod_{l=1}^L N(F^l|\tilde{\mu}^l, \tilde{\Sigma}^l) \tag{4.20}$$

where

$$[\tilde{\mu}^l]_i = \mu_{m^l, Z^{l-1}}(f_i^l) = m(f_i^l) + \alpha(f_i^l)^{\mathrm{T}}(m^l - m(Z^{l-1})), \tag{4.21}$$

$$[\tilde{\Sigma}^l]_{ij} = \Sigma_{S^l, Z^{l-1}}(f_i^l, f_j^l) = k(f_i^l, f_j^l) - \alpha(f_i^l)^{\mathrm{T}}(k(Z^{l-1}, Z^{l-1} - S^l))\alpha(f_j^l). \tag{4.22}$$

Here, $m(x)$ is the mean function of each layer, $k(x,x')$ is the combined kernel of the true kernel and the noise term $\sigma^2\delta_{ij}$, and f_i^l represents the i-th row of F^l. Note that the i-th marginal of the final layer L

$$q(f_i^L) = \int \prod_{l=1}^{L} q(f_i^l | m^l, S^l; f_i^{l-1}, Z^{l-1}) df_i^l \tag{4.23}$$

depends **only** on the i-th marginal of all other layers. This nice property enables us to effortlessly sample $q(f_i^L)$ using the reparametrisation trick.

It can be shown that the lower bound in Equation (4.7) can be rearranged into

$$L = \sum_{i=1}^{N} \mathbb{E}_{q(f_i^L)}[\log p(y_n | f_n^L)] - \sum_l \mathrm{KL}(q(U^l) || p(U^l; Z^{l-1})) \tag{4.24}$$

where $\mathrm{KL}(q||p)$ is the Kullback-Leibler divergence between q and p. The lower bound can be efficiently computed by considering two levels of sampling: firstly, we can sub-sample the data; secondly, we can approximate the expectation with a Monte Carlo sample from Equation (4.12). This is also the origin of the name 'doubly stochastic'.

Prediction at test location only requires us to evaluate $q(f_*^L)$, which can be obtained using the Gaussian mixture

$$q(f_*^L) \approx \frac{1}{S} \sum_{s=1}^{S} q(f_*^L | m^L, S^L; f_*^{(s)^{L-1}}, Z^{L-1}) \tag{4.25}$$

where S denotes the number of samples $f_*^{(s)^{L-1}}$ we draw using Equation (4.12).

Chapter 5

Setup

This chapter details the setup of all experiments we run, and explains some choices we make with regard to dataset preprocessing and model selections. Before proceeding to the particulars, we reiterate the goal of our experiments that we stated in Chapter 3.

Our first set of experiments is about modelling the entire air pollution dataset, as a validation step to all proceeding tasks. Our second set of experiments assesses if Gaussian processes can be used to perform spatial extrapolation, i.e. if Gaussian process models can generalise to locations that are not covered by the training set. The third set of experiments compare the performance of our spatiotemporal modelling approach with that of a collection of individual, local temporal models, on a collective time series forecast task. Finally, we use our spatiotemporal model to predict incidence of exceedance in London.

All sparse GP models (SGPR, SVGP, FITC) are implemented using the `GPflow` library [Matthews et al., 2017], a Gaussian processes library based on `Tensorflow`. KISS-GP models exploiting Kronecker structure are implemented in the `GPytorch` library [Gardner et al., 2018][Pleiss et al., 2018], which is based on `Pytorch`. These libraries leverage the underlying base libraries for fast linear algebra operations such as Cholesky decomposition that GP rely heavily on. They also provide a simple, user-friendly interface when working in a Jupyter Notebook environment. We also make use of `scikit-learn` [Pedregosa et al., 2011] for its various training-test split methods, and `scipy` for the implemented k-means clustering algorithm.

All experiments are run on Google Cloud Platform, in a Virtual Machine instance

Table 5.1: Statistics of the NO_2 Dataset

mean	standard deviation	max	min	25% percentile	median	75% percentile
24.832	17.972	238.496	0.169	11.463	20.968	33.905

with 16 vCPUs and 32 GB memory.

5.1 Dataset

The experiments mainly utilise hourly nitrogen dioxide (NO_2) concentration data collected by AURN among static monitoring sites across Great Britain from 01/01/2017 to 30/06/2018. There are also AURN monitoring sites located in Northern Ireland and West Ireland, but they are excluded due to the ocean separation. There are 144 eligible sites in total (note that not all sites measure NO_2, and some do not have sufficient continuous measurement). Before any preprocessing, the dataset contains 1886832 entries, with 2860 negative values, or 1.52%, and 119451 missing entries, or 6.33%.

The original dataset has an hourly temporal resolution, and is aggregated to obtain daily mean values. Negative measurements and null entries are also filtered out. We believe predicting daily average is a more sensible goal than hourly value, since the latter has too much variability and can be affected considerably by random events we have no knowledge of. For high-emission zones, daily mean value smoothes out the intra-day variability, enabling our models to discover mid-to-long term trend. Due to the nature of spatiotemporal modelling, our first three sets of experiments also require data at as many training locations as possible, and using mean values as a substitute shrinks the dataset tremendously, enabling us to train on the entire dataset which is otherwise impossible due to constrained computational resources. For the last task, which is about forecasting hourly exceedances, we use the original hourly data, but only at 12 London sites.

The dataset contains 74,789 data points after the aforementioned preliminary processing. Statistics of the dataset can be found in Table 5.1. Distribution of the dataset can be seen from the histogram 5.1. There are only two incidences of the concentration exceeding 160 μg m^{-3} and are thus excluded from the range of the histogram.

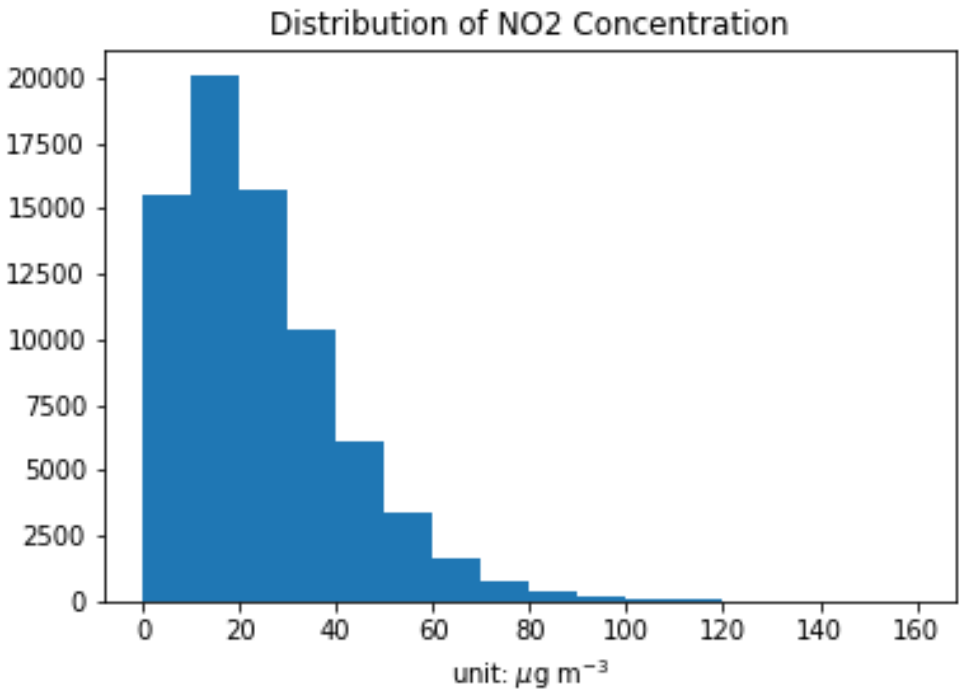


Figure 5.1: Distribution of daily mean NO_2 concentration across all eligible AURN sites

5.1.1 Feature Selection

As is mentioned in Chapter 3, our choice of features is influenced by previous works that employ hierarchical Bayesian methods to model air pollution data. There are several features that may contribute to the deposition and concentration of nitrogen dioxide in the air:

- Spatial coordinates (latitude, longitude, altitude);
- Time;
- Factors affecting NO_2 emission: site type (urban or rural, adjacency to industrial zones/roads), nearby traffic density, weekday or weekend, and other factors relating to human activities;
- Factors affecting NO_2 dispersion: wind speed and direction, pressure, terrain;
- Other factors catalysing the formation of NO_2, such as ozone concentration, temperature and rainfall, etc.

Spatial coordinates affect the mid-to-long range transport of NO_2 particles, whereas emission-related factors like site type influences the short range ambient concentration

weekday	weekend	urban	suburban	rural	traffic	industrial	background
1	0	[1,0,0]	[0,1,0]	[0,0,1]	[1,0,0]	[0,1,0]	[0,0,1]

Table 5.2: Categorical features one-hot encoding

level. Instead of using latitude and longitude, we use northing and easting to pinpoint the coordinates of each monitoring site, since we are interested in the absolute distance between two sites, whereas the difference in latitude/longitude does not reflect the real geodesic distance between two sites. We decide not to use altitude as a feature, since Great Britain is relatively flat.

Most of the experiments make use of the following features: time, northing and easting coordinates, weekday/weekend indicator and site environment types. In some exploratory runs, we only utilise the spatiotemporal features, but solely for the purpose of demonstrating the improvement we have by incorporating all available features. We also use temperature in the last hourly exceedances forecast task.

Categorical features are encoded by binary or one-hot encoding so that they can be used in parallel with numerical features. Their correspondence can be found in Table 5.2. A typical model input for experiments with all features is a 10-dimensional vector.

5.1.2 Normalisation

It is important to ensure that attributes have the same scale when applying machine learning algorithms, especially for kernel methods, unless automatic relevance determination (described in Chapter 2) is used. In kernel methods, the covariance between two points x_1 and x_2 is a function of x_1 - x_2. If different components have varying scales, the bigger component will start to dominate the covariance term. There are two common scaling methods: min-max scaling and standardisation.

Min-max scaling applies the following transformation to the dataset and scales it to [-1, 1].

$$x' = 2 * \frac{x - x_{min}}{x - x_{max}} - 1 \tag{5.1}$$

Standardisation scales the data via

$$x' = \frac{x - \mu}{\sigma} \tag{5.2}$$

where μ is the mean, and σ is the standard deviation of the data, such that each attribute of the transformed dataset has zero mean and unit variance. We use min-max scaling for KISS-GP models because of the requirement that all training and test data must be within the grid. We adopt standardisation method for all other models.

5.2 Hyperparameters, Optimisers and Initialisation

For KISS-GP models, we work with two different grid sizes: 10^3 and 25^3, and we expect to see that KISS-GP-25 performs better, thanks to the expanded grid size leading to more accurate interpolation. Our DKL model includes a neural network feature extractor of dimension $10 \times 1000 \times 1000 \times 500 \times 50 \times 2$ and ReLU activation functions as suggested by Wilson.

We use k-means clustering algorithm (implemented in `scipy.cluster.vq.kmeans2`) to select 100 inducing inputs for each experiment with sparse models. We should see that SGPR performs much better than FITC because SGPR also optimises the set of inducing inputs.

We use Adam optimiser for training KISS-GP (learning rate: 0.1) and DGP (learning rate: 0.01) models. We train for 100 iterations for KISS-GP models, and 1000 iterations for the rest. SGPR and FITC are optimised by the method implemented in `scipy.optimize.minimize`. For DGP models, we also make use of `natural gradients` [Salimbeni et al., 2018] for the outer layer to accelerate convergence.

5.3 Experiments

Experiments conducted in this project can be divided into four groups:

- Modelling experiments;
- Spatial extrapolation experiments;
- Time series forecast experiments;

- Hourly exceedance forecast experiments.

Section 5.3.1 describes the general evaluation metric we use that applies to all our experiments. Details about each set of experiments and their specific evaluation methods are reported in Section 5.3.2 - 5.3.5.

5.3.1 Evaluation Metric

A general assessment metric we use is the root mean square error or RMSE. It is defined by

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - f_i)^2}{N}} \tag{5.3}$$

where f_i, y_i denote the point estimate given by the model at the i-th input and the real target value, and N denotes the size of the test set. We could also use the mean absolute error

$$\text{MAE} = \frac{\sum_{i=1}^{N}|y_i - f_i|}{N} \tag{5.4}$$

but RMSE is more sensitive to outliers.

If distributions of different test sets are dissimilar, we cannot compare models' performance on those test sets by directly comparing the RMSE. For example, if one test set has a much larger variance, then predictions given by the model are also likely to have larger RMSE than predictions on test sets with smaller variance. This problem is especially troublesome for our second, third and fourth experiments that do not use random split for training-test split. Thus, it is important to look at relative metrics transferable between different test sets. We make use of the standardised root mean square error, or SRMSE, defined by

$$\text{SRMSE} = \frac{\text{RMSE}}{\text{RMSE}_{emp}}$$

to gauge the model's performance on different datasets. *emp* stands for errors obtained by assuming the prediction to be an empirical distribution of the test set, i.e. the prediction is equal to test set mean everywhere. Desired models should report small SRMSE < 1.

We make use of the metrics above to assess the quality of different models. Evaluation procedure could vary due to the multi-fold goals we have, which are detailed in subsequent sections.

5.3.2 Modelling NO_2 Concentration

In this set of experiments, we investigate how well GP models can represent the overall distribution of our NO_2 data. This task falls into the category of in-domain learning, and we expect GP to do well even with basic kernels.

We split the dataset described in Section 5.1 into an 80% training set and a 20% test set, using scikit-learn's `ShuffleSplit` method to obtain five different training-test set pairs. The `ShuffleSplit` method applies a random permutation to the dataset before splitting it, and we make five splits to discount the possibility that certain models may perform well on one training-test pair by chance. **We refer to this multiple-splitting scheme as 'validation runs' throughout the project, and use 'exploratory runs' to describe experiments done only on one training-test pair whose aim is to get a glimpse of certain properties.**

We carry out the modelling task with nine models: KISS-GP-10, KISS-GP-25, DKL, FITC, SGPR, SVGP, DGP1, DGP3, DGP5, and utilise RBF kernel only. The aim of this set of experiments is not to obtain highly accurate predictions, but to validate our premise that GP models work and can extract patterns hidden in air pollution data. We report the RMSE and SRMSE of the sampling distribution, in the form

$$\text{(S)RMSE} = m(\text{(S)RMSE}) \pm \frac{\sigma(\text{(S)RMSE})}{\sqrt{n}}. \tag{5.5}$$

Here, m is the sample mean, σ refers to the standard deviation of the samples, and $n = 5$. The same procedure is done for all other validation runs in this project.

We expect all models to work better than the empirical distribution (corresponding to SRMSE < 1) despite the lack of more sophisticated kernel design. We expect to observe that KISS-GP-25 functions better than KISS-GP-10, due to the expanded grid size. We also anticipate that DGP models, especially DGP5, to fare well against all other models thanks to their enhanced capability of modelling more complicated correlations (4.5).

5.3.3 Spatial Extrapolation

In this set of experiments, we investigate the possibility of applying GP models to spatial extrapolation, i.e. if trained GP models can predict NO_2 concentration from

2017-01-01 to 2018-06-30 at a location disjoint from the training sites, when all we know about the said location is its geographic coordinates (and possibly site environment type). As we discussed in Chapter 3, this goal is possible on a smaller scale, and we would like to see if our GP models can perform global spatial extrapolation. Even if we fail, we can still analyse the result and hopefully find out causes behind the failure.

Spatial extrapolation is different from the modelling task in that we have to split the set of AURN sites, which is relatively small in size. To ensure that test sets are representative of the overall distribution, we employ the technique of *stratified sampling*.

The collection of monitoring sites can be divided into subpopulations based on certain criteria, e.g. the government region the site is located in, site environment type, etc. Stratified sampling refers to the method of sampling each subpopulation within an overall population independently. We sample based on the 'Government Region' attribute of each site for this task, and sample based on the 'Environment Type' attribute for spatial extrapolation in London (see Section 5.3.3.1).

We use scikit-learn's `StratifiedShuffleSplit` method and make a roughly 80%/20% training-test split. Figure 5.2 shows the fraction of each government region in one sample test set and among all 144 monitoring sites. We can see that the distribution of test site regions are mostly consistent with that of all sites. Also, the test sample does not contain regions such as Highland or North Wales, for there are too few of them among all monitoring sites.

We evaluate model performance based on RMSE and SRMSE of the stratified sampling distribution. We expect our models, especially those trained with additional features, to at least capture some periodicity or government-region/site-environment-type specific patterns. Making educated guess about model performance is hard, since we are essentially performing out-of-domain learning, and DGP models are not guaranteed to work better in this regard.

5.3.3.1 Spatial Extrapolation in London

Our spatial extrapolation experiments are applications of GP at a global scale. One flaw with that approach is some sites can be distant from others, leading to weak cor-

Figure 5.2: Histogram of government region attributes for one test site sample and all monitoring sites. Top figure: all monitoring sites statistic; bottom figure: test site sample

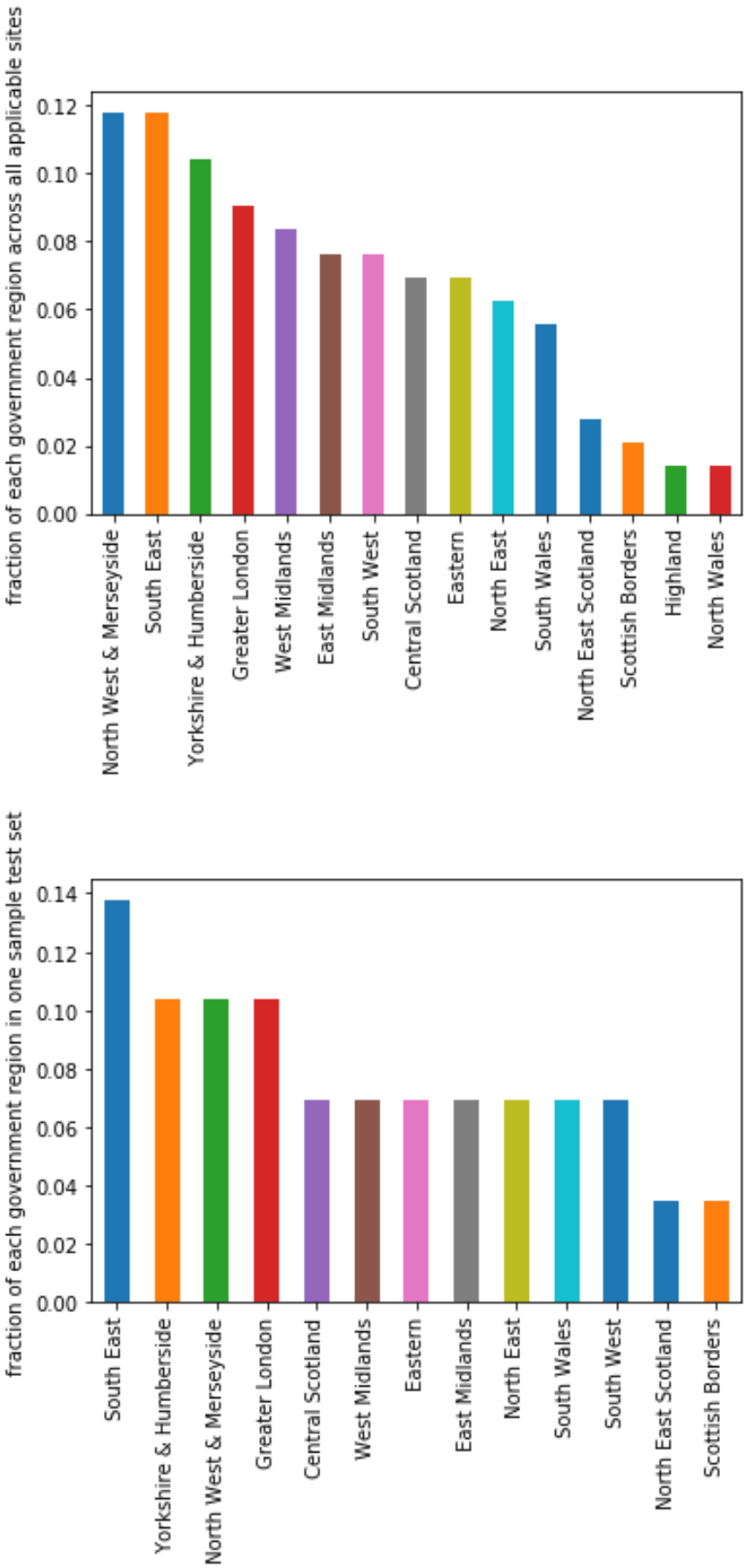
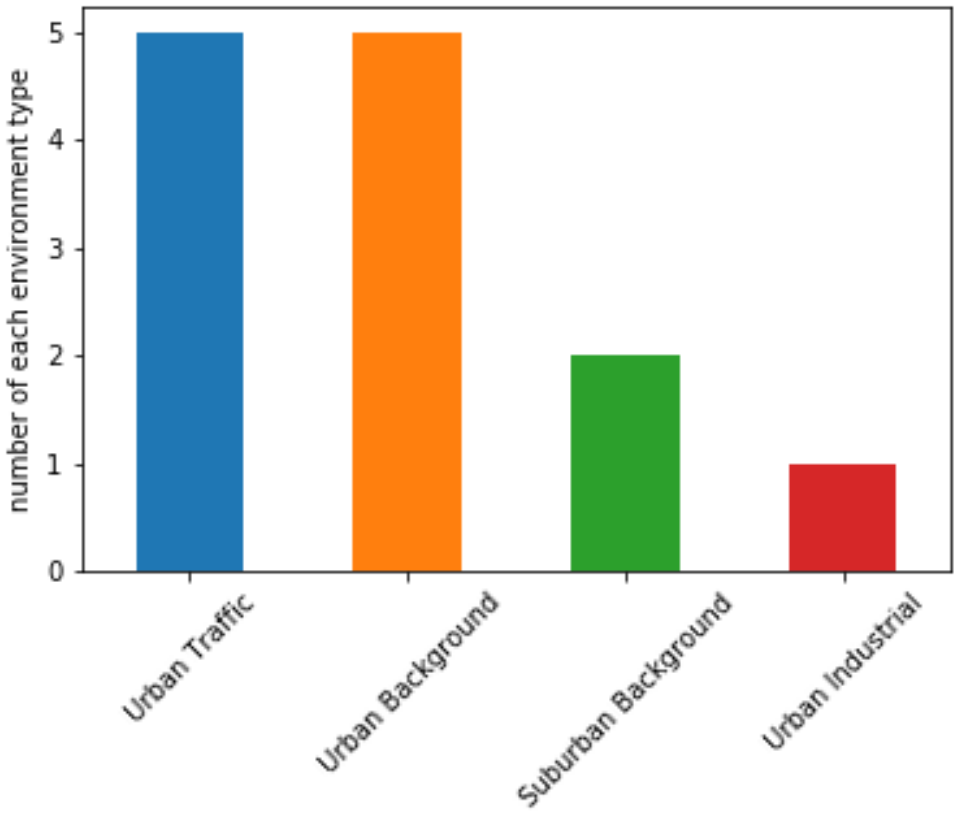


Figure 5.3: Histogram of site environment types in London



relations among them. We can instead focus on sites situated close to each other, for which London monitoring sites are a perfect example.

There are 16 AURN sites in London, and 13 of them record hourly NO_2 concentration. Figure 5.3 shows London sites environment type statistic. We have to remove the only 'Urban Industrial' site, London Harlington, to make environment-type-based stratified sampling possible. A map of the 12 eligible London sites are available in Appendix C. After the removal of London Harlington, the environment type features become de facto binary features, and we reduce the dimension of each input to 6. Also, the lack of sites of 'Suburban Background' type is a cause for concern, because GP models may not learn very well the patterns of this site type based on only one training example.

We again use scikit-learn's `StratifiedShuffleSplit` for sampling, making five 80%/20% splits. We report RMSE and SRMSE of the stratified sampling distribution.

Sites adjacency should enable the GP models to discover stronger correlations in spatial coordinates, but this advantage could be offset by the relative shortage of training data. On balance, we expect GP models to perform roughly in line with their performance in spatial extrapolation experiments.

5.3.4 Time Series Forecast

This set of experiments explores a joint approach to NO_2 concentration time series forecast. Concretely, we train on all monitoring sites but hold out a portion of data for each site, selected based on time. There is no special sampling technique involved; for each site, we slice the data at a certain 'cut-off' point in time, and add all data before that instant to the training set and all data afterwards to the test set.

Time as a feature is different from geographic coordinates, as the time dimension embodies more patterns - intra-week fluctuations, seasonal patterns, etc. Therefore, we need to consider more expressive kernels other than the RBF kernel used in all previous experiments. We conduct an exploratory run to identify the most appropriate model/kernel combination, the result of which can be found in Appendix II. Here, we focus primarily on the evaluation strategy for our validation runs.

Evaluation for time series forecast requires methodology different from evaluation strategies for other kinds of dataset. We cannot randomly and repeatedly split time series into chunks, train on some of them and test on the others; the direction of time implies causality, and we must always train on the past and predict the future. Hence, we adopt the evaluation method based on a fixed-sized 'rolling-window'.

We split the whole dataset five times. At each split, instead of dealing with the whole dataset, we only divide a subset of the whole dataset into two. The whole dataset conveniently consists of exactly 78 weeks; we use a 73-week-long rolling-window so that we have five whole weeks left for testing, and we make predictions for the subsequent week. In other words,

Split	Training	Test
First	01/01/17 - 26/05/18	27/05/18 - 02/06/18
Second	08/01/17 - 02/06/18	03/06/18 - 09/06/18
Third	15/01/17 - 09/06/18	10/06/18 - 16/06/18
Fourth	22/01/17 - 16/06/18	17/06/18 - 23/06/18
Fifth	29/01/17 - 23/06/18	24/06/18 - 30/06/18

This splitting strategy ensures that

- 1) We always train models on the past data and forecast the future;
- 2) Sizes of training sets are approximately the same, so averaging over the RMSE of five sets of predictions is reasonable.

Seven sites are excluded due to lack of eligible test data, bringing the total number of sites in these experiments to 137.

We call our time series forecast approach a 'joint' one at the start of this section, because our models encompass all monitoring sites and train on the whole dataset collectively. The opposite, 'disjoint' approach would be to model each site individually and independently. It is reasonable to expect that the 'joint' approach would work better than a 'disjoint' one, which we use as a baseline model. To verify our view, we run GP regression models five times for each site in accordance with the aforementioned dataset splitting strategy, and evaluate the RMSE and SRMSE for the sampling distribution collectively, that is, we aggregate all predictions for each test batch together and then compute the batch RMSE, instead of calculating RMSE for each individual site. Each individual model uses time and weekday/weekend indicator as inputs, and utilise ARD RBF kernels to facilitate local pattern discovery. Essentially, we are comparing two 'joint' models: one that assumes correlation between sites, and one made out of individual models only considered jointly at the evaluation step.

5.3.5 Hourly Exceedance Forecast

We have been using daily mean value as a surrogate for air quality at static monitoring sites. Some may argue that this approach is not ideal, as intra-day variabilities can be important in certain situations. In high-emission zones, such as spots in agglomerate areas or near busy roads, pollutant concentration could be highly correlated to human activities. Admittedly, daily mean value encodes little such information.

We are also interested in the task of forecasting hourly exceedances, that is, predicting incidence of hourly pollutant concentration exceeding a certain limit value. London is perhaps one of the most polluted cities in the U.K. [DEFRA, 2017], and we would like to predict such exceedence events in London. Restricting the area of interest to London alone enables us to use data at hourly resolution, for there are only 12 eligible sites and around 13000 data points for each site. Both U.K. laws and EU regulations dictate that exceedances of hourly NO_2 concentration should not occur more than 18 times a year; however, the legal limit value is 200 μgm^{-3}, exceedances of which rarely happen even in London. To make sure we have enough relevant data, we set the limit value to 100 μgm^{-3}.

As is done in spatial extrapolation experiments, we remove London Harlington. Remaining data is split for five times as we have done in time series forecast experiments. In addition to existing features, we add a new feature: hourly temperature measured at each site.

Since we are primarily interested in forecasting the incidence, not the precise value, this task is a de facto binary classification problem, and thus we use measures such as precision, recall and F_1 score to gauge model's performance, in addition to RMSE and SRMSE. These metrics are defined as

$$\text{precision} = \frac{\text{number of correct prediction}}{\text{number of predicted exceedance}}, \tag{5.6}$$

$$\text{recall} = \frac{\text{number of correct prediction}}{\text{total number of exceedance}}, \tag{5.7}$$

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{5.8}$$

We can understand precision as a measure of exactness of our model, recall as a measure of completeness of our model, and F_1 score as the (harmonic) average between precision and recall. High precision indicates that our model makes mostly correct predictions, while high recall means our model predicts most of the real exceedances. Ideally, the model should have a high F_1 score, i.e., good in terms of both precision and recall, but if we have to choose, models with high recall is preferable, as it is better to be safe than sorry.

5.4 Overfitting

In theory Bayesian methods like Gaussian processes should not 'overfit', as Bayesian predictions are always presented with a variance estimate. Also, marginal log likelihood of Gaussian processes naturally incorporates a complexity penalty term as is described in Chapter 2. There is no need to add regularisation term, or use cross-validation.

However, since the optimisation of Gaussian processes involves optimising hyperparameters using gradient-based optimisers, there is the possibility that some noise-fitting may occur. In our exploratory runs, we find that KISS-GP models are most prone to this problem, and we have to restrict training to no more than 100 iterations. We compare the prediction accuracy of sparse GP models at iteration 100 and 1000, and find no systemic deterioration in accuracy.

Chapter 6

Results and Discussion

Before proceeding to presenting our results, we reiterate the acronyms we use for our models:

- KISS-GP: kernel interpolation for scalable structured Gaussian processes;
- SGPR: sparse Gaussian process regression;
- SVGP: stochastic variational Gaussian processes;
- FITC: fully independent training conditional model;
- DGP: (doubly stochastic) deep Gaussian processes.

Details about these models can be found in Chapter 2. Our time series models also have prefixes attached to these acronyms, the meaning of which can be found in Appendix B.

We also reiterate the meaning of 'exploratory run' and 'validation run' that we first used in Chapter 5:

- **Exploratory runs** refer to experiments done only on one training-test pair whose aim is to get a glimpse of certain properties. Exploratory runs can be done with spatiotemporal features only, or with all features, and we will always specify which one is the case;
- **Validation runs** refer to experiments done on multiple (5 for all experiments) training-test pairs that are formed using predefined sampling techniques, whose aim is to produce results that reflect statistic of the sampling distribution. Validation runs are always done with all features.

Model	RMSE	SRMSE
KISS-GP-10	15.202±0.067	0.845±0.002
KISS-GP-25	14.751±0.245	0.821±0.014
SGPR	14.484±0.063	0.806±0.001
SVGP	15.045±0.053	0.837±0.002
FITC	15.265±0.088	0.849±0.002
DGP1	14.937±0.082	0.831±0.001
DGP3	**13.575 ± 0.074**	**0.755 ± 0.001**
DGP5	13.700±0.106	0.762±0.004

Table 6.1: Model performance on modelling task with spatiotemporal features only

Model	RMSE	SRMSE
SGPR	13.433±0.056	0.747±0.001
SVGP	13.720±0.057	0.763±0.003
FITC	13.598±0.0736	0.756±0.001
DGP1	13.776±0.057	0.766±0.002
DGP3	12.541±0.048	0.698±0.002
DGP5	**12.315 ± 0.079**	**0.685 ± 0.002**

Table 6.2: Model performance on modelling task with all features

6.1 Modelling Experiments

Results of the modelling task can be found in Table 6.1 and Table 6.2, for experiments with only spatiotemporal features and with all features respectively. KISS-GP methods are on par with sparse approximation methods. Deep Gaussian processes perform better than the rest, with or without additional features, demonstrating their ability to recognise existing **in-domain** patterns.

The DKL model appeared to be unstable and did not produce any meaningful result, reporting an average sampling RMSE of 194.40 ± 20.17 which is on average 10.8 times the standard deviation of the test set empirical distribution.

It is hard to determine the exact cause for this problem. One plausible guess is that combined neural networks and Gaussian process become overfitted to the training data

Model	Spatiotemporal features		All features	
	RMSE	SRMSE	RMSE	SRMSE
KISS-GP-25	26.703	1.242	N/A	N/A
SGPR 100	21.326	0.992	18.527	0.862
SVGP 100	20.361	0.947	18.594	0.865
FITC 100	21.189	0.986	19.674	0.916
DGP1 100	21.260	0.989	18.812	0.875
DGP3 100	21.578	1.004	18.844	0.877
DGP5 100	21.057	0.980	18.491	0.860

Table 6.3: Model performance on spatial extrapolation, exploratory runs

Model	RMSE	SRMSE
SGPR	$\mathbf{15.263 \pm 0.965}$	$\mathbf{0.867 \pm 0.027}$
SVGP	15.430±1.012	0.876±0.031
FITC	17.318±1.442	0.989±0.081
DGP1	15.634±1.048	0.888±0.034
DGP3	16.060±0.806	0.914±0.012
DGP5	16.058±0.772	0.914±0.016

Table 6.4: Model performance on spatial extrapolation, validation runs

due to the large number of weight matrix entries in the neural network feature extractor. A more likely explanation is that we encountered some numerical problem in optimisation algorithms, as we have seen similar behaviour exhibited by KISS-GP models when trained with standardised target values. Unfortunately, this means we cannot apply the KISS-GP methods to the remaining tasks of this project.

6.2 Spatial Extrapolation Experiments

6.2.1 Global Extrapolation Experiments

Table 6.3 shows results of our exploratory run on the task of spatial extrapolation, and Table 6.4 contains results for the validation run. The former confirms that spatiotemporal features alone do not work for spatial extrapolation. Models with spatiotemporal features only exhibit SRMSE close to 1, and thus are no better than empirical distri-

butions, suggesting that the coordinates of each site may not play a defining role in determining the value of NO_2 concentration, at least not on a scale as large as Great Britain. Our validation runs produce moderately good result, the best of which obtains an SRMSE of 0.867 ± 0.027.

To get a better idea about the result, we plot the predictions given by SGPR and DGP5 at two sites: Glasgow High Street and Bath Roadside, shown in Figure 6.1 and Figure 6.2. We choose these two sites to show the importance of site environment type. For the exploratory run, predictions at Glasgow High Street are reasonable, but predictions at Bath Roadside completely mismatch the real values. One may wonder why such a dramatic difference exists.

The answer is proximity to other similar sites. Environment type of Glasgow High Street is 'Urban Traffic'. There are four other monitoring sites in Glasgow (all of them are in training set), all of them are of urban type, and three out of four are also 'Urban Traffic' sites. These four sites serve as 'anchor' for the prediction at Glasgow High Street, and although we use spatiotemporal features only in exploratory runs, the environment-type-specific patterns are automatically inferred due to Glasgow High Street site's proximity to sites of the same type. On the other hand, Bath Roadside is the only monitoring site in Bath, and thus cannot learn from nearby sites of the same type as Glasgow High Street site does.

Another interesting observation is the periodic oscillations that only exist in validation runs. Careful counting shows that the period is roughly seven days, with a five-day 'plateauing' period when predictions are higher, and a two-day 'spike-down' to lower values. This phenomenon is due to the weekday/weekend indicator feature we use. Not only do Gaussian processes recognise long-term seasonal patterns, they are also capable of identifying the fact that weekdays are usually more polluted than weekends.

Our result shows that GP may not be good at global-scale spatial extrapolation if our criterion is precision of predictions. The best model reported by [Pirani et al., 2014] boasts a result equivalent to an SRMSE of 0.592, whereas ours is only 0.867, though their work is on a different kind of pollutant, PM_{10}. Still, Gaussian processes could make sense if we only care about long-term patterns. After all, almost all values are bound within $[m-2\sigma, m+2\sigma]$ range for SGPR predictions in validation runs (m : pre-

Model	Spatiotemporal features		All features	
	RMSE	SRMSE	RMSE	SRMSE
SGPR 100	35.755	2.151	18.260	1.099
SVGP 100	78.619	4.730	17.696	1.065
FITC 100	42.153	2.536	17.295	1.041
DGP1 100	41.885	2.520	17.770	1.069
DGP3 100	34.369	2.068	16.493	0.992
DGP5 100	37.897	2.280	16.756	1.008

Table 6.5: Model performance on spatial extrapolation in London, exploratory runs

Model	RMSE	SRMSE
SGPR	23.456±1.673	1.106±0.074
SVGP	23.721±1.924	1.117±0.081
FITC	22.301±1.780	1.045±0.058
DGP1	23.524±1.899	1.109±0.083
DGP3	22.160±2.265	1.032±0.080
DGP5	$\mathbf{22.091 \pm 2.155}$	$\mathbf{1.029 \pm 0.072}$

Table 6.6: Model performance on spatial extrapolation in London, validation runs

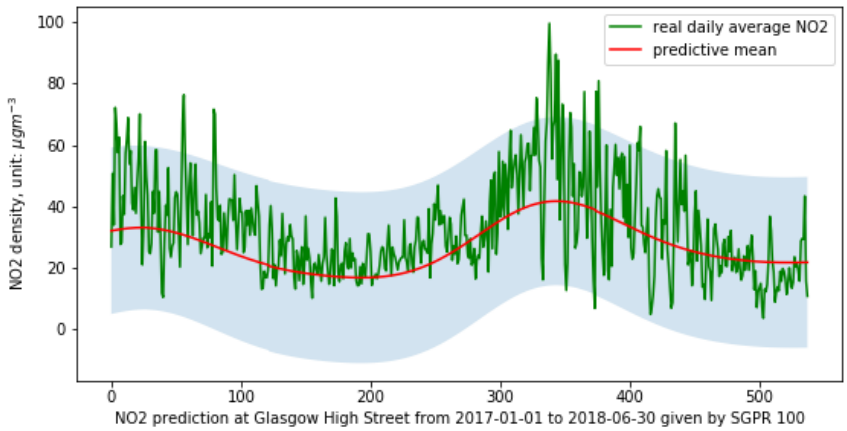
dictive mean, σ : predictive standard deviation). However, to ensure the quality of predictions, it is desirable that we: 1) Train with all available features and as much data as possible; 2) select training set such that test sites share the environment type of training sites, and preferably are in close proximity to training sites.

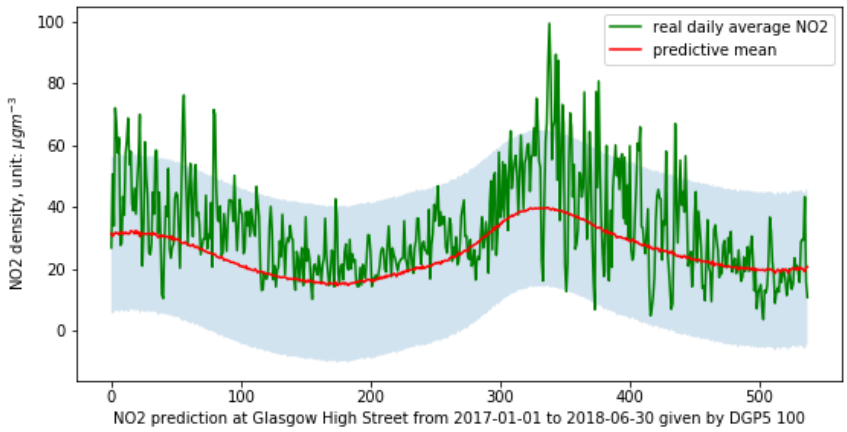
With these in mind, we confine our area of interest to a city where there are 16 monitoring sites sitting close to each other - London.

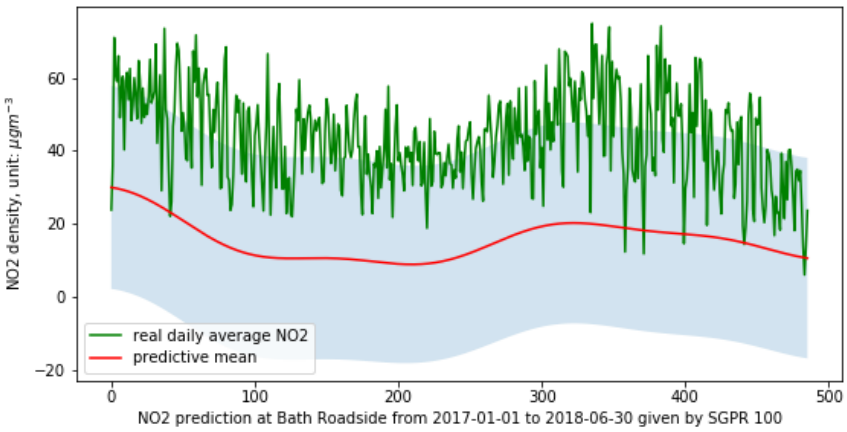
6.2.2 London Experiments

Table 6.5 and Table 6.6 show model performance on spatial extrapolation in London for exploratory runs and validation runs respectively. Models with spatiotemporal features are much worse than empirical distribution, as are evident in Figure 6.3. We can see from both Table 6.6 and Figure 6.4 that predictions are mediocre, even with all features, most notably for site London Bexley.

Figure 6.1: Predictions at Glasgow High Street and Bath Roadside sites, given by SGPR and DGP5, exploratory runs with spatiotemporal features only. X ticks start from 01/01/2017 and end at 30/06/2018. Each unit in x represents a day.







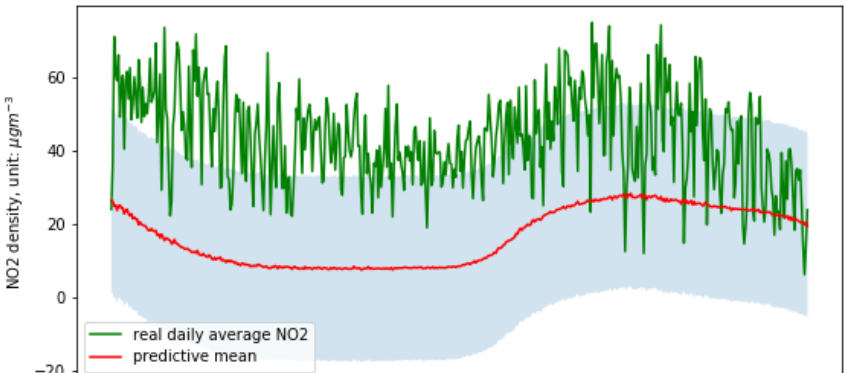
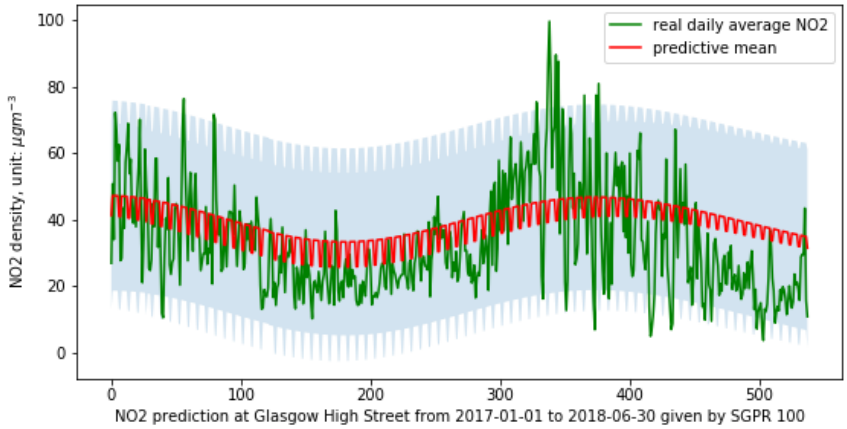
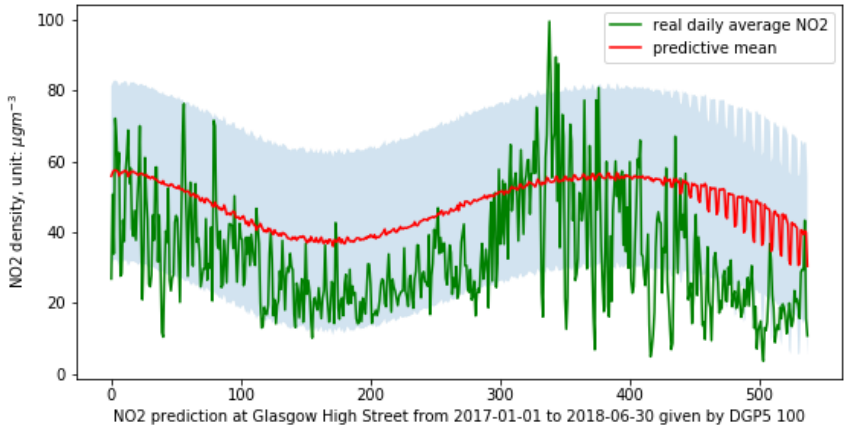
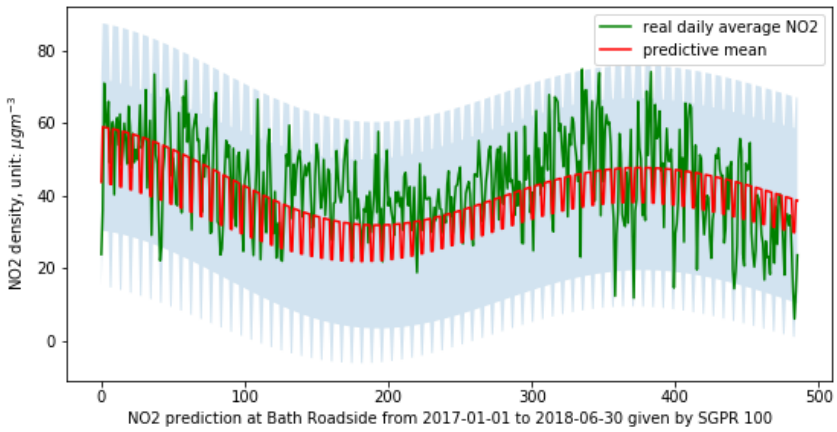


Figure 6.2: Predictions at Glasgow High Street and Bath Roadside sites, given by SGPR and DGP5, validation runs with all features.X ticks start from 01/01/2017 and end at 30/06/2018. Each unit in x represents a day.







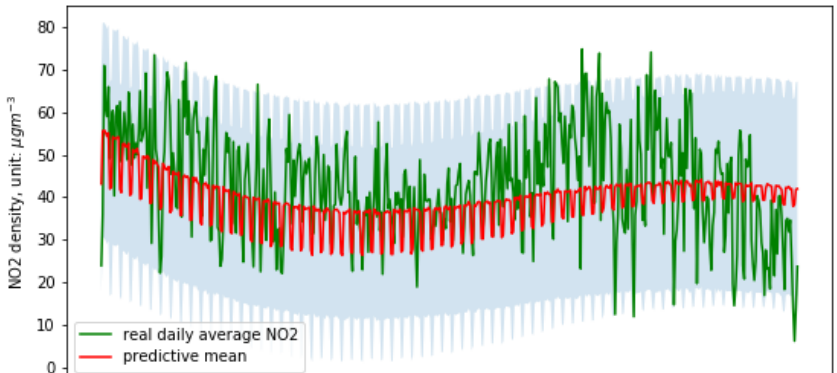
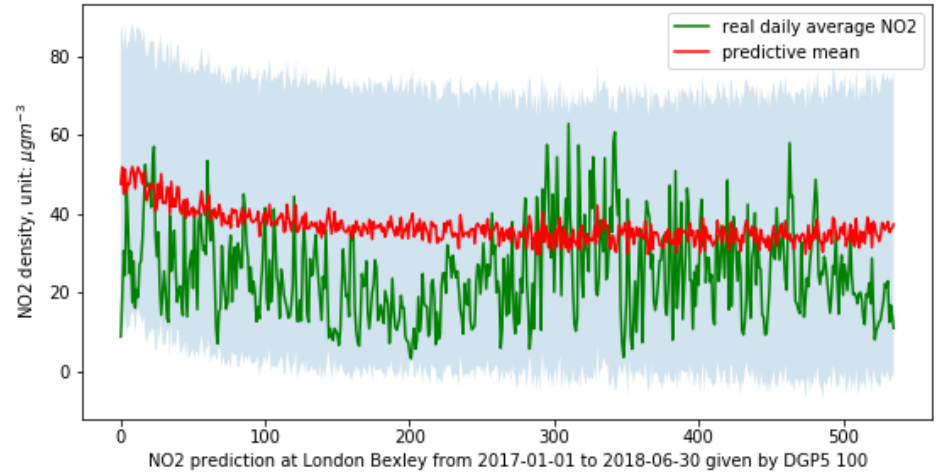
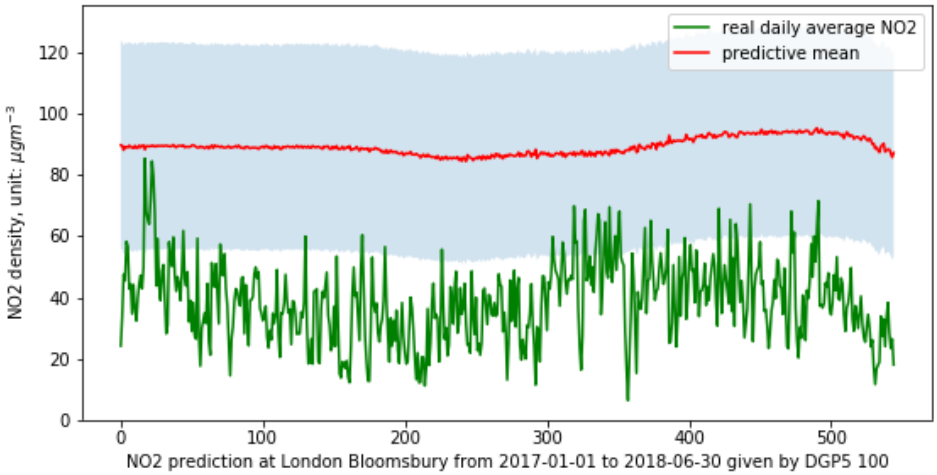


Figure 6.3: Predictions at London Bexley, London Bloomsbury and Tower Halmets Roadside sites, given by DGP5, exploratory runs with spatiotemporal features only. X ticks start from 01/01/2017 and end at 30/06/2018. Each unit in x represents a day.





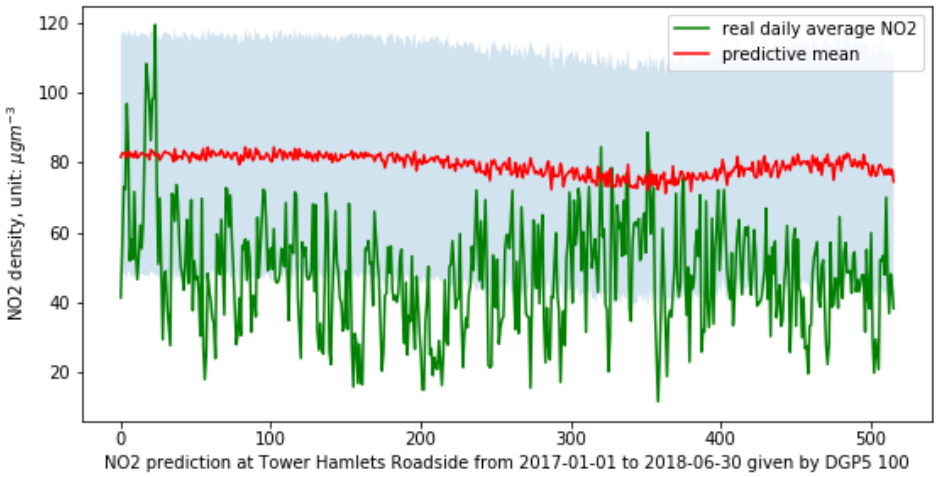
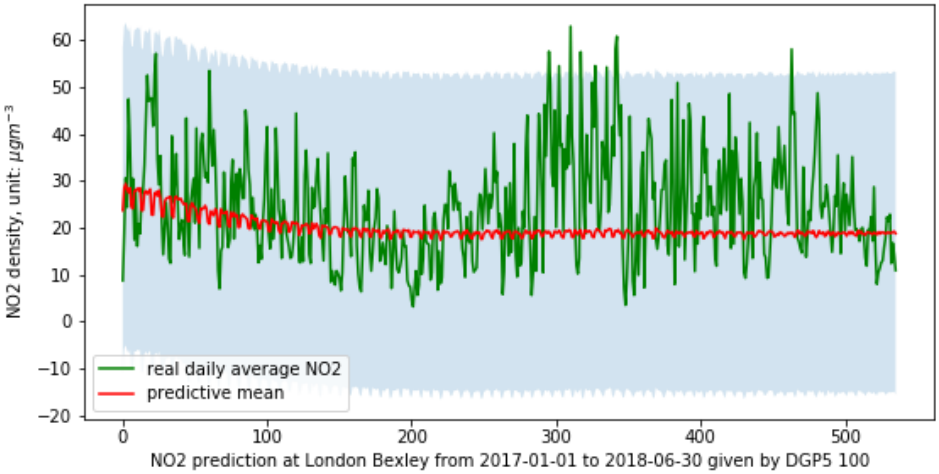
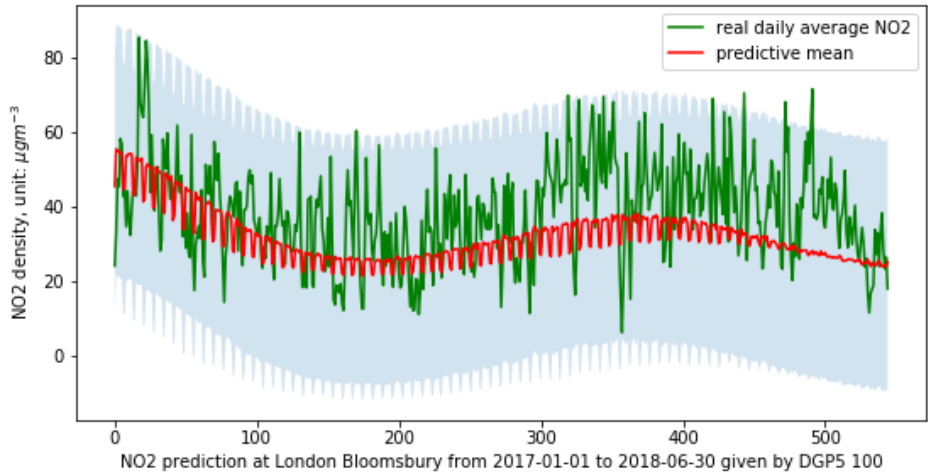
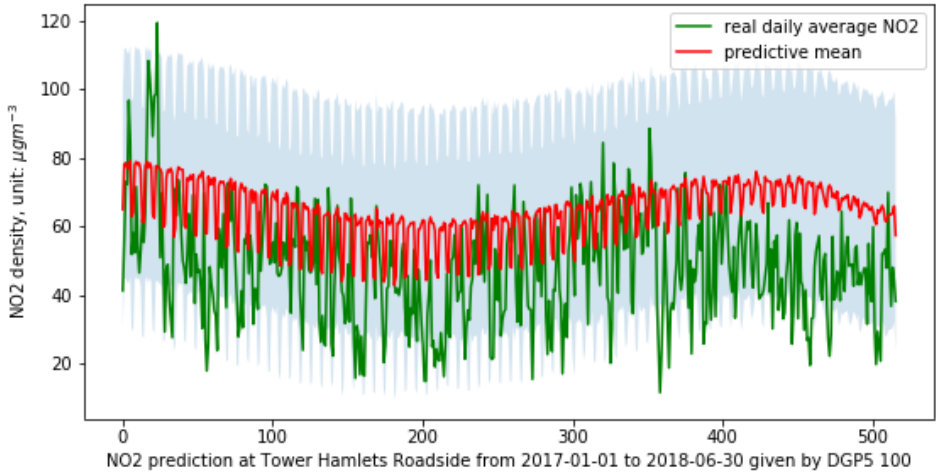


Figure 6.4: Predictions at London Bexley, London Bloomsbury and Tower Halmets Roadside sites, given by DGP5, validation runs with all features.X ticks start from 01/01/2017 and end at 30/06/2018. Each unit in x represents a day.







The unsatisfactory predictions at site London Bexley can be attributed to its environment type: it is a 'Suburban Background' site, and there are only two of them in London. Unfortunately, due to the limited size of London monitoring sites and the environment-type-based stratified sampling technique we employ, London Bexley site is included in four out of five test sets among validation runs, dragging down model performance in these runs. The one experiment whose test set excludes London Bexley reports more favourable results, e.g. an RMSE of 27.976, SRMSE of 0.805 for SGPR model, and an RMSE of 27.581, SRMSE of 0.794 for DGP5 model, which might be more indicative of how well GP models perform under right circumstances.

We include in Figure 6.5 plots of predictions at London Bexley and Tower Halmets Roadside sites, given by DGP5 in one of the spatial extrapolation experiments. Compared with the Bexley plot in 6.6, the one in spatial extrapolation does not witness much improvement, which is unsurprising because of the scarcity of 'Suburban Background' sites - there are only four of them among 144 AURN sites we use. In contrast, Tower Halmets Roadside data gets a substantially better fit, which means that modelling at the global scale is more advantageous than modelling locally, since the GP models can extract information from many more sites sharing the same environment type.

6.2.3 Remark

Reflecting on our discovery that site environment type is far more relevant a feature than geographic coordinates (at least with the data we have), our government-region-based sampling strategy for spatial extrapolation task is perhaps inappropriate.

In Section 6.2.1, we hypothesise that adjacency to sites of same environment type is desirable; however, our experiment with London sites demonstrates that quantity, rather than distance, is more pertinent to the task. Proximity to sites of the same environment type is only good if we already have enough training sites of the same type, scattered or not, to learn the intrinsic patterns of each type.

Once we restrict our attention to London only, we should have used more training data, either by considering more sites or more observations per site. The former is

impossible as there are only 16 sites; we adopt the latter approach in Section 6.4, using the original hourly measurement, but on a different classification task.

6.3 Time Series Forecast Experiments

6.3.1 Model and Kernel Selection

Time series forecast is different as we are performing out-of-domain extrapolation into future. Instead of using RBF kernel alone, a product kernel of periodic kernel and RBF kernel is more suitable as it captures both the trend and fluctuations: the periodic kernel models the seasonal variation, but we do not know if the seasonal trend is precisely periodic; multiplying the periodic kernel with an RBF kernel "allows a *decay away* from periodicity" ([Rasmussen and Williams, 2006]). We first test this idea on a KISS-GP-25 model with spatiotemporal features only. The KISS-GP model with RBF kernel reports an RMSE of 21.578 and SRMSE of 1.380, whereas substituting in a product kernel results in an RMSE of 13.137 and SRMSE of 0.840, a 39% improvement.

To select the best model and kernel combination for our task, we run a series of investigative experiments. Details of the experiments are too lengthy to be included here; readers may find them in Appendix B. Based on results of the exploratory run, we select the following model/kernel combinations for the actual time series forecast experiments:

- PSVGP: SVGP with kernel RBF×Periodic;
- PTSGPR: SGPR with kernel RBF×Periodic, and the periodic kernel is only active on the time dimension;
- PTSVGP: SVGP with kernel RBF×Periodic, and the periodic kernel is only active on the time dimension;
- ARDPTSGPR: SGPR with kernel RBF×Periodic, the RBF kernel has one length-scale hyperparameter per dimension, and periodic kernel is only active on the time dimension;
- ARDPTSVGP: SVGP with kernel RBF×Periodic, the RBF kernel has one length-scale hyperparameter per dimension, and periodic kernel is only active on the

time dimension;

- ARDP2SGPR: SGPR with kernel RBF×Periodic×Periodic, the RBF kernel has one length-scale hyperparameter per dimension, and two periodic kernels are active on the time dimension and the weekday/weekend indicator dimension respectively;
- Deep Gaussian processes: DGP1, DGP3, DGP5.

6.3.2 The Experiment

Results of time series forecast experiments can be found in Table 6.7. The top two models, ARDPTSGPR and ARDP2SGPR are both based on simple FITC approximation to the kernel, yet they still handily beat deep models, thanks to their task-specific kernels.

Surprisingly, the baseline model experimented on the same five training-test pair reports an RMSE of 9.155 ± 0.372 and an SRMSE of 0.647 ± 0.050, which means even the best joint GP model, ARDPTSGPR, is worse than the baseline in terms of RMSE. Figure 6.6 compares the performance of two models on each test set. Although unexpected, the inferior performance can probably be accounted for by the number of (hyper)parameters. For example, ARDPTSGPR only has eleven hyperparameters, two of which (geographic coordinates) are not particularly relevant as we have shown in spatial extrapolation experiments. By contrast, the collection of independent models have 274 hyperparameters, any two of which controls data variability at a local site.

The global GP model discovers site-type-specific patterns, possibly because GP are *linear predictor* or *smoother* that smoothes out the variability of each individual site and groups sites of the same type together to obtain a general predictive mean applicable to all sites of the same type. However, the history of each individual site already encodes all the characteristics of that site, which may or may not have been captured by the global model.

One possibility we could explore is to combine the global and local GP models. We can train a global GP model first, and then use this global GP as the prior for the mean function of each individual site. This approach would preserve the individuality of

Model	RMSE	SRMSE
PSVGP	11.559±0.639	0.806±0.035
PTSGPR	12.077±0.587	0.842±0.029
PTSVGP	12.454±0.717	0.866±0.024
ARDPTSGPR	**9.820 ± 0.554**	**0.685 ± 0.031**
ARDPTSVGP	11.177±0.882	0.797±0.105
ARDP2SGPR	9.849±0.579	0.687±0.033
DGP1	12.550±0.609	0.875±0.030
DGP3	11.079±0.571	0.773±0.028
DGP5	10.786±0.573	0.753±0.034

Table 6.7: Model performance on the time series forecast experiments

Model	Precision	Recall	F_1
ARDPTSGPR	0.274±0.112	0.383±0.153	0.316±0.126
ARDPTSVGP	0.479±0.060	**0.556 ± 0.067**	**0.507 ± 0.059**
ARDP2SGPR	0.273±0.113	0.369±0.149	0.311±0.127
DGP3	0.354±0.092	0.338±0.140	0.279±0.098
DGP5	**0.523 ± 0.076**	0.339±0.122	0.325±0.080

Table 6.8: Model performance on exceedance forecast task

each site while taking into account long-range effects imposed by other sites.

6.4 Forecasting Hourly Exceedance in London

Table 6.8 contains results for the hourly exceedance forecast task. The particularly poor performance for SGPR models and DGP3 is due to the `nan` value they obtain in certain runs, where they fail to make any predictions greater than 100. All `nan` values are manually set to 0.

SVGP model tops the exceedance forecast task, which can be explained by the large size of the dataset and high intra-day variability, as variational methods can discover the optimal set of inducing points that cater to the dataset. The 55.6% recall may not seem high, but if we want to focus on improving the recall, we can adjust our GP models such that the predictions are given by the predictive mean plus one standard

Model	Precision	Recall	F_1
ARDPTSGPR	0.347±0.045	0.718±0.089	0.444±0.043
ARDPTSVGP	0.285±0.060	$\mathbf{0.802 \pm 0.049}$	0.400±0.062
ARDP2SGPR	0.350±0.043	0.718±0.089	$\mathbf{0.449 \pm 0.043}$
DGP3	$\mathbf{0.386 \pm 0.045}$	0.590±0.113	0.425±0.048
DGP5	0.355±0.045	0.630±0.113	0.413±0.050

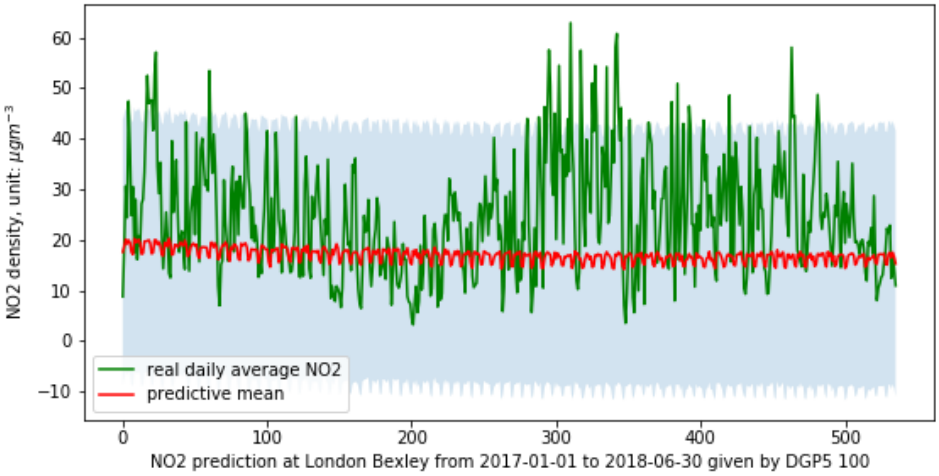
Table 6.9: Model performance on exceedance forecast task. Predictions are given by predictive mean plus one standard deviation

deviation. This adjustment may look like cheating, but since we only care about giving correct categorical predictions, changing the predictions in a systemic manner should be acceptable.

Results for the modified models can be found in Table 6.9. F_1 score witnesses an improvement for all models except SVGP, and SVGP's 80.2% recall rate is more fair. If we are willing to adjust the models further to make predictions two standard deviation away from the predictive mean, we could obtain recall rate as high as 89.5%±3.7%, although too much compromise would be made on the precision.

Figure 6.7 displays performance of the original model and the modified model (giving predictions at predictive mean plus one standard deviation) on each test set, measured by precision, recall and F_1 score. Figure 6.8 and 6.9 illustrate hourly NO_2 concentration predictions at London Bloomsbury and London Marylebone Road sites respectively. SVGP models behave qualitatively different from other two models, and are more consistent with the fluctuations of the real data.

Figure 6.5: Predictions at London Bexley and Tower Halmets Roadside sites, given by DGP5 for the spatial extrapolation runs with all features. X ticks start from 01/01/2017 and end at 30/06/2018. Each unit in x represents a day.



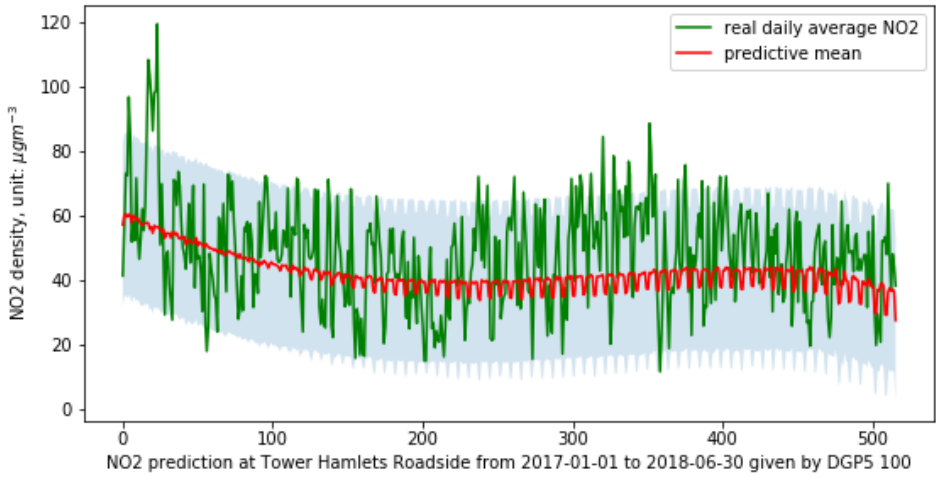


Figure 6.6: Model performance comparison between the baseline and ARDPTSGPR on the time series forecast experiments, over five test sets

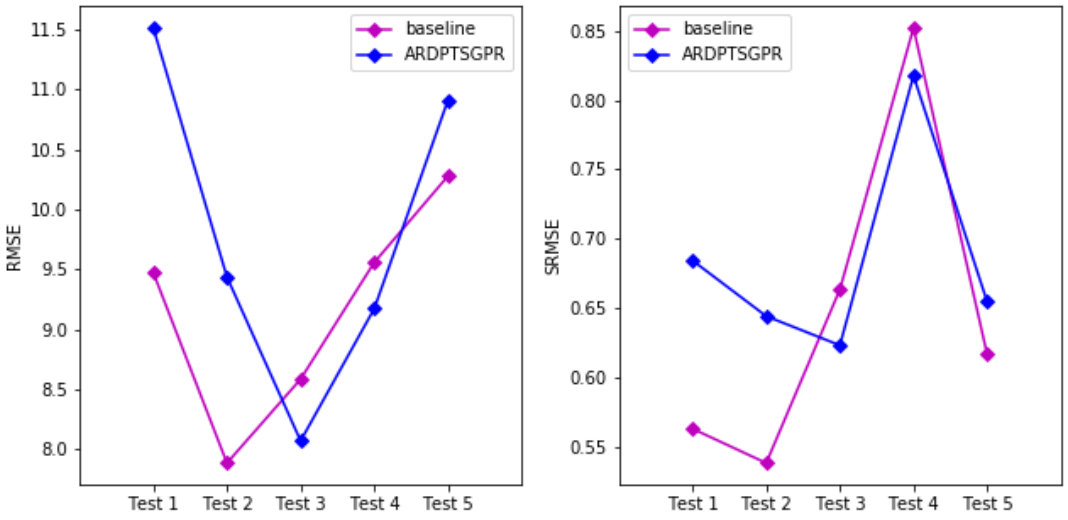


Figure 6.7: Models performance on exceedance forecast task for each test set. The first column represents the unmodified model, and the second column is the modified model. Performance metric from top row to bottom row: precision, recall, F_1 score.

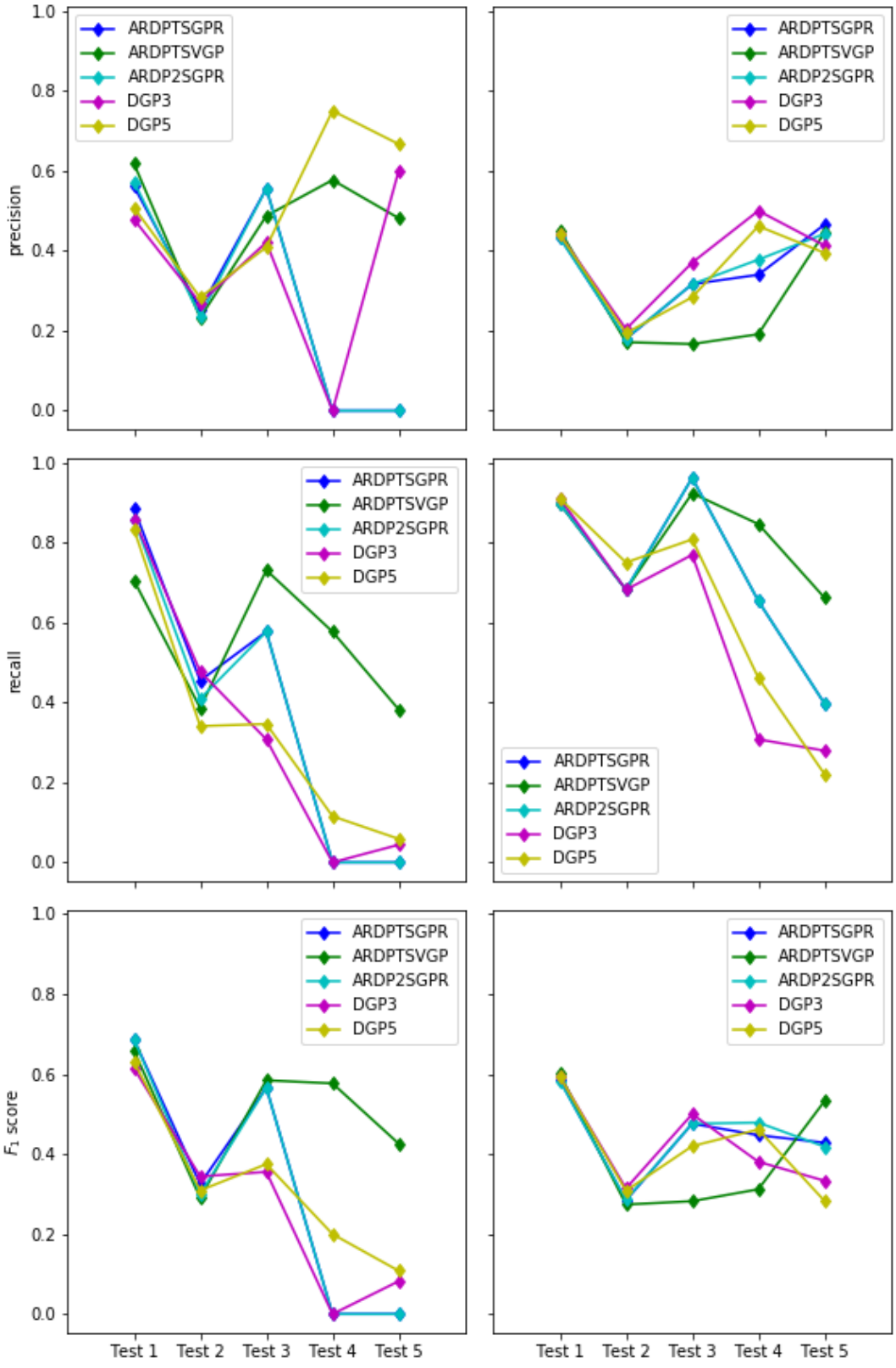


Figure 6.8: Hourly predictions at London Bloomsbury site, given by SGPR, SVGP and DGP5. X ticks start from 01/04/2018 and end at 30/06/2018. Each unit in x represents an hour.

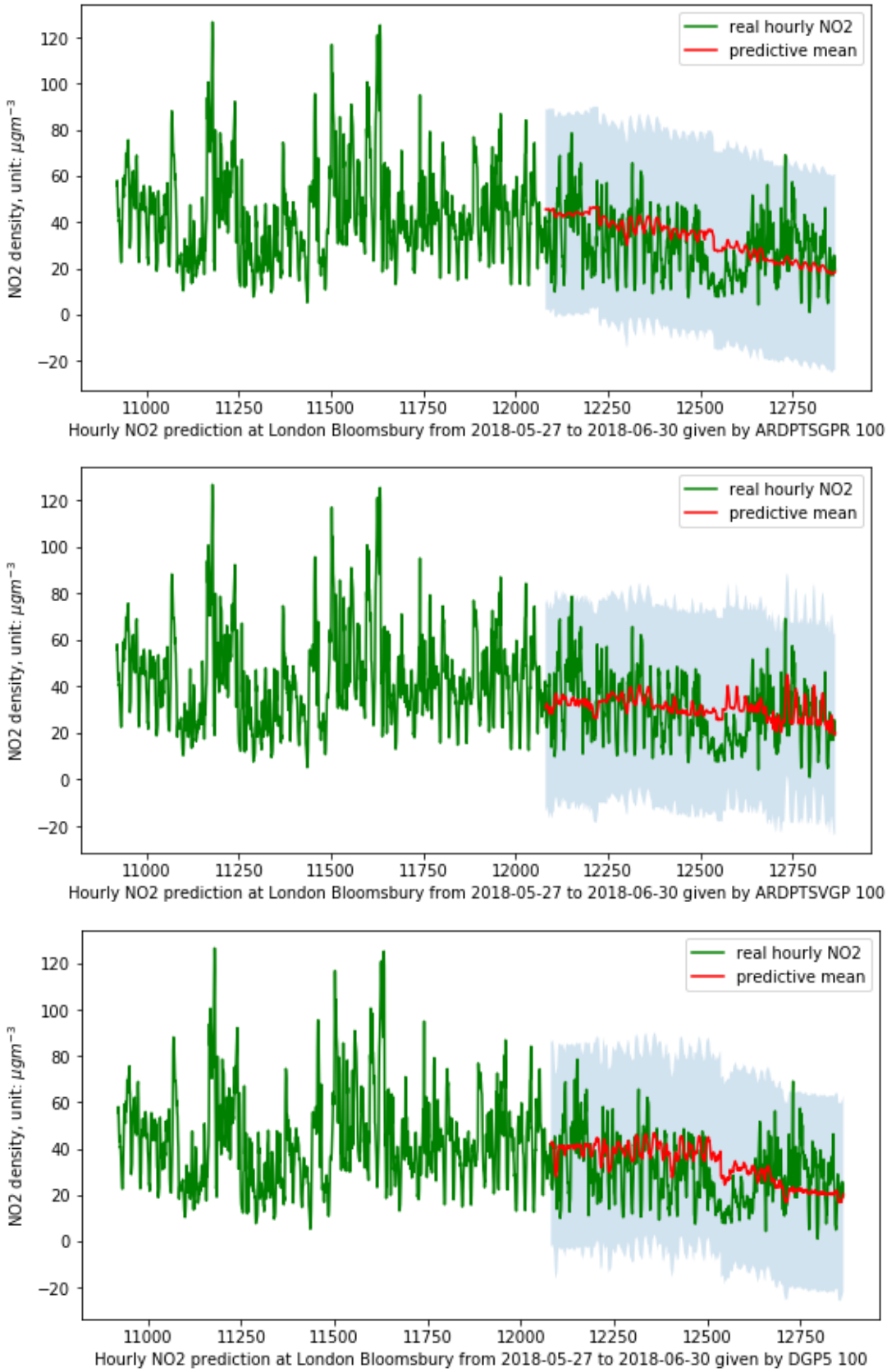
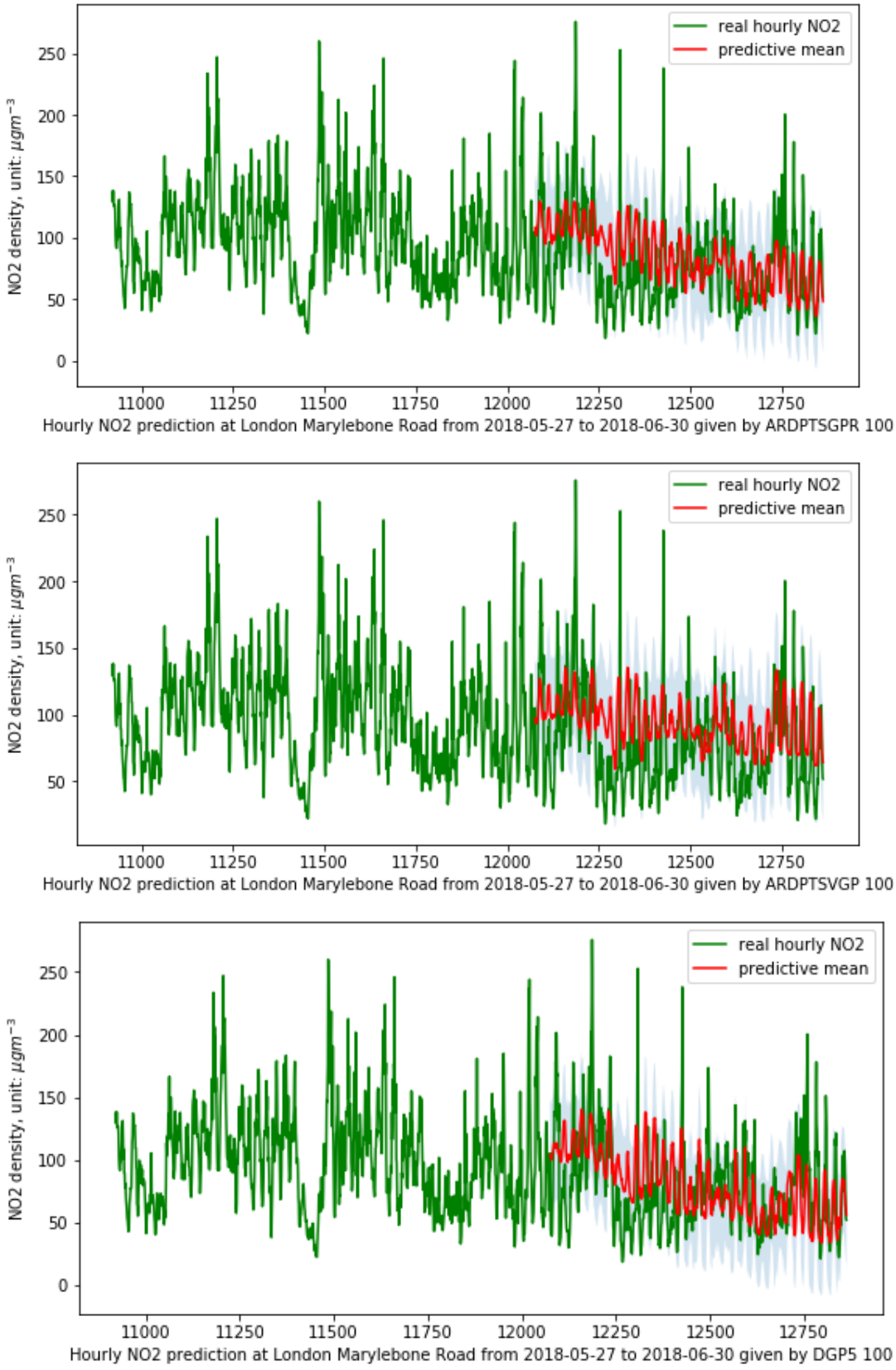


Figure 6.9: Hourly predictions at London Marylebone Road site, given by SGPR, SVGP and DGP5. X ticks start from 01/04/2018 and end at 30/06/2018. Each unit in x represents an hour.



Chapter 7

Conclusion

The aim of this project is to assess the accuracy of results produced by spatiotemporal Gaussian process models when applied to various tasks relating to air pollution data modelling and forecasting. We studied the behaviour of several GP models under four different circumstances:

- Firstly, modelling the distribution of NO_2 concentration across AURN sites;
- Secondly, predicting the distribution of NO_2 concentration in held-out test set selected based on geographic coordinates, both at a nationwide scale and in local metropolitan area;
- Thirdly, forecasting the distribution of NO_2 concentration in held-out test set selected based on time;
- Lastly, forecasting incidence of hourly NO_2 concentration exceeding 100 μgm^{-3} in London.

We showed that Gaussian processes, especially DGP models, are good at performing in-domain predictions such as the modelling task. Although performance of GP on spatial extrapolation task is unsatisfactory compared with previous work done by [Pirani et al., 2014], we discovered that GP models are capable of automatically detecting patterns based on input features. We found out that GP models successfully extract the information that weekdays are usually more polluted than weekends with the help of the weekday/weekend indicator feature. Gaussian processes also appear to learn the impact of different site environment types when trained with enough examples, but fail horribly if there is a shortage of one particular type of sites.

Our experiments with time series forecast did not produce the desired result, which is possibly due to the difference in model representation power caused by the difference in the number of hyperparameters, which we discussed in Section 6.3.2. The qualitatively smooth predictions that we observe in almost all figures (with the exception of predictions in exceedance forecast experiments), prompt us to regard our global GP models as a linear smoother. Even though our global spatiotemporal GP models did not produce the most faithful predictions, they are still good at discovering seasonal trends. Hence, a suitable application of global-scale Gaussian processes is modelling long-term tendencies.

We consider our exceedance forecast task a moderate success, having obtained over 80% recall rate. It is reasonable to assume that GP are better at discovering trends than predicting precise values, hence the good results for classification. Comparing the plots we have in London spatial extrapolation experiments with the ones in exceedance forecast experiments, GP models (especially SVGP) can discover more volatile patterns, when trained with enough data at several adjacent sites.

7.1 Limitations of Our Work

It is worthwhile to point out flaws and limitations of our work so that we can investigate possible further works and improvement.

Firstly, not all pollutant are domestically sourced; trans-border movement of air pollutants cannot be modelled by GP, as we cannot find appropriate features accounting for it.

Secondly, the scale of spatial features mismatches the scale of the time. We were modelling sites across the whole Great Britain but only considered a period of one year and a half. We should have either taken a long-term view and model seasonal or annual trends, or confine the area of interest to a much smaller region, like what we did with hourly exceedance forecast in London. The problem with modelling long-term trend is mainly availability of data, as some sites were not in operation back then, and some sites that were active in the past had been discontinued.

7.2 Possible Future Work

As mentioned earlier, we can experiment with applying a global GP prior to the mean functions of local GP models to take advantage of the global information such as environment-type-specific characteristics while also remain pertinent to local variabilities and individualities.

We can use more spatially dense data. We have seen in spatial extrapolation experiments that correlations inferred by geographic coordinates are only secondary to more important features such as site environment types. We can make use of high-resolution satellite grid data to learn more about the mid-long range pollutant transport, although the computational cost would be high even with scalable models. Another choice is to use data collected by cheap, mobile sensors, and this approach will enable us to obtain highly dense spatiotemporal data of pollutants in a small area. However, readings of these mobile monitors can be inaccurate, requiring us to perform calibrations. [Lin et al., 2017]

We can also incorporate more relevant features into our model. For example, the temperature feature we used in exceedance forecast. Another readily available feature is wind speed at monitoring sites.

Appendix A

Mathematical Background

A.1 Matrix Identities

The *matrix inversion lemma*, or Woodbury-Sherman-Morrison formula, states

$$(Z+UWV^{\mathrm{T}})^{-1} = Z^{-1} - Z^{-1}U(W^{-1}+V^{\mathrm{T}}Z^{-1}U)^{-1}V^{\mathrm{T}}Z^{-1} \tag{A.1}$$

assuming all relevant inverses exist.

A.1.1 Matrix Derivatives

Derivatives of an inverse matrix with respect to parameter θ:

$$\frac{\partial}{\partial\theta}K^{-1} = -K^{-1}\frac{\partial K}{\partial\theta}K^{-1}, \tag{A.2}$$

where $\frac{\partial K}{\partial\theta}$ is a matrix of element-wise derivatives.

Derivatives of the log determinant of a positive definite symmetric matrix with respect to parameter θ:

$$\frac{\partial}{\partial\theta}\log|K| = \mathrm{tr}(K^{-1}\frac{\partial K}{\partial\theta}). \tag{A.3}$$

Appendix B

Model and Kernel Selection for Time Series Forecast Experiments

Table B.1 and B.2 report results of our exploratory runs with spatiotemporal features only and with all features, respectively. To conduct these experiments, we first sort the dataset by time, and then make an 80%/20% training-test split, using the first 80% as training set so that we train our models on the past and predict the future. The prefixes in model names represent different modifications to the kernel function of 'base' models.

- P represents a product kernel of RBF kernel and periodic kernel;
- PT represents a product kernel of RBF kernel and periodic kernel, but the periodic kernel is only active in the time dimension;
- ARDPT represents a product kernel of RBF kernel and periodic kernel, but the periodic kernel is only active in the time dimension, and the RBF kernel has automatic relevance determination (ARD) on, i.e. it has one distinct length-scale for each input dimension;
- ARDP2 represents a product kernel of one RBF kernel and two periodic kernels. The RBF kernel is in ARD state, and the two periodic kernels are active in the time dimension, and the weekday/weekend indicator feature dimension, respectively.

Model	RMSE	SRMSE
SGPR 100	15.0361	0.9619
SVGP 100	14.7058	0.9408
FITC 100	14.7629	0.9444
PSGPR 100	22.6458	1.4487
PSVGP 100	21.0474	1.3465
PFITC 100	13.5650	0.8678
PTSGPR 100	14.7532	0.9438
PTSVGP 100	14.8281	0.9486
PTFITC 100	15.1119	0.9668
ARDPTSGPR 100	**11.3982**	**0.7292**
ARDPTSVGP 100	12.5322	0.8017
ARDPTFITC 100	13.8614	0.8868
DGP1 100	14.6718	0.9386
DGP3 100	13.8057	0.8832
DGP5 100	13.6155	0.8710

Table B.1: Time series forecast exploratory runs with spatiotemporal features only

Model	RMSE	SRMSE
SGPR 100	13.0098	0.8323
SVGP 100	13.3167	0.8519
FITC 100	13.7958	0.8826
PSGPR 100	14.3602	0.9187
PSVGP 100	11.3523	0.7262
PFITC 100	14.3815	0.9200
PTSGPR 100	12.9849	0.8307
PTSVGP 100	13.2130	0.8453
PTFITC 100	13.6160	0.8711
ARDPTSGPR 100	10.4902	0.6711
ARDPTSVGP 100	10.5932	0.6777
ARDPTFITC 100	12.9035	0.8255
ARDP2SGPR 100	**10.4110**	**0.6660**
ARDP2SVGP 100	10.6341	0.6803
DGP1 100	13.3414	0.8535
DGP3 100	11.5943	0.7417
DGP5 100	11.7084	0.7490

Table B.2: Time series forecast exploratory runs with all features

Appendix C

Map of Eligible London Sites

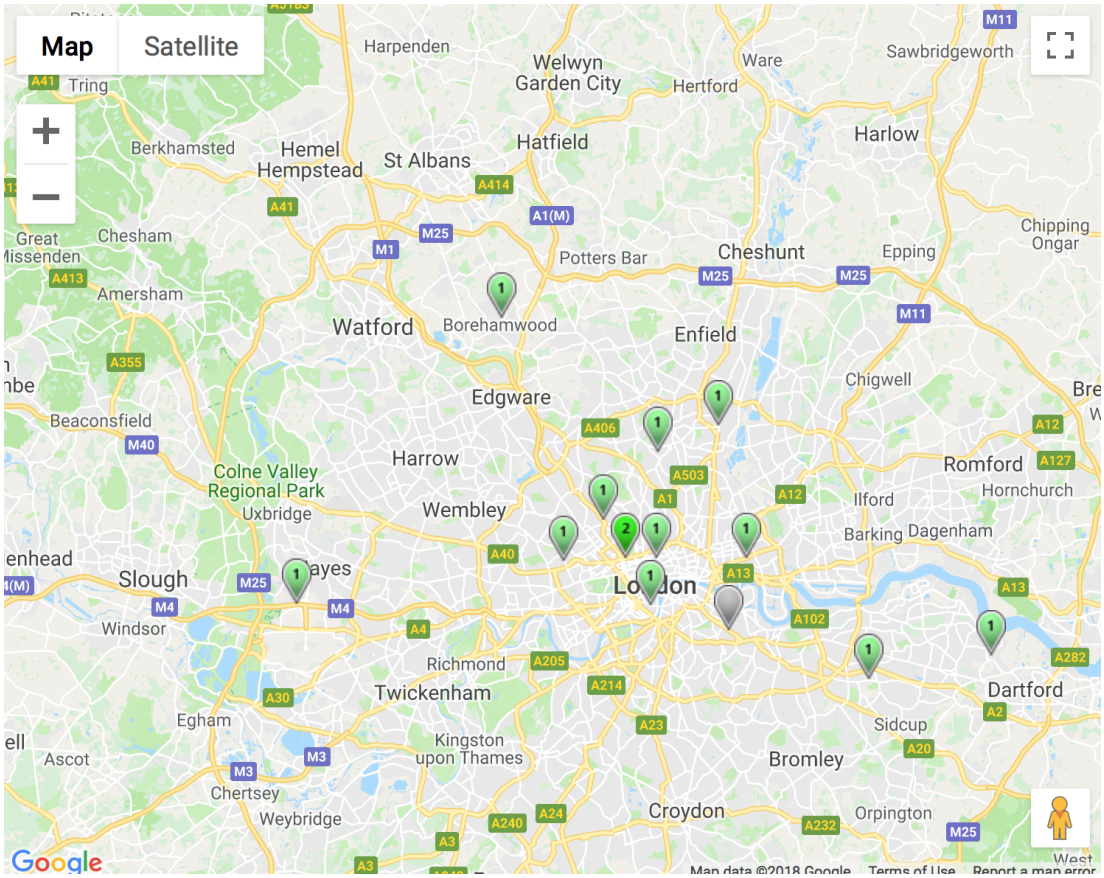
C.1 London sites

Here is a list of eligible AURN sites in London, for our spatial extrapolation experiments.

- Camden Kerbside
- Haringey Roadside
- London Bexley
- London Bloomsbury
- London Eltham
- London Haringey Priory Park South
- London Hillingdon
- London Marylebone Road
- London N. Kensington
- London Westminster
- Southwark A2 Old Kent Road
- Tower Hamlets Roadside

C.2 Map

Figure C.1: Map showing the location of 12 eligible London sites for spatial extrapolation experiments. Only the green ones are active. Source: DEFRA, Google Map



Bibliography

[Byun and Schere, 2006] Byun, D. and Schere, K. L. (2006). Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (cmaq) modeling system. *Applied Mechanics Review*, 59(2):51–77.

[Carruthers et al., 1994] Carruthers, D., Holroyd, R., Hunt, J., Weng, W., Robins, A., Apsley, D., Thompson, D., and Smith, F. (1994). Uk-adms: A new approach to modelling dispersion in the earth's atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 52:139–153.

[Cocchi et al., 2007] Cocchi, D., Greco, F., and Trivisano, C. (2007). Hierarchical space-time modelling of pm10 pollution. *Atmospheric Environment*, 41:532–542.

[Damianou and Lawrence, 2013] Damianou, A. C. and Lawrence, N. (2013). Deep gaussian processes. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*.

[DEFRA, 2011] DEFRA (2011). The air quality strategy for england, scotland, wales and northern ireland. Technical report, Department for Environment, Food and Rural Affairs, Government of the United Kingdom.

[DEFRA, 2017] DEFRA (2017). Air pollution in the uk 2016. Technical report, Department for Environment, Food and Rural Affairs, Government of the United Kingdom.

[EU, 2004] EU (2004). Directive 2004/107/ec of the european parliament and of the council of 15 december 2004 relating to arsenic, cadmium, mercury, nickel and polycyclic aromatic hydrocarbons in ambient air. Technical report, The European Parliament and the Council of the European Union.

[EU, 2008] EU (2008). Directive 2008/50/ec of the european parliament and of the council of 21 may 2008 on ambient air quality and cleaner air for europe. Technical report, The European Parliament and the Council of the European Union.

[Freeman et al., 2018] Freeman, B., Taylor, G., Gharabaghi, B., and Thé, J. (2018). Forecasting air quality time series using deep learning. *Journal of the Air and Waste Management Association*, 68(8):866–886.

[Gardner et al., 2018] Gardner, J. R., Pleiss, G., Wu, R., Weinberger, K. Q., and Wilson, A. G. (2018). Product kernel interpolation for scalable gaussian processes. *AISTATS*.

[Ghosal and Roy, 2006] Ghosal, S. and Roy, A. (2006). Posterior consistency of gaussian process prior for nonparametric binary regression. *Annals of Statistics*, 34:2413–2429.

[Gocheva-Ilieva et al., 2014] Gocheva-Ilieva, S., Ivanov, A., Voynikova, D., and Boyadzhiev, D. (2014). Time series analysis and forecasting for air pollution in small urban area: an sarima and factor analysis approach. *Stochastic Environmental Research and Risk Assessment*, 28:1045–1060.

[Gulliver et al., 2011] Gulliver, J., Morris, C., Lee, K., Vienneau, D., Briggs, D., and Hansell, A. (2011). Land use regression modeling to estimate historic (1962-1991) concentrations of black smoke and sulfur dioxide for great britain. *Environmental Science and Technology*, 45:3526–3532.

[Hensman et al., 2013] Hensman, J., Fusi, N., and Lawrence, N. (2013). Gaussian processes for big data. *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*.

[Hensman et al., 2012] Hensman, J., Rattray, M., , and Lawrence, N. (2012). Fast variational inference in the exponential family. *NIPS 2012*.

[Hoek et al., 2008] Hoek, G., Beelen, R., de Hoogh, K., Vienneau, D., Gulliver, J., Fischer, P., and Briggs, D. (2008). A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric Environment*, 42:7561–7578.

[Huerta et al., 2004] Huerta, G., Sanso, B., and Stroud, J. (2004). A spatiotemporal model for mexico city ozone levels. *Journal of the Royal Statistical Society: Series C*, 53:231–248.

[Keys, 1981] Keys, R. G. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160.

[Lawrence et al., 2003] Lawrence, N., Seeger, M., and Herbrich, R. (2003). Fast sparse gaussian process methods: the informative vector machine. *Advances in Neural Information Processing Systems 15*, pages 625–632.

[Lin et al., 2017] Lin, C., Masey, N., Wu, H., Jackson, M., Carruthers, D., Reis, S., Doherty, R. M., Beverland, I. J., and Heal, M. R. (2017). Practical field calibration of portable monitors for mobile measurements of multiple air pollutants. *Atmosphere*, 8:231.

[MacKay, 1994] MacKay, D. (1994). Bayesian nonlinear modeling for the prediction competition. *Ashrae Transactions*, 100(2):1053–1062.

[MacKay, 1998] MacKay, D. (1998). Introduction to gaussian processes. In Bishop, C., editor, *Neural Networks and Machine Learning*, chapter 11, pages 133–165. Springer-Verlag.

[Matthews et al., 2017] Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

[McHugh et al., 1997] McHugh, C., Carruthers, D., and Edmunds, H. (1997). Adms-urban: an air quality management system for traffic, domestic and industrial pollution. *International Journal of Environment and Pollution*, 8.

[McMillan et al., 2010] McMillan, N., Holland, D., Morara, M., and Feng, J. (2010). Combining numerical model output and particulate data using bayesian space-time modeling. *Environmetrics*, 21:48–65.

[Neal, 1996] Neal, R. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag.

[Niska et al., 2004] Niska, H., Hiltunen, T., Karppinen, A., Ruuskanen, J., and Kolehmainen, M. (2004). Evolving the neural network model for forecasting air pollution time series. *Engineering Applications of Artificial Intelligence*, 17:159–167.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pirani et al., 2014] Pirani, M., Gulliver, J., Fuller, G. W., and Blangiardo, M. (2014). Bayesian spatiotemporal modelling for the assessment of short-term exposure to particle pollution in urban areas. *Journal of Exposure Science and Environmental Epidemiology*, 24:319–327.

[Pleiss et al., 2018] Pleiss, G., Gardner, J. R., Weinberger, K. Q., and Wilson, A. G. (2018). Constant-time predictive distributions for gaussian processes. *ICML*.

[Quiñonero-Candela and Rasmussen, 2005] Quiñonero-Candela, J. and Rasmussen, C. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

[Rasmussen and Williams, 2006] Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

[Saatchi, 2011] Saatchi, Y. (2011). *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge.

[Sahu et al., 2007] Sahu, S., Gelfand, A., and Holland, D. (2007). High resolution space-time ozone modeling for assessing trends. *Journal of the American Statistical Association*, 102:1221–1234.

[Sahu et al., 2010] Sahu, S., Gelfand, A., and Holland, D. (2010). Fusing point and area level space-time data with application to wet deposition. *Journal of the Royal Statistical Society: Series C*, 59:77–103.

[Salimbeni and Deisenroth, 2017] Salimbeni, H. and Deisenroth, M. P. (2017). Doubly stochastic variational inference for deep gaussian processes. *31st Conference on Neural Information Processing Systems*.

[Salimbeni et al., 2018] Salimbeni, H., Eleftheriadis, S., and Hensman, J. (2018). Natural gradients in practice: Non-conjugate variational inference in gaussian process models. *Artificial Intelligence and Statistics*.

[Seeger, 2003] Seeger, M. (2003). *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations*. PhD thesis, School of Informatics, University of Edinburgh.

[Shaddick and Wakefield, 2002] Shaddick, G. and Wakefield, J. (2002). Modelling daily multivariate pollutant data at multiple sites. *Journal of the Royal Statistical Society: Series C*, 51:351–372.

[Shepard, 1968] Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *proceedings of the 1968 ACM National Conference*, pages 517–524.

[Silverman, 1985] Silverman, B. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society: Series B*, 47(1):1–52.

[Singles et al., 1998] Singles, R., Sutton, M., and Weston, K. (1998). A multi-layer model to describe the atmospheric transport and deposition of ammonia in great britain. *Atmospheric Environment*, 32(3):393–399.

[Snelson and Ghahramani, 2006] Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1257.

[Titsias, 2009] Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*.

[Tresp, 2000] Tresp, V. (2000). A bayesian committee machine. *Neural Computation*, 12(11):2719–2741.

[Walton et al., 2015] Walton, H., Dajnak, D., Beevers, S., Williams, M., Watkiss, P., and Hunt, A. (2015). *Understanding the Health Impacts of Air Pollution in London*. King's College London.

[Williams and Seeger, 2001] Williams, C. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 13*, pages 682–688.

[Wilson, 2014] Wilson, A. G. (2014). *Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes*. PhD thesis, University of Cambridge.

[Wilson et al., 2015] Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2015). Deep kernel learning. *https://arxiv.org/abs/1511.02222*.

[Wilson and Nickisch, 2015] Wilson, A. G. and Nickisch, H. (2015). Kernel interpolation for scalable structured gaussian processes. *Proceedings of the 32nd International Conference on Machine Learning*.