KERNEL METHODS FOR LEARNING ABOUT COMPLEX DYNAMICAL SYSTEMS

Thesis by

Dmitry Burov

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended May 13, 2024

Dmitry Burov
ORCID: 0000-0002-5060-6794

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Professor Andrew M. Stuart, for his unwavering support, guidance and help throughout my whole PhD journey. Working with him was a pleasure and an honor, especially because we shared many research traits: attention to detail, preference of simple yet instructive problems over complex ones, miminalism in many aspects. Working through problems together, discussing them at the blackboard, sharing results on the TV in the studio, and helping as a teaching assistant in 107 will forever remain as some of the happiest memories. In these years, I learned not just the academic, but also many life lessons from him. At times, it was difficult to see the end, and Professor Stuart's kindness and *very stubborn* faith in me (quite often, surpassing my own!) has always given me extra encouragement to keep going, and all that despite my annoying character quirks and inconsistent behavior! Simply put, this thesis would not have happened if it was not for him. Thank you, Andrew.

I would also like to thank my family for their unconditional love and support, and especially my mom, whose faith in me *always* surpassed mine, and who always had the right words at the right moment. She is the reason I succeeded in many things in my life, including coming to Caltech in the first place. Not a single day, not even a single second did I feel that I was alone in this, because she truly shared the burden of the whole graduate school with me. Love you, mom.

Finally, I certainly would not have survived without my friends. I am truly blessed with many dear friends, here, back home, and elsewhere, and the time spent together, all the lunches, teas, parties, trips, concerts, pubs, board games, jokes, laughs, singing, dancing, skiing, surfing, grilling and so on — all that kept me sane, smiling and happy. Cheers!

ABSTRACT

The ubiquitous spread of machine learning tools in natural sciences in recent years has seen trully exponential growth. What sounded like an expression from a sci-fi novel mere 7 years ago, "solving PDEs with machine learning" is hardly surprising to anyone today. The variety of methods is very large, but most of them revolve around the artificial neural networks. Despite tremendous success of applications to problems in natural sciences, and despite many strides towards a fundamental theory of neural networks, they still often lack interpretability and robustness of the results. An alternative, much narrower class of machine learning algorithms is comprised of the kernel methods. These methods, in contrast, offer deep analytical theory, with many approximation results and interpretable components. The firm foundation of the kernel methods, however, is offset by the practical difficulties, such as high computational cost, the burden of high-dimensional optimization and the necessity to manually choose kernel parametrization. This thesis explores a few applications of the kernel methods to dynamical systems, with the goal to address some of those issues. The comparison between the kernel analog forecasting and the plain Gaussian process regression is made, both from theoretical and practical sides, and a parametric extension of the former is proposed. An application of kernel methods to closures of dynamical systems is showcased. Finally, an application of data assimilation machinery to an epidemiological model is shown.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Dmitry Burov, Dimitrios Giannakis, Krithika Manohar, and Andrew Stuart. "Kernel Analog Forecasting: Multiscale Test Problems." In: *Multiscale Modeling & Simulation* 19.2 (2021), pp. 1011–1040. DOI: 10.1137/20M1338289.

D. B. wrote numerical code for several dynamical systems and Gaussian-process closures, performed simulations and provided theoretical explanations, and helped write Sections 4 and 5 (Averaging and Conclusions).

[2] Tapio Schneider et al. "Epidemic management and control through risk-dependent individual contact interventions." In: *PLOS Computational Biology* 18.6 (June 2022), pp. 1–32. DOI: 10.1371/journal.pcbi.1010171.

D. B. wrote extensive amounts of project's code, such as integration of ordinary differential equations, various data assimilation (DA) algorithms and graph operations; conducted numerical experiments, participated in developement of specific to the problem DA methods, investigation of the results and their interpretation.

CONTENTS

1 Introduction 1
1.1 Overview of the thesis 1
1.2 Broader context 3

I Kernel Methods in Dynamical Systems Forecasting
2 Theoretical Analysis of Kernel Analog Forecasting 6
2.1 Introduction 7
2.1.1 Notation and RKHS Theory Basics 7
2.1.2 KAF Theory 12
2.2 GP Interpretation of KAF 18
2.2.1 Problem Formulation 18
2.2.2 Overview of the KAF Methodology 19
2.2.3 Second Step of KAF is Truncated-Spectrum GP Regression 20
2.2.4 Three Modalities 24
2.2.5 Least-Squares Derivation of KAF 29
2.3 Analyis of KAF Properties 30
2.3.1 Interpolation Property 30
3 Numerical Experiments Illustrating KAF 33
3.1 Introduction 33
3.2 Test Problems and Data Sets 35
3.3 Comparison of GPR versus Single-switch-on Variants 37
3.3.1 Experiment I 38
3.3.2 Experiment II 41
3.3.3 Experiment III 42
4 Parametric Extension of the KAF 44
4.1 Introduction to Parametric Dependence 44
4.2 Organization and summary of the results 44
4.3 Test problems 45
4.3.1 Harmonic oscillator 46
4.3.2 Lorenz '63 47
4.4 Explicit Parameter Values 49
4.4.1 Fixed Parameters; One Time Series 49
4.4.2 Fixed Parameters; Multiple Time Series 50
4.4.3 Variable Parameters; Multiple Time Series 54
4.4.4 Case Studies 55
4.4.5 Harmonic Oscillator 55

4.4.6 Lorenz '63 56
4.5 Implicit Parameter Values 59
4.5.1 Delay Embedding 59
4.5.2 KAF with Delay Embedding 61
4.5.3 Case Studies 62
4.5.4 Harmonic Oscillator 62
4.5.5 Lorenz '63 62
4.6 Conclusion and further directions 62

II Data-Driven Model Augmentation
5 Closures for Multiscale Systems Using Kernel Methods 66
5.1 Kernel-based closures 67
5.1.1 The Model 67
5.1.2 Closure designs 69
5.2 Forecasting comparisons 72
5.2.1 Conditional Expectation and Variance 72
5.2.2 Comparison Of Data-Driven And Model-Data-Driven Prediction 75
5.2.3 Non-Markovian regime 79
5.2.4 Comparison with Lorenz' method 79
6 Graph-based Epidemiological Models with Data Assimilation 81
6.1 Introduction 83
6.2 Network model and equations 86
6.2.1 SEIHRD model on a contact network 87
6.2.2 Reduced master equations 91
6.2.3 Closure of reduced master equations 92
6.3 Data assimilation 95
6.3.1 The algorithm 95
6.3.2 Testing strategy 97
6.3.3 Parameter Learning 97
6.3.4 Classification in Risk Network 99
6.3.5 Classification in testing, contact tracing, and isolation (TTI) 100
6.3.6 Contact Interventions 100
6.4 Synthetic network for proof-of-concept 101
6.4.1 Synthetic network for proof-of-concept 102
6.4.2 Selection of subnetwork for user base 105
6.4.3 Surrogate world simulation 106
6.4.4 Synthetic data 107
6.5 Large-sized network numerical results 110
6.5.1 Lockdown and world avoided 110
6.5.2 Accuracy of individual risk assessment 112
6.5.3 Risk-tailored contact interventions 117

6.5.4 Discussion 127

Bibliography 130

LIST OF FIGURES

Figure 2.1 Commutative diagram of the spaces in the dynamical systems perspective. 14
Figure 2.2 Examples of the self-tuning kernels. 25
Figure 2.3 Function $w(x) = \frac{x}{x+\beta}$ plotted for various values of β. 31
Figure 3.1 Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 0.001$. 39
Figure 3.2 The VB kernel with a poorly-tuned sparsification parameter. 40
Figure 3.3 Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 1.0$. 40
Figure 3.4 The VB kernel with a poorly-tuned sparsification parameter. 40
Figure 3.5 Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 300\,000$. 41
Figure 3.6 Result of applying regression without (left) and with (right) normalization. 41
Figure 3.7 Result of applying regression without (left) and with (right) normalization. 42
Figure 3.8 Result of applying regression without (left) and with (right) truncation. 42
Figure 3.9 Spectrum of the kernel matrix, evaluated on El Niño–Southern Oscillation (ENSO) dataset (first 400 values). 43
Figure 4.1 Lorenz '63 attractors resulting from various values of ρ. 47
Figure 4.2 Lorenz '63 attractors resulting from various values of σ. 48
Figure 4.3 Same as Figure 4.2 (with same coloring), but 2-d projections onto $3x = 5y$ plane. 48
Figure 4.4 Comparison of d-KAF (4.6) and cc-KAF (4.7). 51
Figure 4.5 Comparison of the average (purple) of two GP regressors each trained on half points vs one trained on all (black), ordered by decreasing length scale. 53
Figure 4.6 RMSE for (HO-AMP) test problem, forecasting for $E = 2.0$ and $E = 2.5$. 55

Figure 4.7 RMSE for (HO-FREQ) test problem, forecasting for $\kappa = 1.6$ and $\kappa = 2.0$. 55
Figure 4.8 RMSE for (HO-PH) test problem, forecasting for $\alpha = 0.0$ and $\alpha = 0.6$. 56
Figure 4.9 Variable parameters; multiple time series prediction of $x_0 \approx 15$. 57
Figure 4.10 RMSE for (L-SIGMA) test problem, forecasting for $\sigma = 18.0$ and $\alpha = 14.0$. 58
Figure 4.11 Comparison of cc-KAF (4.7) and p-KAF (4.9), for parameter value $\rho = 48$ which not in the training set; both have $\ell = 1000$ eigenpairs. 58
Figure 4.12 Applying kernel analog forecasting (KAF) to harmonic oscillator (HO) with delay-embedding dimension 24. 63
Figure 4.13 Applying KAF to Lorenz '63 (L-63) with delay-embedding dimension 5. 63
Figure 5.1 Coupling diagram for Lorenz '96 multiscale (L-96). 68
Figure 5.2 L-96 regimes of increasing complexity (left to right). 69
Figure 5.3 Gaussian Process Regression. 70
Figure 5.4 L-96 numerical simulations. 71
Figure 5.5 Probability density functions of L-96 and the system with offline closure. 72
Figure 5.6 Online GPR iterations: $i = 0$ through $i = 3$. 73
Figure 5.7 Predictability: periodic, quasiperiodic and chaotic regimes. 74
Figure 5.8 root-mean-square error (RMSE). 75
Figure 5.9 Mean of Gaussian process regression as a closure. 76
Figure 5.10 Comparison of data-driven and model-data-driven prediction. 77
Figure 5.11 RMSE comparison for the four cases a)–d) described in the text, in the periodic, quasiperiodic, and chaotic regimes. 78
Figure 5.12 Prediction in non-Markovian regime. 79
Figure 5.13 Lorenz' method (5.11) vs. KAF. 80
Figure 6.1 Schematic of the personalized risk assessment platform. 84
Figure 6.2 Schematic of SEIHRD model [10]. 86
Figure 6.3 Overall epidemic dynamics from SEIHRD model using mean-field approximation. 93
Figure 6.4 Overall epidemic dynamics from SEIHRD model using mean-field approximation with ensemble correction. 93

Figure 6.5 Histograms of correction coefficients (top row) $\mathscr{C}_M\left[S_i(t), I_j(t)\right]$ and (bottom row) $\mathscr{C}_M\left[S_i(t), H_j(t)\right]$ at different times during the simulated epidemic. 94
Figure 6.6 Distribution of ensemble averaged model parameters across nodes as a function of time during the epidemic. 98
Figure 6.7 Distribution of degrees k in synthetic contact network with 97,942 nodes. 101
Figure 6.8 Dynamic contact network behavior in the first five simulated days, batched into 3-hour windows (starting at midnight). 103
Figure 6.9 Illustration of different user base topologies. 105
Figure 6.10 Evolution of an outbreak in surrogate-world simulations with a lockdown (blue) and without (orange). 111
Figure 6.11 Hospitalization rates in surrogate-world simulation with a lockdown (blue) and without (orange). 112
Figure 6.12 ROC curves for classification as possibly infectious. 114
Figure 6.13 Receiver operating characteristic (ROC) curves for classification as possibly infectious. 115
Figure 6.14 Comparison of different contact intervention scenarios for full user base with $\tilde{N}/N = 100\%$. 119
Figure 6.15 Comparison of different contact intervention scenarios for a user base with $\tilde{N}/N = 75\%$. 120
Figure 6.16 Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 75\%$. 121
Figure 6.17 As in Figure 6.15, but with constant exterior connectivity. 122
Figure 6.18 As in Figure 6.16, but with constant exterior connectivity. 123
Figure 6.19 Comparison of different contact intervention scenarios for neighborhood user base with $\tilde{N}/N = 50\%$ and with a classification threshold $c_I = 0.5\%$. 123
Figure 6.20 Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 50\%$ and with a lower classification threshold $c_I = 0.25\%$. 124
Figure 6.21 Comparison of different contact intervention scenarios for neighborhood user base with $\tilde{N}/N = 25\%$ and with a classification threshold $c_I = 0.25\%$. 124
Figure 6.22 Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 25\%$ and with a lower classification threshold $c_I = 0.01\%$. 125

Figure 6.23 Cumulative death rate of users vs. non-users for the $\tilde{N}/N = 25\%$ user base consisting of nodes selected at random from the overall population network. 126
Figure 6.24 Cumulative death rate of users vs. non-users for the $\tilde{N}/N = 25\%$ user base consisting of neighborhoods in the overall population network. 126

LIST OF TABLES

Table 2.1 Three KAF modalities summarized. 29
Table 3.1 Eight variants, spanning the "distance" between GPR and KAF. 34
Table 4.1 Parameter values used to generate training and testing data sets for HO experiments. 46
Table 4.2 Parameter values used to generate training and testing data sets for L-63 experiments. 48
Table 5.1 GP parameter bounds, initial and typical posterior values. 70
Table 6.1 Details of the different user bases. The percentage represents approximately $\tilde{N}/N$, for the user population $\tilde{N}$. The interior defines how many users are completely surrounded by other users. The exterior connectivity gives the average number of exterior nodes connected to a node inside the user base. 106
Table 6.2 Mean transmission and transition rates and maximum/minimum contact rates for the surrogate-world simulations with the stochastic SEIHRD model (Fig. 6.2). The mean rates are taken to be the same for all nodes; hence, the nodal indices are suppressed. The latent period σ^{-1} and duration of infectiousness in the community γ^{-1} are approximated from those in refs. [60] and [46]; the duration of hospitalization γ'^{-1} is from ref. [86], and the transmission rate β is fit to be consistent with data for respiratory viruses [89] and to roughly reproduce NYC data. 107

Table 6.3 Age-dependent mean hospitalization and mortality rates in the surrogate-world simulation. The share $f(a)$ of the population in each age group a is taken from U.S. Census data [107]. The age-dependent death rate in hospitals d' is obtained from cumulative hospitalization and death rates in NYC by June 1, 2020 [74], under the assumption that 90% of deaths occurred in hospitals. Age-dependent hospitalization rates $h(a)$ and mortality rates $d(a)$ in the community (outside hospitals) are difficult to obtain directly from NYC data because of an age-dependent undercount of infections [118]. We choose hospitalization rates $h(a)$ that approximate data from France [90], adjusting the rates so that the overall hospitalization rate is $\sum_a f(a)h(a) \approx 3.1\%$, which is NYC's overall hospitalization rate if one assumes a cumulative COVID-19 incidence rate of 23% [88], together with NYC's actual hospitalization count (52,333 on June 1, 2020) and population (8.34 million) [74]. Similarly, the mortality rate in the community $d(a)$ is chosen such that the overall infection fatality rate is $\sum_a f(a)\left[d(a) + h(a)d'(a)\right] \approx 1.1\%$, which is NYC's overall infection fatality rate if one considers the same cumulative incidence of 23% and the confirmed and probable cumulative death count from COVID-19 (21,607 by June 1, 2020). 108

LISTINGS

ACRONYMS

ENSO	El Niño–Southern Oscillation
SVD	singular value decomposition

RKHS	reproducing kernel Hilbert space
GP	Gaussian process
GPR	Gaussian process regression
KAF	kernel analog forecasting
KRR	kernel ridge regression
KDE	kernel density estimation
RMSE	root-mean-square error
SL	supervised learning
AR	autoregressive
DA	data assimilation
RBF	radial basis function
VB	variable-bandwidth
s. p. d.	symmetric, positive definite
*k*NN	*k*-nearest neighbors
SST	sea surface temperature
CCSM4	Community Climate System Model v. 4
ODE	ordinary differential equation
PDE	partial differential equation
delay embedding	delay embedding
p-d	pairwise distances
L-96	Lorenz '96 multiscale
L-63	Lorenz '63
HO	harmonic oscillator
TTI	testing, contact tracing, and isolation
SEIR	Susceptible, Exposed, Infected, Resistant
SEIHRD	Susceptible, Exposed, Infected, Hospitalized, Resistant, Deceased

EAKF	ensemble adjustment Kalman filter
ROC	receiver operating characteristic
TPR	true positive rate
PPF	predicted positive fraction
SBM	stochastic block model
PPV	positive predictive value
FPR	false positive rate

CHAPTER 1

INTRODUCTION

This chapter serves two purposes: (1) to provide an overview of the thesis and how chapters link together, and what the main contributions are; and (2) to present a context in which this thesis fits and how it relates to the field.

1.1 OVERVIEW OF THE THESIS

This work consists of six chapters. The current chapter, Chapter 1, is an introduction to the thesis and the broader field, and the remaining five are split into two parts, representing the results of the work done in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied and Computational Mathematics, in the Department of Computing and Mathematical Sciences at California Institute of Technology.

The first part is titled "Kernel Methods in Dynamical Systems Forecasting", and it consists of Chapters 2 to 4. Each of them has KAF as their main subject of study, a data-driven method of forecasting dynamical systems. Being a rather recent addition to the family of forecasting tools, KAF has not yet enjoyed the success and attention of some of the more well-known machine learning methods, and the first part of the thesis aims to fill in some of the gaps, exploring and extending theoretical and numerical aspects of the methodology.

Chapter 2 introduces the method, along with the necessary notation, definitions and developed theoretical background, and establishes a connection with GPR. Despite being a principally kernel method, KAF was not explicitly cast in the GPR terms before current work, and therefore this connection presents the main contribution of the chapter. Bridging two independently developed methods facilitates borrowing of the ideas and theorems from each other, enriching both. It is shown that there are three turning points (ways of performing a certain step) in the KAF algorithm which distinguish it from the GPR. These three options are referred to as modalities, and can be briefly summarized as (1) the choice of kernel, (2) bi-stochastic normalization of the kernel matrix, and (3) spectrum truncation of the eigendecomposition. Analysis of the interpolation property is also performed in this chapter.

Chapter 3 continues the study with numerical experiments. In particular, the three modalities are explored one at a time on a number of forecasting problems,

represented as time series datasets. By choosing each of the three modalities to be performed either the GPR or the KAF way, it is possible to theorize what effects they have on the forecasting skill. The datasets are obtained from two canonical low-dimensional dynamical systems — HO and L-63, and from a pre-industrial integration of the Community Climate System Model v. 4 (CCSM4), with ENSO index as the quantity of interest.

Chapter 4 proposes an extension to KAF which allows one to forecast parametrically-dependent dynamical systems. Two settings are considered: one where parameter values are explicitly known and presented as part of the time series dataset, and one where parameter values are not explicitly known, but trajectories come as delay-embedded variables. Several extensions are proposed, and then tested numerically. The dynamical systems used to generate parameter-dependent time series are again HO and L-63, with three and two parameters varied, respectively, for a total of five test problems.

The second part is titled "Data-Driven Model Augmentation" and consists of Chapters 5 and 6. These chapters use several techniques to augment dynamical systems from data, in particular, closures for subsystems and data assimilation (DA). The principal difference here is that a model of the dynamical system, possibly imperfect, is known, whereas in the first part forecasting is model-free.

Chapter 5 introduces three methods for obtaining a closure for the slow subsystem of the Lorenz '96 multiscale (L-96). The latter is a system of ordinary differential equations (ODEs) with explicit scale separation, i. e. it has interdependent slow and fast subsystems. The motivation behind seeking closures is in speeding up numerical integration of such systems. Since variables of a multiscale system depend on each other, numerical solution requires time steps small enough to resolve all scales of the dynamics, thus, the fastest mode determines the step size for the whole system. By obtaining a closure, or in other words, a function of variables of the slow subsystem that approximates behavior of the fast subsystem, it is possible to increase the step size and, hence, shorten computation times. Using GPR as a regression method, three closure variants are proposed, and numerically compared against each other, as well as against KAF.

Chapter 6 presents a model of estimating individual viral probabilities in a population. It combines a compartmental model of epidemiological evolution (akin to the Susceptible, Exposed, Infected, Resistant (SEIR) model), a graph network representing the population, and DA techniques for incorporating data. At the core of the model is a system of ODEs. It is derived from the reduced master equations of the compartmental model by introducing a data-driven closure. Vertices of the graph network represent individuals, while edges define connections between them. These connections are used to compute various epidemiological coefficients in the model. Finally, a form of Kalman filter is used as the DA method. Assuming that such a model could be implemented as a mobile app, data used in the DA loop can come from two sources: proximity

data (such as bluetooth sensors) and medical data (for example, results of the viral tests, hospitalization etc.). The model is used to estimate individual probabilities of being infected at the current moment or in the very near future, which avoids the usual complications of forecasting SEIR models. Large-scale numerical runs show that the proposed model provides a way of achieving lower death and hospitalization rates when compared to taking no measures, whilst maintaining a relatively low proportion of the quarantined population.

1.2 BROADER CONTEXT

Here we provide a few brief notes on the state of the field in general, and what place current work takes.

The field of dynamical systems modelling is phenomenally vast, spanning several disciplines in applied mathematics and natural sciences. Ranging from the classical numerical methods to the modern-day very-large-scale neural networks, there are flavors of all kinds of dynamics forecasting. On this spectrum, KAF lies in the family of model-free methods, i. e. purely data-driven. For such methods, assumptions of the underlying system generating the data can be made, but a particular form of the equations is not known. In contrast, the classical numerical modelling uses a given discretized model (whether given or discretized by a numerical method), and only takes geometry, initial and boundary conditions, and other parameters of the system as input. Finally, there are methods that combine the two approaches, making use of both data and models. In particular, among such methods are data assimilation methods and methods that use data for obtaining closures.

Among the data-driven methods, Autoregressive (AR) models represent a large and widespread class. They provide a general framework for data-driven modelling of dynamical systems, with vast literature devoted to the subject. The most basic AR model is typically written as

$$x_{n+1} = A(x_n) + \xi_n, \tag{1.1}$$

where $x_n \in \mathbb{R}^d$ is a state vector at a discrete time step n, $A\colon \mathbb{R}^d \to \mathbb{R}^d$ is an operator mapping the state space into itself, and ξ_n is noise. The operator A was first historically considered to be linear, but in general is not limited to such consideration [98]. It can also take in multiple vectors from the past, say $x_n, x_{n-1}, \ldots, x_{n-b+1}$, in which case it is said to be a b-order AR model.

The use of AR models has seen success in a number of fields, such as quantitative finance modelling [98], geophysical modelling [71, 92], biological and medical modelling [20], and has recently been extensively applied in the neural network community [15, 56, 57, 61].

One of the key differences between the framework presented by AR models and the modelling of time series using KAF is the semigroup stucture that AR

models exhibit, which is lacking in KAF case. In plain words, forecasting the model (1.1) 2 steps forward is the same as repeatedly applying it twice; for KAF this is not the case. This can be seen as a disadvantage of KAF.

However, as introduced in Chapter 2, there is rather profound theory developed for KAF, making the method stand out among other data-driven predictors. As an example, it guarantees that in the large data limit its forecast converges to the conditional expectation of the Koopman operator, applied to the response variable:

$$\mathbb{E}\left[\mathsf{U}^{\theta} F \,\middle|\, H\right] \tag{1.2}$$

Within the context of kernel methods, KAF takes a relatively niche place. The Koopman formalism has seen a lot of attention in recent years [19], similarly to kernel methods in general supervised learning. However, KAF manages to unify two theories and thus, provides a solid theoretical foundation.

The first part of this work attempts to place KAF within the family of kernel methods, study specifics of the method, and show that an extension of its application to parameter-dependent dynamical systems is possible. The second part proposes several particular closure methods derived for models of dynamical systems from time series data, and demonstrates an application of data assimilation methods to a large, nonlinear and non-local system.

Part I

KERNEL METHODS IN DYNAMICAL SYSTEMS FORECASTING

In the first part of the thesis, we focus our attention on purely data-driven forecasting of the dynamical systems, and more specifically, on the method called kernel analog forecasting. In particular, we consider some theoretical underpinnings of the method, its applications to periodic, quasi-periodic and chaotic systems, its parametric extension, and compare it to other forecasting techniques.

CHAPTER 2

THEORETICAL ANALYSIS OF KERNEL ANALOG FORECASTING

The subject of the study laid out in Chapters 2 and 3 begins its story more than half a century ago with Edward Lorenz, the famed meteorologist, who had been trained by the mathematician (and dynamical systems expert) George Birkhoff at Harvard, before switching to a PhD at MIT in the atmospheric sciences. Lorenz introduced a method for model-free prediction of the dynamical evolution [65]. Dubbed later *analog forecasting*, it essentially prescribed to search the available time series data and find the closest analog to a given initial condition from which the prediction was to be made ("closest" in this context is defined by a metric relevant for the problem at hand). Then, the forecast was formed by taking an appropriate number of discrete time steps forward along the time series, stopping at the step closest to the desired forecasting horizon (unless all steps were discrete and equal in size, and the exact time was attainable). This method, which we will call the *Lorenz method* to avoid confusion, gained a lot of attention but quickly fell out of favor for the following two reasons: it is discontinuous as a function of input, i. e. initial condition, and it did not work well with the available data (for example, if the data was too sparse). However, its simplicity and intuitive explanation remained appealing conceptually, and prompted a revisit years later, bringing the method into the new century with a more sophisticated approach in the era of massive data sets.

Kernel analog forecasting (KAF) is a method of forecasting dynamical systems, based on and aimed at unifying two methodologies: kernel operator theory and Koopman operator theory. It was first introduced in 2016 by Zhao and Giannakis [120], and has been since developed theoretically [2] and applied in numerical experiments to forecasting model problems in climate science [113]. Conceptually, KAF can be seen as a moral extension of the Lorenz method, addressing its disadvantages and supplementing it with rigorous theory.

In this chapter, we provide a view of KAF from a different angle: as one of the kernel regression family of methods. In particular, we compare it to the *Gaussian process regression* (GPR), and show that there are three modalities by which the two methods differ. We then explore each of these aspects of KAF in more detail. In the following chapter, we revisit this comparison with numerical experiments.

This chapter is organized as follows. Section 2.1 serves as a brief introduction, establishing notation that will be used throughout this and the next chapter, and giving an overview of KAF and the theory that was developed by Giannakis and collaborators, provided here for the completeness of the picture, and mostly following the work of Alexander and Giannakis [2]. Our contributions begin in Section 2.2, where we cast KAF in the form of GPR, discovering the relation between the two methods. We also draw parallels with the *kernel ridge regression* (KRR) by showing how the main formula for regression can be derived via linear least-squares. Finally, in Section 2.3 we look into the interpolation property of KAF.

2.1 INTRODUCTION

Kernel analog forecasting is a method for timeseries prediction. It aims to solve the following problem: given a timeseries of a (possibly) partially-observed dynamical system, a matching timeseries of a response variable (i. e. variable of interest), and a (possibly) unseen initial condition, predict the evolution of the response variable into the future. A few trivial observations:

- it is a discrete time-step method,
- it is a *model-free* method, that is, it uses no information of the underlying dynamical system,
- the response variable may be one of the observed states of the system, a function thereof, or a hidden state (or, again, a function of some number of them) that is not directly observed but whose evolution can, in principle, be inferred from the observed ones,
- the initial condition is given in the observables space.

2.1.1 *Notation and RKHS Theory Basics*

We now introduce notation that will allow us to formalize the method and theory behind it. We start with dynamical systems, and then move towards kernel definitions.

Loosely speaking, a dynamical system describes evolution of some state over time. Instances of such evolution are commonly referred to as *trajectories*, although they may not resemble physical intuition of the word "trajectory", depending on definitions of a system. For example, time can be continuous or discrete (naturally, giving rise to *continuous-time* and *discrete-time* systems, respectively). Continuous-time systems are more common and agree with our intuition of the surrounding world, while discrete-time systems arise with iterative processes, or as a result of discretization of a continuous-time system. Some prototypical

examples include school-level physics problems (like motion of a projectile) and iteration maps (e. g. logistic map), correspondingly.

Another branching in the richness of dynamical systems comes from the space in which trajectories lie. We call it the *phase space*, and it is, in most generality, simply a non-empty set Ω.

In many practical cases of interest, however, the phase space Ω is (1) a finite or countable set, (2) a subset of Euclidean space $\mathbb{R}^n$ (possibly, with Riemannian structure), or (3) a subset of a function space. The prototypical examples for each of these cases are (1) iterations on integers (such as the one used in the Collatz conjecture), (2) ordinary differential equations, and (3) partial differential equations. We limit the scope of our attention to the evolution on a subset $\Omega \subseteq \mathbb{R}^n$.

Definition 2.1. Let Ω be a non-empty subset of $\mathbb{R}^n$, and let $\Phi\colon \mathbb{R}^+ \times \Omega \to \Omega$ be a continuous function in both arguments. We call such function a *dynamical flow*, and use Φ^t as a shorthand notation for a fixed $t \in \mathbb{R}^+$, if and only if it satisfies two conditions:

(i) $\Phi^0(x) = x$ for any $x \in \Omega$, and

(ii) $\Phi^s \circ \Phi^t = \Phi^{s+t}$ for any $s, t \in \mathbb{R}^+$.

Condition (ii) is often referred to as the *semigroup property*, for the following reason. It is common to view a dynamical flow as a *monoid*, i. e. a semigroup with an identity element. Indeed, let the set from the monoid definition be comprised of the elements Φ^t for any $t \in \mathbb{R}^+$, and let the binary operation be the function composition $\circ$. The identity element is defined as $\Phi^0 \equiv \mathrm{Id}$ and is guaranteed to exist by condition (i). Associativity is guaranteed by condition (ii), and in fact, it also gives distributivity: $\Phi^s \circ \Phi^t = \Phi^{s+t} = \Phi^{t+s} = \Phi^t \circ \Phi^s$. The semigroup property will become relevant later for the discussion of the KAF predictor.

Definition 2.2. A *dynamical system* is a tuple (Ω, Φ) such that Φ is a dynamical flow, and Ω satisfies $\Phi^t(\Omega) \subseteq \Omega$ for all $t \in \mathbb{R}^+$.

The flow Φ can be defined on the whole Euclidean space $\mathbb{R}^n$, but very often the "interesting" behavior happens on a proper subset, and then it is enough to consider restriction of Φ on that subset. One such case is when a system has one or several attractors.

Definition 2.3. An *attractor* of a dynamical system (Ω, Φ) is a set $\Omega_0 \subset \mathbb{R}^n$ that satisfies three conditions:

(i) $\Phi^t(\Omega_0) = \Omega_0$ for all $t \in \mathbb{R}^+$;

(ii) there exists an open set $\Omega_b \supset \Omega_0$, called *basin of attraction*, with the following property: for any open set V satisfying $\Omega_0 \subset V \subseteq \Omega_b$, there exists $t \in \mathbb{R}^+$ such that $\Phi^t(x) \in V$ for any $x \in \Omega_b$;

(iii) there does not exist a proper subset $\Omega_1 \subset \Omega_0$ that satisfies the first two conditions.

The existence of an attractor from a numerical point of view means that, after initializing in the basin of attraction and integrating the system for some sufficiently long time (called *spin-up*), the evolution is constrained to a small neighbourhood of the attractor itself (or rather, its numerical, floating-point representation). Hence, we will simply use Ω further on to denote the phase space, without specifying whether it is an attractor or not because we can always assume that the dynamics will have converged to the attractor.

Definition 2.4. Let (Ω, Φ) be a dynamical system. An *observation function*

$$H\colon \Omega \to \mathcal{X}$$

is a continuous function that maps the phase space to the *observation space* $\mathcal{X} \subseteq \mathbb{R}^c$, with $c \in \mathbb{N}$.

Definition 2.5. Let (Ω, Φ) be a dynamical system. A *response variable*

$$F\colon \Omega \to \mathbb{R}$$

is any continuous, measurable and square-integrable function. We call the space of such functions the *space of responses* and denote it $\mathscr{C}(\Omega)$.

If both an observation function and a response variable are given, then we will write (Ω, Φ, H, F).

Part of the KAF methodology relies on the measure-theoretic framework, which includes such mathematical objects as a *σ-algebra* (typically denoted $\mathcal{F}$), a *probability measure* (denoted μ), a *random variable* and *Lebesgue integrals* (including expectation, variance and their conditional variants). We omit these definitions here and simply take them to be defined as usual. Similarly, we will use *spaces of square-integrable functions* without a formal definition, noting that, for a measure space $(\Omega, \mathcal{F}, \mu)$, we will denote them as $L^2(\mu)$. For σ-algebras, we will also write $\mathcal{F}(\Omega)$ assuming that some specific σ-algebra is defined on the set Ω.

We now establish the basics of the kernel theory. First, following Definition 1.1 and some derivations of Paulsen and Raghupathi [76], we introduce the reproducing kernel Hilbert space (RKHS).

Definition 2.6. Let $\mathbb{R}^{\mathcal{X}}$ denote the set of all functions from $\mathcal{X}$ to $\mathbb{R}$. A *reproducing kernel Hilbert space* $\mathcal{K}$ (on $\mathcal{X}$) is a subset of $\mathbb{R}^{\mathcal{X}}$ if it satisfies three conditions:

(i) $\mathcal{K}$ is a vector subspace of $\mathbb{R}^{\mathcal{X}}$,

(ii) $\mathcal{K}$ is endowed with an inner product $\langle \cdot, \cdot \rangle$, with respect to which $\mathcal{K}$ is a Hilbert space, and

(iii) for every $x \in \mathcal{X}$, the linear evaluation functional $E_x \colon \mathcal{K} \to \mathbb{R}$, defined by $E_x(f) \coloneqq f(x)$, is bounded.

Applying Riesz representation theorem to E_x, we conclude that for every $x \in \mathcal{X}$ there exists an element $k_x \in \mathcal{H}$ such that

$$E_x(f) = \langle f, k_x \rangle.$$

Furthermore, we can define $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ by

$$k(x, y) \coloneqq k_y(x),$$

which, in turn, allows to establish the so called *kernel trick*:

$$k(x, y) = k_y(x) = E_x(k_y) = \langle k_y, k_x \rangle.$$

The function k is called the *reproducing kernel* for $\mathcal{H}$. However, below we introduce a narrower and more useful for our purposes definition of a kernel, relying on the assumed measure-theoretic framework.

Definition 2.7. Let $(\mathcal{X}, \mathcal{F}, \mu)$ be a probability space. A function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel function* or, simply, *kernel* if it is a measurable function and is square-integrable in the joint space:

$$\int_{\mathcal{X} \times \mathcal{X}} |k(x, s)|^2 \, d\mu(x) d\mu(s) < \infty.$$

Before we introduce the next definition, we need to fix notation for a kernel matrix.

Definition 2.8. Let $\{x_1, x_2, \ldots, x_N\}$ and $\{y_1, y_2, \ldots, y_M\}$ be two collections of points in $\mathcal{X}$. We will call $K \in \mathbb{R}^{N \times M}$ a *kernel matrix* defined by:

$$K_{nm} = k(x_n, y_m),$$

and write $K = \{k(x_n, y_m)\}$ as a shorthand.

Definition 2.9. A kernel k is called *positive-definite* if for any number of distinct points $x_1, x_2, \ldots, x_N \in \mathcal{X}$, the kernel matrix $K = \{k(x_n, x_m)\}$ is positive semi-definite, i. e. for any $v \in \mathbb{R}^N$, it holds that $v^\top K v \geqslant 0$.

Definition 2.10. A kernel k is called *symmetric* if $k(x, y) = k(y, x)$ for all $x, y \in \mathcal{X}$.

Next, the following famous characterization of RKHSs allows to build a connection between two definitions of a kernel function.

Theorem 2.11 (Moore–Aronszajn)**.** *Let k be a symmetric, positive-definite kernel on $\mathcal{X}$. Then there exists a reproducing kernel Hilbert space $\mathcal{H}$ on $\mathcal{X}$ such that k is its reproducing kernel.*

The converse is also true.

Lemma 2.12. *If k is a reproducing kernel for some RKHS on $\mathcal{X}$, then it is a symmetric, positive-definite kernel on $\mathcal{X}$.*

As a consequence of these two statements, there is a one-to-one correspondence between RKHSs and symmetric, positive definite (s. p. d.) kernels, thus we will write $\mathcal{H}(k)$ when we mean "an RKHS induced by k". It also explains the importance of the s. p. d. property, and going forward, we will only consider such kernel functions.

Since we assume a probability measure μ on $\mathcal{X}$ exists, we can use it to introduce Lebesgue integration, and with it, integral operators on $L^2(\mu)$ (as we did in Definition 2.7).

Definition 2.13. Let k be an s. p. d. kernel on $\mathcal{X}$. We define *kernel operator* $\mathsf{K}\colon L^2(\mu) \to L^2(\mu)$ as follows:

$$\mathsf{K}[f](x) = \int_{\mathcal{X}} k(x,s) f(s) d\mu(s).$$

Definition 2.14. Let k be a kernel on $\mathcal{X}$, and let $X = \{x_n\}_{n=1}^N$ be a collection of points in $\mathcal{X}$, $X \subset \mathcal{X}$. The *kernel field given data* X is a mapping $k\big|_X\colon \mathcal{X} \to \mathbb{R}^N$ defined as follows:

$$k\big|_X(x) = \big(k(x,x_1), k(x,x_2), \cdots, k(x,x_N)\big)^\top.$$

Sometimes we will simply write $k(x)$ with some abuse of notation, when it is clear from the context which data set it refers to.

It is important to distinguish the two spaces: $\mathcal{H}(k)$ and $L^2(\mu)$. They consist of different objects (functions mapping $\mathcal{X}$ to $\mathbb{R}$ and *classes of equivalence* of such functions, respectively), and even though both spaces are Hilbert, their inner products are, again, different. Obviously, it is possible to define an operator ι that maps $\mathcal{H}(k)$ to $L^2(\mu)$ in the most natural way (i. e. by simply mapping to the equivalence class $[f] \in L^2(\mu)$ of a function $f \in \mathcal{H}(k)$), but the converse is not as straightforward. The situtation becomes much easier if $\mathcal{X}$ is compact and k is a *Mercer* kernel (continuous and s. p. d.). However, even in this case the inner products, in general, are not equal:

$$\langle f, g\rangle_{\mathcal{H}(k)} \neq \langle \iota f, \iota g\rangle_{L^2(\mu)}.$$

In fact, these two extra conditions (Mercer kernel and compactness of $\mathcal{X}$) allow to write a closed form of the inner product. On the one hand, continuity of the kernel leads to continuity of any $f \in \mathcal{H}(k)$, which means that $\mathcal{H}(k)$ is a "subset" of $L^2(\mu)$ (more technically, it is continuously embedded if one considers

$\iota\colon f \mapsto [f]$ as the inclusion operator). On the other hand, application of the spectral theorem gives the following representation:

$$k(x,s) = \sum_{j=1}^{\infty} \lambda_j e_j(x) e_j(s), \tag{2.1}$$

where $\lambda_j \geqslant 0$, with $e_j \in \mathcal{H}(k)$, and $\{\iota e_j\}_{j=1}^{\infty} \subset L^2(\mu)$ form an orthonormal basis in $L^2(\mu)$. This allows the usual expansion for any $[f] \in L^2(\mu)$:

$$[f] = \sum_{j=1}^{\infty} \langle [f], \iota e_j \rangle_{L^2(\mu)} \iota e_j,$$

and at the same time, using the representation property of the kernel,

$$f(x) = \langle f, k_x \rangle_{\mathcal{H}(k)},$$

and, plugging in eq. (2.1), we obtain the following:

$$f(x) = \sum_{j=1}^{\infty} \lambda_j \langle f, e_j \rangle_{\mathcal{H}(k)} e_j(x).$$

With slight abuse of notation, the formula relating the two basis expansions is

$$\langle f, e_j \rangle_{\mathcal{H}(k)} = \frac{1}{\lambda_j} \langle \iota f, \iota e_j \rangle_{L^2(\mu)}. \tag{2.2}$$

Finally, substituting eq. (2.2) into $\langle f, g \rangle_{\mathcal{H}(k)}$ three times, we arrive at

$$\begin{aligned} \langle f, g \rangle_{\mathcal{H}(k)} &= \sum_{i,j=1}^{\infty} \lambda_i \lambda_j \langle f, e_i \rangle_{\mathcal{H}(k)} \langle g, e_j \rangle_{\mathcal{H}(k)} \langle e_i, e_j \rangle_{\mathcal{H}(k)} \\ &= \sum_{i=1}^{\infty} \lambda_i \frac{\langle \iota f, \iota e_i \rangle_{L^2(\mu)}}{\lambda_i} \frac{\langle \iota g, \iota e_i \rangle_{L^2(\mu)}}{\lambda_i} = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} f_i g_i, \end{aligned}$$

where we used that $\langle e_i, e_j \rangle_{\mathcal{H}(k)} = \lambda_j^{-1} \delta_{ij}$, and denoted $f_i = \langle \iota f, \iota e_i \rangle_{L^2(\mu)}$.

2.1.2 *KAF Theory*

We start by presenting KAF as a method of predicting a response variable f, driven by a partially-observed dynamical system (Ω, Φ, H, F):

$$\begin{aligned} \omega(t) &= \Phi^t(\omega_0), \quad \omega_0 \in \Omega,\ t \in \mathbb{R}^+, \\ x(t) &= H(\omega(t)), \\ y(t) &= F(\omega(t)). \end{aligned}$$

For the ease of notation, we will additionally use $x^t \colon \Omega \to \mathcal{X}$ and $y^t \colon \Omega \to \mathbb{R}$:

$$x^t = H \circ \Phi^t, \qquad y^t = F \circ \Phi^t.$$

Later, we will reformulate KAF as a method for general *supervised learning* (SL), but in this section we introduce it the way it was originally developed. Definitions and theorems used in this section follow Alexander and Giannakis [2] unless otherwise noted.

Definition 2.15. The *Koopman operator* $\mathsf{U} \colon \mathbb{R}^+ \times \mathscr{C}(\Omega) \to \mathscr{C}(\Omega)$ maps the space of responses into itself, for every $t \in \mathbb{R}^+$, and is defined through the following relation:

$$\mathsf{U}(t, F) \coloneqq F \circ \Phi^t, \quad \text{for any } t \in \mathbb{R}^+ \text{ and any } F \in \mathscr{C}(\Omega).$$

The common shorthand notation is U^t, and since it is a linear operator by definition, we will simply write $\mathsf{U}^t F$.

The idea behind the Koopman operator is to provide an alternative way of "changing coordinates": instead of staying in a finite-dimensional phase space (or a space of changed coordinates) with nonlinear evolution, Koopman operator lifts the dynamics into an infinite-dimensional phase space with linear evolution. With slight abuse of notation, we will also use the same notation for the Koopman operator defined on a space of μ-square-integrable response variables:

$$L^2(\Omega, \mathbb{R}; \mu) = \left\{ F \colon \Omega \to \mathbb{R} \;\middle|\; \int_\Omega |F(\omega)|^2 d\mu(\omega) < \infty \right\}.$$

This discrepancy should not be of serious importance since in the case of compact phase space Ω, the space of responses $\mathscr{C}(\Omega)$ is a "subset" of $L^2(\Omega, \mathbb{R}; \mu)$, in the same sense as $\mathscr{H}(k)$ was a "subset" of $L^2(\mu)$ in the previous section.

Definition 2.16. For a given measurable observation function $H \colon \Omega \to \mathcal{X}$, the *pushforward measure* (or simply, *pushforward*) $\mu_H \colon \mathscr{F}(\mathcal{X}) \to \mathbb{R}^+$ is a measure on the observation space $\mathcal{X}$ defined by

$$\mu_H(A) \coloneqq \mu\left(H^{-1}(A)\right), \quad \text{for every } A \in \mathscr{F}(\mathcal{X}).$$

Definition 2.17. A *target function* $\zeta \colon \mathbb{R}^+ \times \mathcal{X} \to \mathbb{R}$ is a function that, for every $\theta \in \mathbb{R}^+$, approximates evolution of the response variable from a given observed state:

$$\zeta(\theta, x(t)) \approx y(t + \theta), \tag{2.3}$$

and as a shorthand, we will write $\zeta^\theta \circ x^t \approx y^{t+\theta}$. In addition, we require that ζ^θ is a μ_H-square-integrable function.

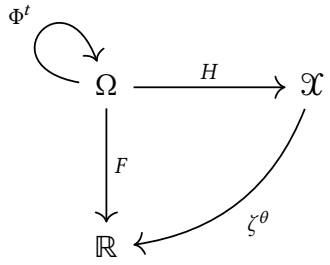


Figure 2.1: Commutative diagram of the spaces in the dynamical systems perspective.

Here we deliberately leave out any specifics as to what is meant by "approximating" as it will depend on the appropriate metric, tolerance level etc. It is important to note, however, that in general, such approximation might not be possible. As a simple illustratory example, in the system

$$\begin{aligned}\omega_1(t) &= a, \quad a \in \mathbb{R},\\ \omega_2(t) &= bt, \quad b \in \mathbb{R},\\ x(t) &= \omega_1(t),\\ y(t) &= \omega_2(t),\end{aligned}$$

the residual $\left\|\zeta^\theta \circ x^t - y^{t+\theta}\right\|$ can be made arbitrarily large.

The example above shows what exactly could go wrong, and, more importantly, that when it does go wrong, there is nothing we can hope for in regards to reaching our forecasting goal. Clearly, if some crucial information is lost by the observation function H (like the time t in this case), then it is impossible to minimize the residual. However, the next natural question to ask is: what can be said if only *some* information is lost?

To answer that, we need a more granular approach, and the measure-theoretic perspective comes to rescue. Since we assume that all considered spaces (namely, Ω, $\mathcal{X}$ and $\mathbb{R}$, see Figure 2.1) have a measure, one of the natural approaches is to consider the residual in the L^2-space of functions (rather, their classes of equivalence) that map Ω to $\mathbb{R}$:

$$\mathscr{E}_\theta[\zeta] = \left\|\zeta^\theta \circ x^t - y^{t+\theta}\right\|^2_{L^2(\mu)}. \tag{2.4}$$

Obviously, for this to be defined, both functions $\zeta^\theta \circ x^t$ and $y^{t+\theta}$ must be measurable functions w. r. t. the σ-algebra $\mathcal{F}$ on Ω. In both cases, this condition is always true by assumptions.

However, the function H defines a sub-σ-algebra $\mathcal{F}(H)$ on Ω which is, in general, coarser than the σ-algebra $\mathcal{F}$. Those subsets of Ω on which H takes constant values are indistinguishable for any function defined on the range of

H, i. e. the observation space $\mathcal{X}$. This means that the function $\zeta^\theta \circ H$ is only $\mathcal{F}(H)$-measurable.

For brevity, we omit some of the theoretical details of the construction here. In plain words, let $L^2(\mathcal{F}(H), \mu)$ be the subspace of the $L^2(\mu)$ space of functions which consists of the $\mathcal{F}(H)$-measurable functions. This subspace is the search space, and by considering the orthogonal projection of $\mathsf{U}^\theta F \in L^2(\mu)$ onto $L^2(\mathcal{F}(H), \mu)$, it can be shown that there exists a unique function $R^\theta \circ H$ such that

$$\left\langle \varphi, \mathsf{U}^\theta F \right\rangle_{L^2(\mu)} = \left\langle \varphi, R^\theta \circ H \right\rangle_{L^2(\mu)}, \tag{2.5}$$

for any $\varphi \in L^2(\mathcal{F}(H), \mu)$. Since this is the orthogonal projection, it also delivers the minimum in the $\mathscr{C}_\theta[\zeta]$ functional. Finally, it also has the interpretation of the conditional expectation:

$$R^\theta \circ H = \mathbb{E}\left[\mathsf{U}^\theta F \,\middle|\, H \right], \tag{2.6}$$

where we use common shorthand notation $\mathbb{E}[\cdot|\, H]$ instead of $\mathbb{E}[\cdot|\, \mathcal{F}(H)]$.

Definition 2.18. The *regression function* at lead time θ is the unique function $R^\theta \colon \mathcal{X} \to \mathbb{R}$ which stems from the orthogonal projection of the image of the Koopman operator applied to the response variable:

$$R^\theta \circ H = \mathbb{E}\left[\mathsf{U}^\theta F \,\middle|\, H \right].$$

We have now implicitly defined the best possible solution to the forecasting problem: it is the function $R \colon \mathbb{R}^+ \times \mathcal{X}$, of course, with the usual remark that $R \in L^2(\mathcal{F}(\mathcal{X}), \mu_H)$ is an equivalence class and there needs to be a way to "extract" a function out of it, which we will revisit later. However, theoretical existence and uniqueness of such function alone does not specify how it can be found in practice. Taking the optimization approach requires defining a functional and a search space.

Note that having identified the source of error, we can now decompose it into two parts: one which is *intrinsic* to the problem and depends solely on the dynamical system, and one which is defined by our approximation to the sought ideal solution (we call it the *excess* error):

$$\mathscr{C}_\theta[\zeta] = \mathscr{e}_\theta^{(i)} + \mathscr{e}_\theta[\zeta],$$

with

$$\mathscr{e}_\theta^{(i)} = \left\| R^\theta \circ H - \mathsf{U}^\theta F \right\|^2_{L^2(\mu)}, \tag{intrinsic}$$

$$\mathscr{e}_\theta[\zeta] = \left\| \zeta^\theta - R^\theta \right\|^2_{L^2(\mu_H)}. \tag{excess}$$

Clearly, minimizing $\mathscr{e}_\theta[\zeta]$ is equivalent to minimizing the whole error.

Definition 2.19. The *hypothesis space* $\mathscr{H} \subseteq L^2(\mathscr{F}(\mathcal{X}), \mu_H)$ is a closed and convex subset over which the excess error $e_\theta[\zeta]$ is minimized.

Remark 2.20. As usual, for ease of notation, we identify a *set* of L^2 functions with its image under the ι operator that maps into the L^2 *space*. An important requirement here is that ι must be injective, i. e. each function in the preimage of $\mathscr{H}$ is mapped to its own equivalent class. This, however, is a mild constraint because usually the hypothesis space contains at functions that are at least continuous (and in our case, it will be an RKHS).

We are now ready to give the final definition before moving on to the final-dimensional setting where we will work with data. Since $\mathscr{H}$ is a closed and convex subset of a Hilbert space, there exists an orthogonal projection $\mathsf{P}_{\mathscr{H}} : L^2(\mu_H) \to \mathscr{H}$. Due to orthogonality, we can further decompose the excess error into two terms:

$$e_\theta[\zeta] = \left\| \mathsf{P}_{\mathscr{H}} R^\theta - \zeta^\theta \right\|^2_{L^2(\mu_H)} + \left\| \left(\mathrm{Id} - \mathsf{P}_{\mathscr{H}}\right) R^\theta \right\|^2_{L^2(\mu_H)}.$$

And just as before, since the second term does not depend on ζ, minimizing the excess error is equivalent to minimizing the first term in the above equation.

Definition 2.21. The *ideal target function* $Z : \mathbb{R}^+ \times \mathcal{X} \to \mathbb{R}$ is the unique element of the hypothesis space $\mathscr{H}$, and is defined as the minimizer of the excess error:

$$Z^\theta = \mathsf{P}_{\mathscr{H}} R^\theta.$$

There are three things to note here. First, note that the ideal target function depends on the hypothesis space, but for brevity we omit this dependence from notation. Second, the existence of this function follows from the fact that the ι opeartor is injective, as discussed in Remark 2.20. Finally, the uniqueness follows from the orthogonality of the projection $\mathsf{P}_{\mathscr{H}}$, which, in turn, is possible because the hypothesis space is closed and convex.

2.1.2.1 *Finite-dimensional Setting*

Moving to more concrete objects, kernel analog forecasting employs an RKHS as the hypothesis space, or rather, a finite-dimensional subspace thereof. As mentioned in Section 2.1.1, the kernel operator $\mathsf{K} : L^2(\mu_H) \to L^2(\mu_H)$ is compact and self-adjoint, thus, invoking the spectral theorem, we obtain the eigendecomposition:

$$\mathsf{K} v_i = \lambda_i v_i,$$

with $v_i \in L^2(\mu_H)$ and $\lambda_i \in \mathbb{R}$. Moreover, eigenfunctions form an orthonormal basis in $L^2(\mu_H)$, and because K is trace-class and positive (taking into the view Definition 2.7 of a kernel k and requiring it to be a Mercer kernel, see Theorems

4.26 and 4.27 of Steinwart and Christmann [102]), its eigenvalues can be sorted in descending order $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0$. Define

$$u_i = \lambda^{-1/2} \iota^* v_i,$$

where $\iota^* \colon L^2(\mu_H) \to \mathcal{K}$ is the adjoint operator. Using eq. (2.2), it is easy to show that u_i form an orthonormal set in $\mathcal{K}$, leading to the following definition.

Definition 2.22. An ℓ-dimensional *KAF hypothesis space* $\mathcal{H}_\ell \subseteq \mathcal{K}(k)$ is defined as $\mathcal{H}_\ell = \operatorname{span}\{u_i\}_{i=1}^{\ell}$.

Given the existence and explicit form of the orthonormal basis of $\mathcal{H}_\ell$, we can now write out expansion of $R^\theta \in L^2(\mu_H)$:

$$R^\theta = \sum_{i=1}^{\infty} \alpha_i(\theta) v_i,$$

with the following expression for the coefficients:

$$\alpha_i(\theta) = \left\langle v_i, R^\theta \right\rangle_{L^2(\mu_H)} = \left\langle v_i \circ H, R^\theta \circ H \right\rangle_{L^2(\mu)} = \left\langle v_i \circ H, \mathsf{U}^\theta F \right\rangle_{L^2(\mu)}.$$

Here in the last equality we used eq. (2.5), and before that we used the fact that the mapping $v \mapsto v \circ H$ is an isometric isomorphism between $L^2(\mu_H)$ and $L^2(\mu)$ (this follows directly from the definition of the pushforward μ_H).

From this derivation we can write out the closed form of the ideal target function Z:

$$Z^\theta = \sum_{i=1}^{\ell} \alpha_i(\theta) \lambda_i^{-1/2} u_i.$$

Definition 2.23. The *variable-bandwidth* kernel $k_{\mathrm{vb}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined as

$$k_{\mathrm{vb}}(x, y) = \exp\left(-\frac{\|x - y\|^2}{\delta\, r(x)\, r(y)}\right),$$

where $\delta \in \mathbb{R}^+$ is a scale parameter, and $r \colon \mathcal{X} \to \mathbb{R}$ is a surrogate for a density function of the data.

In practice, both δ and r are treated as parameters, and so are estimated from the data points. We omit the details of that algorithm here, simply noting that it relies on the *kernel density estimation* (KDE) techniques (see Berry and Harlim [13] and Giannakis [39]). Importantly, this procedure only uses domain data. This is in contrast to the usual tuning of some parametrized kernel: more often, tuning involves some form of an optimization algorithm with cross-validation, and thus requires both domain and regression data points.

Definition 2.24. The *(symmetric) bi-stochastic normalization* of an s. p. d. kernel k is a procedure that produces another s. p. d. kernel $g\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the Markov property:

$$\int_{\mathcal{X}} g(x,s)\, d\mu_H(s) = 1, \quad \text{for all } x \in \mathcal{X}.$$

The kernel g is explicitly defined through the following relations:

$$g(x,y) = \int_{\mathcal{X}} \frac{k(x,s)\, k(s,y)}{d(x)\, q(s)\, d(y)}\, d\mu_H(s), \tag{2.7a}$$

with

$$d(x) = \int_{\mathcal{X}} k(x,s)\, d\mu_H(s), \qquad q(x) = \int_{\mathcal{X}} \frac{k(x,s)}{d(s)}\, d\mu_H(s). \tag{2.7b}$$

2.2 GP INTERPRETATION OF KAF

In this section, we look into how part of the KAF methodology can be re-interpreted in terms of the Gaussian process (GP) regression.

2.2.1 *Problem Formulation*

We first introduce a general SL problem. Assume that data χ is given in the form of pairs:

$$\chi = \{x_n, y_n = f(x_n)\}_{n=1}^N, \quad \text{with } x_n \neq x_m \text{ for } n \neq m, \tag{2.8}$$

where $x_n \in S$, $y_n \in \mathbb{R}$, and $f\colon S \to \mathbb{R}$ is the unknown function. Our goal is to approximate f on a compact set $S \subset \mathcal{X}$ provided that data points x_n are sampled from the set S densely enough to recover the sampling measure. To express regression data (2.8) succinctly, it will be useful to introduce the following notation:

$$f(X) = y, \tag{2.9}$$

where $y = (y_1, \ldots, y_N)^\top$, $X = \{x_n\}_{n=1}^N$ agrees with notation of Definition 2.14, and $f(X) = (f(x_1), \ldots, f(x_N))^\top$.

A particular case of the problem above is in the context of dynamical systems, briefly introduced in Section 2.1.1 as the original setup for KAF. To establish a link between two notations, let (Ω, Φ, H, F) be a dynamical system, and let the observation space be the same as the phase space, $\mathcal{X} \equiv \Omega$, with $H \equiv \mathrm{Id}_\Omega$. Fixing the time $t = \theta$, we can define

$$f := F \circ \Phi^\theta, \tag{2.10}$$

and, thereby, transform the problem of forecasting $\mathsf{U}^t \circ F(s)$ into the SL problem of approximating $f(s)$. To preserve the generality of the dynamical systems setting, i. e. when $\mathcal{X} \neq \Omega$, some additional work is required. For example, we might consider function $f(x;\xi)$ with a latent variable ξ to account for the information "lost" by the observation function H. Here we avoid such complications and only work with the simpler version.

2.2.2 *Overview of the KAF Methodology*

To highlight the GPR–KAF connection, we summarize the practical details outlined in Section 2.1.2, and view KAF as a two-step algorithm.

I Using X data only, construct a data-driven s. p. d. kernel $g(\cdot,\cdot)$, and compute the first ℓ eigenpairs of the kernel operator $\mathsf{G}\colon L^2(\nu) \to L^2(\nu)$. Recall that $\mathsf{G} = \iota\iota^*$, with $\iota\colon f \mapsto [f]$, and its adjoint ι^* maps L^2 to $\mathcal{H}(g)$, the RKHS induced by kernel g. The operator G is defined in a standard way:

$$\mathsf{G} f = \int_S g(\cdot, s)\, f(s)\, \nu(ds), \tag{2.11}$$

with ν being the sampling measure.

II Even though G acts on an infinite-dimensional space $L^2(\nu)$, in practice only the kernel matrix G with entries $G_{nm} = g(x_n, x_m)$ can be computed; thus, a form of Nyström extension is used to construct basis functions $\{\psi_i\}_{i=1}^{\ell} \subset \mathcal{H}(g)$ from eigenvectors u_i of G. KAF approximation is then formed by projecting the function of interest f onto these basis functions.

Analysis of Step II comprises Section 2.2.3, while Step I is discussed in more detail in Section 2.2.4. Here we only note that Step I itself consists of two parts:

I(a) assuming that S is a compact manifold, the k-nearest neighbors (kNN) algorithm and kernel density estimation (KDE) with the radial basis function (RBF) kernel are used to estimate (1) the manifold dimension of S, (2) the sampling density, and (3) the scale parameter, all from domain data X only; these quantities are plugged into the formula for the variable-bandwidth (VB) kernel k_{vb};

I(b) symmetric bi-stochastic normalization is applied to k_{vb}, resulting in the data-driven kernel g; this normalization guarantees that (1) the spectrum of G (and, by extension, G) lies in $[0, 1]$, (2) its top eigenvalue is equal to 1, and (3) the top eigenvector is constant (i. e. G is a constant-preserving, or averaging, operator).

In matrix notation, the normalization from part (b) is written as follows:

$$G = D^{-1} K_{\mathrm{vb}} Q^{-1} K_{\mathrm{vb}}^{\top} D^{-1}, \tag{2.12a}$$

where

$$D = \mathrm{diag}(d), \quad d = K_{\mathrm{vb}} \mathbb{1}, \quad Q = \mathrm{diag}(q), \quad q = K_{\mathrm{vb}} D^{-1} \mathbb{1}, \tag{2.12b}$$

and $K_{\mathrm{vb}} = \{k_{\mathrm{vb}}(x_n, x_m)\}$ is the unnormalized kernel matrix. As the VB kernel is symmetric, so is the kernel matrix K_{vb}, thus we can drop its transpose in eq. (2.12a). Formulas (2.12) are in direct correspondence with the continuous version of the normalization in eq. (2.7).

2.2.3 *Second Step of KAF is Truncated-Spectrum GP Regression*

We now prove that Step II of the KAF algorithm can be viewed as an approximation of the classic GP regression. More precisely, this approximation is the truncated-spectrum GP regression (introduced later in Definition 2.27). First, we recall the basic definitions and formulas of the GP theory.

Let $(W(s))_{s \in S}$ be a GP with zero mean and s. p. d. an covariance function g. Given data (2.8), we assume $\{W(x_n) = y_n\}_{n=1}^{N}$. Recall notation (2.9), the formula for conditional expectation is then:

$$\begin{aligned} \mathbb{E}\left[W(s) \,\middle|\, W(X) = y \right] &= \sum_{n,m=1}^{N} y_n \, \Theta_{nm} \, g(x, x_m) = y^{\top} \Theta \, g(s) \\ &= g(s)^{\top} \Theta \, y, \end{aligned} \tag{2.13}$$

where $\Theta := G^{-1}$ is the precision matrix, and $g(s) = g\big|_X(s)$ is the kernel field given data X (Definition 2.14).

By assumption, g is symmetric, which means that G is also symmetric. Invoking the finite-dimensional spectral theorem, let the eigendecomposition of the kernel matrix G be denoted by

$$G = U \Lambda U^{\top}, \tag{2.14}$$

where $U \in \mathbb{R}^{N \times N}$ is orthogonal, with eigenvectors u_i as columns, and $\Lambda \in \mathbb{R}^{N \times N}$ is a diagonal matrix with eigenvalues λ_n, ordered from highest to lowest. Plugging this into eq. (2.13) yields

$$\mathbb{E}\left[W(s) \,\middle|\, W(X) = y \right] = g(s)^{\top} U \Lambda^{-1} U^{\top} y. \tag{2.15}$$

Before we move to proving the main result of the chapter, several definitions need to be introduced.

Definition 2.25. Let $A \in \mathbb{R}^{N\times N}$ be a matrix of rank r, and let $A = U\Sigma V^\top$ be its singular value decomposition (SVD). The *Moore–Penrose pseudoinverse* $A^\dagger$ is defined as

$$A^\dagger := U\Sigma^\dagger V^\top,$$

where

$$\Sigma^\dagger = \operatorname{diag}\left(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0\right).$$

Definition 2.26. Let $A \in \mathbb{R}^{N\times N}$ be a matrix of rank r, and let $A = U\Sigma V^\top$ be its SVD. The *γ-truncation* of A, for $\gamma \leqslant r$, $\gamma \in \mathbb{N}$, is called the following matrix:

$$A_{[:\gamma]} := U\Sigma_{[:\gamma]}V^\top,$$

where

$$\Sigma_{[:\gamma]} = \operatorname{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_\gamma, 0, \ldots, 0\right).$$

Similarly, the *γ-tail* of A is defined as

$$A_{[\gamma:]} := A - A_{[:\gamma]} = U\left(\Sigma - \Sigma_{[:\gamma]}\right)V^\top,$$

so it contains only the remaining $r - \gamma$ singular values.

Thus, for example, in the case of a symmetric matrix G and its eigendecomposition (2.14), the matrices $\Lambda_{[:\ell]}$ and $\Lambda_{[\ell:]}$ contain upper ℓ and lower $N - \ell$ eigenvalues, respectively, sorted in descending order:

$$\Lambda_{[:\ell]} = \begin{pmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_\ell & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}, \quad \Lambda_{[\ell:]} = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \lambda_{\ell+1} & & \\ & & & & \ddots & \\ & & & & & \lambda_N \end{pmatrix}$$

Note that, in general, operations of truncation and taking pseudo-inverse are not commutative:

$$\begin{aligned} \left(\Sigma^\dagger\right)_{[:\ell]} &= \left(\operatorname{diag}\left(\sigma_r^{-1}, \ldots, \sigma_1^{-1}, 0, \ldots, 0\right)\right)_{[:\ell]} \\ &= \operatorname{diag}\left(\sigma_r^{-1}, \ldots, \sigma_{r-\ell}^{-1}, 0, \ldots, 0\right) \\ &\neq \operatorname{diag}\left(\sigma_1^{-1}, \ldots, \sigma_\ell^{-1}, 0, \ldots, 0\right) \\ &= \left(\operatorname{diag}\left(\sigma_1, \ldots, \sigma_\ell, 0, \ldots, 0\right)\right)^\dagger = \left(\Sigma_{[:\ell]}\right)^\dagger. \end{aligned}$$

This is due to a simple fact that sorting singular values in descending order, and taking their reciprocals are also not commutative. Therefore, unless specified, we will always assume that truncation takes precedence:

$$A^\dagger_{[:\ell]} \equiv \left(A_{[:\ell]}\right)^\dagger.$$

Definition 2.27. Let $\big(W(s)\big)_{s\in S}$ be a GP with zero mean and an s. p. d. covariance function g. Given regression data $\chi = (X, y)$, the *truncated-spectrum GP regression* is a map $W\big|_\chi \colon S \times \{1, \dots, N\} \to \mathbb{R}$ defined as follows:

$$W\big|_\chi(s; \ell) \coloneqq g(s)^\top G^\dagger_{[:\ell]}\, y = g(s)^\top U\, \Lambda^\dagger_{[:\ell]}\, U^\top y.$$

Comparing this definition with formula (2.15) reveals that the posterior mean of $W(s)$ is recovered when $\ell = N$:

$$W\big|_\chi(s; N) \equiv \mathbb{E}\Big[W(s) \;\Big|\; W(X) = y\Big].$$

Finally, let $p \coloneqq \lfloor \frac{\theta}{\Delta t} \rfloor$, i. e. an integer number of steps, rounded down. In what follows, we identify p and θ because we only consider a fixed timestep Δt. Similarly, we will write the Koopman operator U^p to mean $\mathsf{U}^{p\Delta t}$.

Lemma 2.28. *Let $\{g_n(\cdot)\}_{n=1}^N$ be a collection of functions defined on S and mapping into $\mathbb{R}$, and let U and $A = \operatorname{diag}(a_1, \dots, a_N)$ be two matrices on $\mathbb{R}^{N\times N}$. Then Nyström extensions $\{\psi_i(\cdot)\}_{i=1}^N$ defined by the formulas*

$$\psi_i(s) = a_i \sum_{n=1}^N U_{ni}\, g_n(s)$$

can be written in matrix notation:

$$\psi(s) = AU^\top g(s),$$

where we used vectors $\psi(s)$ and $g(s)$:

$$\psi(s) = \Big(\psi_1(s), \dots, \psi_N(s)\Big)^\top \quad \text{and} \qquad g(s) = \Big(g_1(s), \dots, g_N(s)\Big)^\top,$$

Proof. The proof is a straightforward application of the rules of matrix multiplication. Denote

$$b_i(s) \coloneqq \sum_{n=1}^N U_{ni}\, g_n(s),$$

then vector $b(s)$ is simply a product of $U^\top$ and $g(s)$ by definition (transpose comes from the swapped indices under the sum). Noting that the expression of the form $\psi_i(s) = a_i b_i(s)$ defines vector $\psi(s) = Ab(s)$ concludes the proof. □

Lemma 2.29. *Let $\{\psi_i(\cdot)\}_{i=1}^N$ be a collection of functions defined on S and mapping into $\mathbb{R}$, let $y \in \mathbb{R}^N$, and let U and $A = \operatorname{diag}(a_1, \dots, a_N)$ be two matrices on $\mathbb{R}^{N\times N}$. Then, for any $\ell = 1, \dots, N$, a function $z \colon S \to \mathbb{R}$ defined by the formula*

$$z(s) = \sum_{n=1}^N \sum_{i=1}^\ell y_n\, U_{ni}\, a_i\, \psi_i(s),$$

can be written in matrix notation:

$$z(s) = y^\top U A_{[:\ell]}\, \psi(s).$$

Proof. Define $b(s) \in \mathbb{R}^N$ with

$$b_n(s) \coloneqq \sum_{i=1}^{\ell} U_{ni}\, a_i\, \psi_i(s),$$

then, clearly, $z(s) = \sum_{n=1}^{N} y_n b_n(s)$ can be expressed as the inner product of two vectors, that is $z(s) = y^\top b(s)$.

Now define $c(s) \in \mathbb{R}^N$ with

$$c_i(s) \coloneqq \begin{cases} a_i\, \psi_i(s), & \text{for } i \leqslant \ell, \\ 0, & \text{for } \ell < i \leqslant N, \end{cases}$$

then the upper limit in definition of $b_n(s)$ can be changed to N instead:

$$b_n(s) = \sum_{i=1}^{N} U_{ni}\, c_i(s),$$

and by definition of matrix multiplication, vector $b(s)$ is simply equal to the product $Uc(s)$. Finally, using the truncation notation, for vector $c(s)$ we write:

$$c(s) = \begin{pmatrix} a_1 & & & & & \\ & \ddots & & & & \\ & & a_\ell & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \begin{pmatrix} \psi_1(s) \\ \psi_2(s) \\ \vdots \\ \psi_N(s) \end{pmatrix} = A_{[:\ell]}\, \psi(s).$$

Combining expressions for $z(s)$, $b(s)$ and $c(s)$, we obtain the needed result:

$$z(s) = y^\top b(s) = y^\top U c(s) = y^\top U A_{[:\ell]}\, \psi(s).$$

□

Theorem 2.30. *Let g be an s.p.d. kernel, and let $\chi = (X, y)$ be the regression data. Then, the KAF predictor $Z^p : S \to \mathbb{R}$ from Step II is equivalent to the truncated-spectrum GP regression $W\big|_\chi$.*

Proof. Recalling expressions from Algorithms 2 and 3 in [21], the KAF formula is written as follows:

$$Z^p(s) = \sum_{n=1}^{N} \sum_{i=1}^{\ell} \frac{\psi_i(s)\, u_i(x_n)}{\lambda_i^{1/2}}\, y_n^{(p)}, \tag{2.16}$$

where $y_n^{(p)} = \mathsf{U}^p F(x_n)$, $u_i(x_n) = U_{ni}$, $\lambda_i = \Lambda_{ii}$, and

$$\psi_i(s) = \lambda_i^{-1/2} \sum_{n=1}^{N} g(s, x_n) u_i(x_n). \tag{2.17}$$

Note that if in formula (2.17) we had λ_i^{-1} instead we would arrive at the standard Nyström extension; however, the benefit of normalizing with the square root is that functions given by eq. (2.17) form an orthonormal basis in the RKHS $\mathcal{H}(g)$ [2].

We can now cast the KAF formula (2.16) in matrix notation. Defining

$$y^{(p)} := \left(y_1^{(p)}, \ldots, y_N^{(p)}\right)^\top \quad \text{and} \quad \psi(s) := \left(\psi_1(s), \ldots, \psi_N(s)\right)^\top,$$

and applying Lemmas 2.28 and 2.29 to eqs. (2.16) and (2.17), we arrive at the following:

$$\begin{aligned} Z^p(s) &= \left(y^{(p)}\right)^\top U \left(\Lambda_{[:\ell]}^{1/2}\right)^\dagger \psi(s) \\ &= \left(y^{(p)}\right)^\top U \left(\Lambda_{[:\ell]}^{1/2}\right)^\dagger \left(\Lambda^{-1/2} U^\top g(s)\right) \\ &= \left(y^{(p)}\right)^\top U \Lambda_{[:\ell]}^\dagger U^\top g(s). \end{aligned} \tag{2.18}$$

Here we used the fact that for a product of two diagonal matrices, if one of them has zeros on the diagonal then the corresponding values of the other matrix can be set to any value (for example, also zero) without changing the product.

Since $Z^p(s) \in \mathbb{R}$, it is equal to its own transpose:

$$Z^p(s) = g(s)^\top U \, \Lambda_{[:\ell]}^\dagger \, U^\top y^{(p)} = g(s)^\top G_{[:\ell]}^\dagger \, y^{(p)}, \tag{2.19}$$

and noting that $y_n^{(p)} = \mathsf{U}^p F(x_n) = f(x_n) = y_n$ concludes the proof. □

Corollary 2.31. *If $\ell = N$, then the KAF formula recovers standard GP regression.*

Proof. Follows trivially from the observation after Definition 2.27. □

Remark 2.32. By assumption, kernel g is s. p. d., which allows one to talk about the "first" eigen-pairs, sorted by eigenvalues from largest to smallest. This has the usual interpretation of the "amount" of energy contained in each mode, and ℓ-truncation, therefore, cuts the tail of the spectrum.

2.2.4 *Three Modalities*

We continue our analysis of the KAF predictor from the point of view of the GP regression. We have now seen that, given an arbitrary s. p. d. kernel g, performing Step II is equivalent to doing the truncated-spectrum GP regression. Here we interpret Steps I(a) and I(b), and provide intuition behind each of the three major differences between GPR and KAF, which we call modalities.

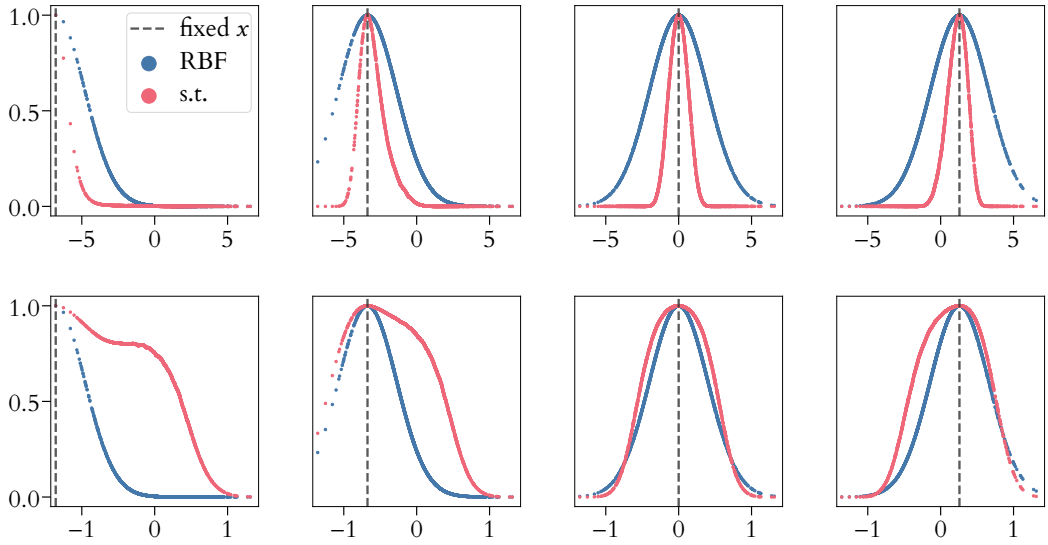


Figure 2.2: Examples of the self-tuning kernels.

Blue and red depict the RBF and self-tuning kernels, respectively. Data points are sampled from a normal distribution $\mathcal{N}(0, \sigma)$, with $\sigma = 2$ and 0.4 for top and bottom rows, respectively.

2.2.4.1 *The Variable-Bandwidth Kernel*

Step I(a) constructs a specific data-adopted kernel k_{vb}, called the variable-bandwidth kernel (see Definition 2.23). This construction involves two alike optimization algorithms, which tune the parameters δ and $r(s)$ using the data points. Most notably, though, only domain data points $X = \{x_n\}_{n=1}^N$ are used. This means that, unlike most conventional GPR tuning methods, it does not involve loss function minimization or cross-correlation techniques.

From a theoretical point of view, the VB kernel aims to adjust for the sampling density irregularities. In this sense, it is similar to the *self-tuning* kernel of Zelnik–Manor and Perona [119] (here given in discrete form):

$$\{K_{\text{st}}\}_{nm} = \exp\left(-\frac{\|x_n - x_m\|^2}{\sigma_n\,\sigma_m}\right), \qquad \text{where } \sigma_n = \|x_n - x_{J(n)}\|, \tag{2.20}$$

and $J(n)$ is the κ-th nearest neighbor of point x_n, $\kappa \in \{1, \ldots, N-1\}$. This still leaves the parameter κ to be determined (and, in fact, a similar parameter exists in the VB kernel tuning algorithm), but it has been observed in practice that perfomance is not too sensitive w. r. t. κ, and empirically it is often set to some fraction of the number of points (for example, $\kappa = \lceil 0.1N \rceil$ is common).

The difference between K_{st} and K_{vb} kernels is in the level of sophistication. It can be argued that for certain problems, they will produce similar results. Examples of both kernels are depicted in Figure 2.2.

Comparing to a standard GPR kernel like RBF, the benefits and drawbacks of using the VB kernel are summarized below:

+ for problems with significant variation in the sampling density, the VB kernel adjusts and performs more accurately;
+ it also provides computational speed-ups in training because no costly $\mathcal{O}(N^3)$ matrix inversions are needed during tuning, since the loss function is not computed;
+ moreover, if two (or more) functions, say, $f_1(X) = y^{(1)}$ and $f_2(X) = y^{(2)}$, are to be regressed, and the X data is unchanged for both, then the tuning only happens once;
- however, some information about the approximated function is lost, for example, its smoothness, which could lead to possibly incorrect scale parameter estimation.

Note that, while the third positive might seem arbitrary in the general SL setting, it becomes important for forecasting of dynamical systems. Indeed, recalling that each discrete time horizon p comes with a unique regression function $f_p(X) = \mathsf{U}^p F(X) = y^{(p)}$, tuning the kernel only once per dataset X saves a lot of computational time.

2.2.4.2 *Bi-stochastic Normalization*

Second deviation from the common GPR usage is the normalization of the kernel. Step I(b) takes *any* s. p. d. kernel and outputs a normalized one. In practice, the matrix form (2.12) is used, without resorting to formulas from Definition 2.24, but the kernel function $g(\cdot, \cdot)$ is always implicitly defined, and can be, in principle, restored (via Nyström extension, for example). The aim of the normalization is two-fold, as summarized below.

SPECTRUM NORMALIZATION:
: The spectrum of the resulting kernel operator G — and, by extension, that of the kernel matrix G — is normalized to $[0, 1]$. Recall that starting from an s. p. d. kernel k, the bi-stochastic normalization produces an s. p. d. kernel g, which allows to sort the eigenvalues in descending order, with a guaranteed first eigenvalue equal to one:

$$1 = \lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0.$$

Moreover, the eigenfunction that corresponds to λ_1 is constant:

$$\mathsf{G}\,\mathbb{1}_S(x) = \mathbb{1}_S(x).$$

Having a constant function in the hypothesis space, in turn, provides a necessary tool for proving convergence results with $t \to \infty$. In plain words, it means that in the worst case KAF recovers the time-independent conditional expectation $\mathbb{E}[F|H]$. At the same time, spectrum lying in $[0, 1]$ allows for more robust spectrum truncation, in view of Remark 2.32.

DENSITY CORRECTION:
The bi-stochastic normalization originates from Coifman and Hirn [26], with a construction similar to the diffusion maps algorithm [27]. This suggests that the normalization is able to correct variability in sampling density.

Note that both VB kernel and normalization aim to adjust for the sampling density irregularities. This poses seemingly no problem when these techniques are used in conjuction, however, it is unclear whether one or another is to be preferred. We investigate this numerically in Chapter 3.

2.2.4.3 *Spectrum Truncation*

The third and last modality that separates KAF and GPR is the spectrum truncation. In Section 2.2.3, we showed that it comprises Step II of the KAF algorithm, here we outline its purpose.

COMPUTATIONAL SPEED-UP:
As mentioned before, in practical applications using kernel methods such as KAF essentially hinges on inverting the kernel matrix G. Due to the fact that it is an s. p. d. matrix, finding G^{-1} is trivial if its eigendecomposition is known. Finding ℓ largest eigenvalues (and corresponding eigenvectors) is possible with iterative Krylov-type methods such as *Arnoldi iteration*. This can be achieved in $\mathcal{O}(\ell N^2)$ operations, as opposed to $\mathcal{O}(N^3)$ for a full matrix inversion. The constant in front of ℓN^2 is slightly higher for the Arnoldi mehtod, however, since ℓ is typically at least on order of magnitude smaller than N, it provides a significant computational speed-up. For example, running KAF on the ENSO dataset (see Chapter 3) provides at least a ten-fold speed-up. At the same time, for small-scale problems (say, $N \lesssim 10^3$) using Arnoldi iterations can be slower.

REGULARIZATION:
Simultaneously, spectrum truncation acts as a regularizer. It is well-known that in practice, GP regression often produces an ill-conditioned kernel matrix G [85], and thus, requires some form of regularization. Another reason to regularize is to avoid overfitting issues. Among the most popular methods are adding white-noise kernel, method of induced points and using pseudoinverse. The latter differs slightly from spectrum truncation, as pseudoinverse method uses all non-zero eigenvalues, i. e. $\ell \equiv \operatorname{rank} G$.

So far, we have omitted details on selecting ℓ. This parameter can be chosen via three different approaches, listed from easiest to hardest to compute:

(i) setting ℓ to a constant value, typically a fraction of the whole number of data points, e. g. $\ell = \lceil 0.1N \rceil$;

(ii) choosing ℓ such that only a certain spectrum level is retained;

(iii) learning ℓ with a cross-validation technique.

The second approach involves a simple minimization procedure, for example:

$$\ell = \arg\min_{1 \leqslant j \leqslant N} \eta(\lambda_1, \ldots, \lambda_N; j), \quad \text{where}$$

$$\eta(\lambda_1, \ldots, \lambda_N; j) = \left|\lambda_j - \varepsilon_{\text{cutoff}}\right| \quad \text{or} \quad \eta(\lambda_1, \ldots, \lambda_N; j) = \left|\sum_{i=1}^{j} \lambda_i - \varepsilon_{\text{cutoff}}\right|,$$

and $\varepsilon_{\text{cutoff}} \in [0, 1)$ is a predetermined parameter. This way of choosing ℓ works well with the iterative nature of the Arnoldi method (that is, at each step of computing the next eigenpair, η can be easily evaluated). The bi-stochastic normalization makes this procedure uniform across problems of various nature.

The third approach is the most accurate one, but requires computing a loss function, and therefore, is also the most computationally intensive. Note that thanks to the use of the VB kernel, the dataset is only split into training and testing subsets *once*, so there should be no ambiguity when using these descriptions in the KAF setting. We also note that for dynamical systems, since forecasting each time horizon p means utilizing a different function $y^{(p)}$, the truncation parameter becomes a function of p: $\ell = \ell(p)$.

2.2.4.4 *Summary*

For quick reference, the three modalities and their effects are summarized here in Table 2.1. As follows from the previous discussion, each modality is a separate way of performing a certain part of the generalized regression algorithm. Despite being executed consecutively, these parts are largely independent from each other, in the sense that they do not need to know what other parts are doing. In Section 2.2.4.3, we briefly mention that normalization of the spectrum helps with interpreting spectrum truncation (and perhaps, makes it easier to choose appropriate cutoff threshold), however, the former is not necessary for the latter to work. Likewise, we emphasized that normalization works for any s. p. d. kernel function, not just the VB kernel.

This observation provides us a way to interpret the three modalities as binary switches, where each one can be independently turned on and off. Notwithstanding the context of the problem (dynamics forecasting or function regression) the KAF method then is the one with all three turned on. On the contrary, the GP regression is the one with all three turned off. This brings us to an idea for numerical inverstigation of the method: by switching the modalities on and off one by one, we can compare different variants and reason about the effects of each of them. The study that lays this out comprises Chapter 3.

	KAF	GPR
kernel	VB kernel k_{vb}; tuned with X data only	parameter-dependent kernel $k(\cdot,\cdot;\vartheta)$; tuned with (X,y) data
normalization	bi-stochastic normalization	typically none
regularization	spectrum trucation	white-noise kernel, induced points etc.

Table 2.1: Three KAF modalities summarized.

2.2.5 *Least-Squares Derivation of KAF*

Taking a data-driven basis as a starting point, the KAF predictor can be derived via ordinary least squares. In this view, it can be linked to the *kernel ridge regression* (KRR), which is not surprising given that GPR and KRR provide essentially the same method.

Assume that we are given an RKHS $\mathcal{K}(g)$ induced by a kernel g, and the first ℓ functions of the basis $\{u_j\}_{j=1}^{\infty}$ evaluated at fixed data points $X = \{x_n\}_{n=1}^{N}$ only:

$$U_\ell = \begin{pmatrix} u_1(x_1) & \dots & u_\ell(x_1) \\ \vdots & & \vdots \\ u_1(x_N) & \dots & u_\ell(x_N) \end{pmatrix}. \tag{2.21}$$

Since matrix U_ℓ comes from the eigendecomposition of an s. p. d. matrix G, it is orthogonal: $(U_\ell)^\top U_\ell = I$.

Formally writing an unknown function $f \in L^2$ as a decomposition $f = \sum_{j=1}^{\infty} \alpha_j u_j$, then truncating it to the first ℓ functions and evaluating them on X data points gives:

$$U_\ell \alpha = f(X) =: y, \tag{2.22}$$

where α is a vector of coefficients α_j, and we used notation

$$f(X) = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix}. \tag{2.23}$$

The least-squares solution without regularization is given by

$$\alpha = \left(U_\ell^\top U_\ell\right)^{-1} U_\ell^\top y = U_\ell^\top y. \tag{2.24}$$

Using the standard Nyström extension, we write

$$\varphi_j(s) = \lambda_j^{-1} \sum_{n=1}^{N} g(s, x_n) u_j(x_n), \tag{2.25}$$

with $\varphi_j(x_n) \equiv u_j(x_n)$ for any $n = \overline{1,N}$ and any $j = \overline{1,\ell}$.

Let $\Lambda_\ell \in \mathbb{R}^{\ell\times\ell}$ be a diagonal matrix containing the first ℓ eigenvalues on the diagonal. Plugging the last two formulas into the truncated decomposition of f gives

$$\begin{aligned} f(s) &= \sum_{j=1}^{\ell} \alpha_j \varphi_j(s) \\ &= \sum_{j=1}^{\ell} \alpha_j \lambda_j^{-1} \sum_{n=1}^{N} g(s,x_n) u_j(x_n) \\ &= g(s)^\top U_\ell \Lambda_\ell^{-1} \alpha \\ &= g(s)^\top U_\ell \Lambda_\ell^{-1} U_\ell^\top y. \end{aligned} \tag{2.26}$$

Observing that $U_\ell \Lambda_\ell^{-1} U_\ell^\top = U \Lambda_{[:\ell]}^\dagger U^\top$, we arrive at equation (2.19) once more.

2.3 ANALYIS OF KAF PROPERTIES

2.3.1 *Interpolation Property*

In this section, we look into several methods based on GPR and KAF, and whether they have the *interpolation property*, i.e. they evaluate exactly y_n for each x_n from the data set. We also consider the residual in each case.

As our starting point, we write the approximation formula in the general form:

$$z(s) = g(s)^\top \Gamma^\dagger y, \tag{2.27}$$

where we do not limit ourselves to the VB kernel, but we require it to be s. p. d. By varying the matrix Γ, we consider three specific cases (using z_1, z_2 and z_3 notation, respectively):

(i) standard GP regression, $\Gamma = G$,

(ii) GP regression with white noise, $\Gamma = G + \beta I$,

(iii) GP regression with ℓ-truncation of the spectrum, $\Gamma = G_{[:\ell]}$.

Since we are interested in the interpolation property, in each case we evaluate $z(\cdot)$ on the whole dataset:

$$z\,(X) = G\Gamma^\dagger y. \tag{2.28}$$

As is known for the GP regression, it interpolates the data:

$$z_1\,(X) = GG^{-1} y = y. \tag{2.29}$$

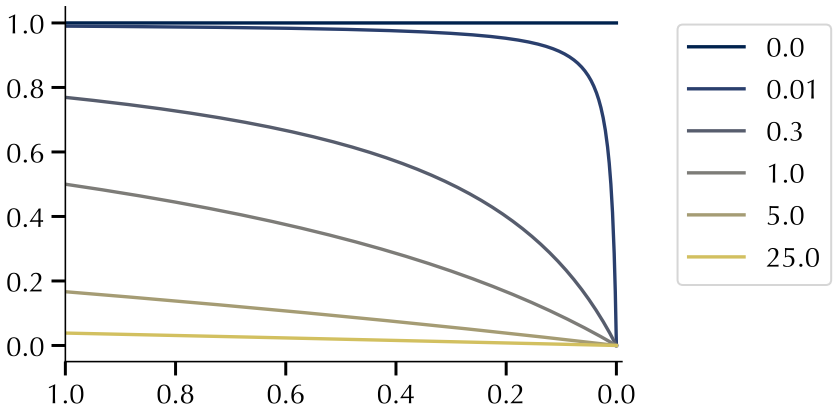


Figure 2.3: Function $w(x) = \frac{x}{x+\beta}$ plotted for various values of β.

The residual $r_1 = y - z_1(X)$ is equal to zero in this case, as is KAF conditional variance, for *any* input data:

$$v_1(s) = g(s)^\top G^{-1}(y - z_1(X))^2 = 0. \tag{2.30}$$

Here and throughout the rest of the section we write y^2 for a vector y to denote entry-wise square.

2.3.1.1 *GP Regression with White Noise*

Let $\Gamma = G + \beta I$, the regression formula then gives:

$$\begin{aligned} z_2(X;\beta) &= G\left(G + \beta I\right)^{-1} y \\ &= G\left(U\left(\Lambda + \beta I\right)U^\top\right)^{-1} y \\ &= U\Lambda\left(\Lambda + \beta I\right)^{-1} U^\top y \\ &= U \begin{pmatrix} \frac{\lambda_1}{\lambda_1+\beta} & & \\ & \ddots & \\ & & \frac{\lambda_N}{\lambda_N+\beta} \end{pmatrix} U^\top y. \end{aligned} \tag{2.31}$$

We see that the white-noise regularization does not preserve the interpolation property, and instead each projection of the data vector y onto the eigen-space is multiplied by $\frac{\lambda_i}{\lambda_i+\beta}$. The function $w(x) = \frac{x}{x+\beta}$ is depicted in Figure 2.3 for several values of β. We only consider the interval $(0, 1]$ because we assume that the bi-stochastic normalization was applied to the kernel matrix.

We now look at the residual, i. e. $r_2(\beta) = y - z_2(X;\beta)$:

$$\begin{aligned} r_2(\beta) &= y - U\Lambda(\Lambda+\beta I)^{-1}U^\top y \\ &= \left(UU^\top - U\Lambda(\Lambda+\beta I)^{-1}U^\top\right) y \\ &= U\left(I - \Lambda(\Lambda+\beta I)^{-1}\right)U^\top y \\ &= U\left(\beta(\Lambda+\beta I)^{-1}\right)U^\top y \\ &= U\begin{pmatrix} \frac{\beta}{\lambda_1+\beta} & & \\ & \ddots & \\ & & \frac{\beta}{\lambda_N+\beta} \end{pmatrix} U^\top y. \end{aligned} \tag{2.32}$$

It is easy to see that in the two limiting cases, $\beta = 0$ and $\beta \to \infty$, $z_2(X;\beta)$ takes the values y and 0, correspondingly.

2.3.1.2 *GP Regression with Spectrum Truncation*

Let $\Gamma = G_{[:\ell]} = U\Lambda_{[:\ell]}U^\top$, the regression formula then gives:

$$\begin{aligned} z_3(X) &= GG^\dagger_{[:\ell]}y \\ &= U\Lambda U^\top U\Lambda^\dagger_{[:\ell]}U^\top y \\ &= UI_{[:\ell]}U^\top y \\ &= U\begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} U^\top y, \end{aligned} \tag{2.33}$$

where $I_{[:\ell]}$ is a diagonal $N \times N$ matrix with ones in the first ℓ entries.

For the residual r_3, only the tail of the spectrum is retained:

$$r_3 = y - z_3(X) = UI_{[\ell:]}U^\top y, \tag{2.34}$$

where $I_{[\ell:]}$ is a diagonal matrix with ones in $\ell+1, \ell+2, \ldots, N$ positions, and zeros everywhere else.

It is clear that if $y \in \operatorname{span}\{u_1, u_2, \ldots, u_\ell\}$ then $z_3(x) \equiv z_1(x)$, i.e. we recover GPR formula exactly, which also means that $z_3(x)$ interpolates the data. This explains why in some numerical experiments we see almost exact prediction and, correspondingly, almost zero conditional variance: if the data vector y approximately lies in the span of the first ℓ eigenvectors then, given that data points are sampled sufficiently densely, the interpolation will be almost exact and the residual will be close to zero.

CHAPTER 3

NUMERICAL EXPERIMENTS ILLUSTRATING KAF

3.1 INTRODUCTION

The three modalities described in Chapter 2 (the choice of the specific VB kernel, the bi-stochastic normalization that normalizes the spectrum to the $[0, 1]$ interval and guarantees that the eigenfunction corresponding to the maximum eigenvalue 1 is constant, and truncation of the spectrum that serves as both a regularization and a computational speed-up technique) provide a theoretical connection between the two regression methods, kernel analog forecasting (KAF) and Gaussian process regression (GPR). And even though these two methods have been historically used in different contexts (dynamics forecasting and function interpolation, respectively), we may apply them to the same set of problems. This will open up the use of new methodologies in specific areas, and help to develop an understanding of the scope of their application, complexity estimates, pros and cons etc.

In this chapter we continue the comparison of the two methods, now turning our attention to the numerics. In order to disentagle the effects of the three modalities that KAF introduces into the kernel regression, we conduct a series of numerical experiments with one modality turned on at a time, thereby providing a more fine-grained, in-depth analysis. As outlined in the previous chapter, these modalitities can be thought of as binary switches. Each of them may be turned on or off, giving a total of $2^3 = 8$ combinations, with the GPR case being when all three are switched off, and the KAF case — when all three are switched on (see Table 3.1).

Remark. The field of kernel regression has developed a deep theory on s. p. d. kernels, owing its roots to the early advancements in functional analysis, and beyond. The question of choosing a specific kernel has been widely studied in many contexts, and is still a topic of active research. There is a plethora of heuristically successful kernels used by practitioners in various fields of science, and in full generality, theory (including one developed for KAF) allows the use of any s. p. d. kernel. However, here we limit the scope of our comparison to the RBF kernel only:

$$k_{\text{rbf}}(x, y) = \exp\left(-\frac{\|x - y\|^2}{\varepsilon}\right).$$

The motivation for doing so is twofold: relying on the existing literature, we hope that comparison to the RBF kernel would provide enough insight into the VB kernel, by way of reflexivity; moreover, the VB kernel is *morally* a squared-exponential kernel, similar to the self-tuning [119] and Gibbs kernels, and its closest analog among "standard" kernels is precisely the RBF. Thus, in this chapter we switch between VB (on) and RBF (off) kernels.

	GPR	V_1	V_2	V_3				KAF
VB kernel	0	1	0	0	1	1	0	1
bi-stoch. norm.	0	0	1	0	1	0	1	1
spectrum trunc.	0	0	0	1	0	1	1	1

Table 3.1: Eight variants, spanning the "distance" between GPR and KAF.

Regression variants that are used in this chapter (besides GPR and KAF) are labeled V_1–V_3, giving 5 methods to be tested in total. Since each of the variants (including GPR and KAF) must be tested against *some other* regression method, the possible number of combinations quickly grows too large to have a chance at providing a meaningful summary of the results. Therefore, we focus on just three experiments, listed below:

I. GPR vs V_1,

II. GPR vs V_2,

III. GPR vs V_3.

Experiments I-III are comparisons of the standard GPR versus a variant with one single switch on. These are the most straightforward setups, and we note that in one way or another, all three have been carried out before in the existing literature [13]. We run these experiments on several data sets though, amongst which are dynamical systems timeseries, with the usual goal of forecasting evolution from a given initial condition.

The choice of the Experiments I-III is dictated by our interest in understanding the effects of each of the switches on the forecasting skill. Another natural choice would be to compare KAF with a single switch turned off, however, leaning towards simpler and more well-studied methods (such as GPR) provides a clearer picture.

The chapter is organized as follows. Section 3.2 introduces test problems and data sets that are utilized in numerics throughout this chapter. The results of Experiments I-III are reported in Section 3.3.

3.2 TEST PROBLEMS AND DATA SETS

Numerical experiments in this chapter are performed on three datasets: *harmonic oscillator* (HO), *Lorenz '63* (L-63), and *El Niño–Southern Oscillation* (ENSO). The first two come from model dynamical systems, while ENSO is a dataset generated by a general circulation model which emulates the atmospheric patterns of Earth. Common with these datasets is the structure of the data:

- each dataset is one continuous timeseries of dynamical evolution;
- data is provided as a tuple

  $$\chi = \left\{x_n, y_n = F(x_n)\right\}_{n=1}^{N},$$

  where each $x_n \in \mathcal{X} \subseteq \mathbb{R}^c$ is a vector representing the observable coordinates, $y_n \in \mathbb{R}$ is a response variable, and $n = \overline{1, N}$ represents the time index;
- discrete time steps are fixed and constant throughout each dataset (although, they differ between datasets).

Before the main computation, datasets are split into several *chunks* and *sections*. First, the whole timeseries is partitioned into *train*, *validation* and *test* chunks. As mentioned before, the VB kernel does not require hyperparameter tuning (rather, its tuning does not use the y part of the data), and so validation set is not needed. However, since we also use the RBF kernel in some experiments, for consistency, we always split the timeseries into the three chunks.

Second, each of the three chunks is partitioned into the *main* and *extra* sections. Because of the nature of timeseries forecasting, the maximum time horizon $p_{\max}$ must be set in advance. This is because, for a given time horizon p, responses are formed by marching p steps forward in relation to the observables:

$$Z^p(x_n) \approx y_{n+p}.$$

Therefore, only the first $N - p_{\max}$ data points from the observation space can be used, which constitutes the main section; the rest goes into the extra section (i. e. points with indices $N - p_{\max} + 1, N - p_{\max} + 2, \ldots, N$).

To prevent ambiguity, in this chapter we use superscript for coordinate notation x^i to distinguish it from time index with subscript x_n.

Harmonic Oscillator

Harmonic oscillator is a continuous-time two-dimensional dynamical system:

$$\begin{aligned} x^1(t) &= \cos(1.7t), \\ x^2(t) &= \sin(1.7t). \end{aligned} \tag{3.1}$$

In Chapter 4, we revisit this system, adding amplitude, frequency and phase shift parameters in order to test KAF parametric extensions; here we simply use system (3.1) with fixed parameters.

The data set consists of a single trajectory, starting from initial point $(0, 1)$ and spanning $N = 7000$ steps with a time step $\Delta t = 0.01$. In all numerical runs, the initial condition is chosen so that it does not coincide with any of the points from the training set (but still lies on the 1.7-radius circle).

Lorenz '63

The second model dynamical system is a three-dimensional system of ODEs, colloquially known as Lorenz '63:

$$\begin{aligned} \dot{x}^1 &= \sigma(x^2 - x^1), \\ \dot{x}^2 &= x^1(\rho - x^3) - x^2, \\ \dot{x}^3 &= x^1 x^2 - \beta x^3, \end{aligned} \tag{3.2}$$

where $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$ are parameters of the system. These values are known as the *classical* parameter values, and it is proven that a chaotic attractor with Hausdorff dimension ≈ 2.06 exists for these values [106, 110].

The data set generated with this system contains a single timeseries with $N = 61\,000$ points. Of these points, $40\,500$ is split for the training chunk, and $p_{\max} = 500$ are reserved for time horizon forecasting. Thus, exactly $40\,000$ points are used for training (i. e. the kernel matrix is $40\,000 \times 40\,000$.

The whole trajectory is obtained using the Runge–Kutta Dormand–Prince method of 5^{th} order (`DOPRI5`). Even though the method itself has an adaptive time step, it has the so called *dense output* feature, which means that it can interpolate between the steps without loss of order. We use this feature to mimic the fixed step size $\Delta t = 0.01$. Additionally, we set both absolute and relative tolerances to 10^{-7} to guarantee high precision.

Since L-63 is a chaotic system with a global attractor, we are interested in the behaviour of the system on that attractor. Hence, we let the `DOPRI5` integrator run for spin-up time $T_{\text{spinup}} = 100.0$ first, and then restart the integrator from the last step of the spin-up run. This way we obtain a trajectory that lies in a close neighbourhood of the attractor.

In order to obtain a set of initial conditions for testing purposes, we perform the above procedure one more time: choose a different starting initial condition, run `DOPRI5` for $T_{\text{spinup}} = 100.0$, and then restart the integrator. This results in an entirely new trajectory (much shorter, with $N = 500$) without any points overlapping with the training chunk, yet still in the vicinity of the attractor. This new trajectory is used to evaluate the performance of the methods (e. g. compute the RMSE).

El Niño–Southern Oscillation

The last dataset comes from a pre-industrial control integration of the CCSM4 [38]. This is the same control integration as used in Wang, Slawinska, and Giannakis [113] to evaluate performance of KAF, and is available at:

`https://www.earthsystemgrid.org/dataset/ucar.cgd.ccsm4.joc.b40.1850.track1.1deg.006.html`

The dataset we use consists of 1300 years of monthly averages (thus, $N = 15\,600$) of the sea surface temperature (SST) fields sampled at approximately 1 resolution, on the Indo-Pacific longitude-latitude box 28°E–70°W and 30°S–20°N. The total number of coordinates is 44 771. The output variable is the (one-dimensional) Niño 3.4 index [7]. It is computed as the difference between a 3-month running mean of the SST and a reference value (computed over a 30-year period of observations). The spatial average is performed over the longitude-latitude box 170°W–120°W and 5°S–5°N.

We split the whole dataset into the training chunk with 1100 years (that is, $N = 13\,200$), and the testing chunk with 200 years ($N = 2400$). For this dataset, we use $p_{\max} = 50$. Moreover, for this dataset we also employ *delay embedding* to achieve better forecasting skill, following several works in this area [40, 100, 113]. Delay embedding is discussed in more detail in Chapter 4, Section 4.5.1; here we simply note that, in simple terms, it stacks several consecutive $x_n, x_{n+1}, \ldots, x_{n+b-1}$ vectors into one long one $\tilde{x}_n$ whose dimension is b times the original dimension of x_n's. Since each vector in the ENSO dataset represents a monthly average, a natural delay embedding dimension is $b = 12$, as then it contains timeseries of one consecutive year. Therefore, the total dimension becomes 537 252. Delay embedding also decreases the total number of data points by $b - 1$, so taking into account the maximum forecasting horizon $p_{\max}$, main section of the training chunk contains $N = 13\,139$ data points.

3.3 COMPARISON OF GPR VERSUS SINGLE-SWITCH-ON VARIANTS

This section comprises the first set of experiments, namely comparisons between the plain GP regression, and GP regression with a single modality borrowed from KAF.

In addition to the standard methodology described in Chapter 2, here we employ a few practical enhancements. First, as briefly mentioned before, the algorithm that tunes the variable-bandwidth kernel requires an external parameter: the number of nearest neighbors from which to compute the adhoc density. We express it as a fraction of total number of training points (e. g. $\lceil 0.1N \rceil$). From hundreds of numerical test runs, we have observed that a value between 0.07 and 0.2 is typically sufficient, with higher values providing smoother results (both

in terms of the final density estimation, and the resulting eigenvectors, in turn, influencing the smoothness of the forecast). Thus, we fix it to 0.15 in all our experiments unless otherwise noted.

Second, previously it was observed that GPR in general, and KAF in particular, performs better when sparsification of kernel matrices is used. Continuing this idea, we employ the following procedure to obtain a sparse version of the kernel matrix.

Definition 3.1. Let the *sparsification parameter* α_{sparse} be a real number in $[0, 1)$, and let K be a matrix in $\mathbb{R}^{N\times M}$. The *sparsifying operator* $\mathsf{S}\colon \mathbb{R}^{N\times M}\times[0,1) \to \mathbb{R}^{N\times M}$ is defined as follows:

$$\left\{\mathsf{S}(K, \alpha_{\text{sparse}})\right\}_{nm} := \begin{cases} K_{nm}, & \text{if } K_{nm} > \min K + \alpha_{\text{sparse}}(\max K - \min K), \\ 0, & \text{otherwise,} \end{cases}$$

and $\max K$, $\min K$ are computed over all entries of the matrix K.

Here we use $\alpha_{\text{sparse}} = 0.8$ for the main kernel matrix, and $\alpha_{\text{sparse}} = 0.6$ for the computation of the kernel field. Such high values of the sparsification parameter should not be surprising: as we only use squared exponential kernels (RBF and VB), the values of the kernel matrix decay very quickly with distance. Thus, high α_{sparse} values mostly zero out values that are already close to zero.

An alternative approach to sparsification is to set a certain number N_{sparse} of non-zero values per each row or column. In the context of kernels depending on the distance, this can be interpreted as keeping N_{sparse} nearest neighbors for each data point. The relationship of being a nearest neighbor is not transitive, though, in other words, if x_n is within the N_{sparse} nearest neighbors of x_m, the converse is not necessarily true. This means that iterating over rows, for example, such sparsification might result in a non-symmetric matrix, and some form of symmetrization is needed at the end. To avoid such complications, we stick to the sparsification operator S, having observed that numerical results do not vary significantly when using one or the other sparsification procedure.

3.3.1 *Experiment I*

Here we compare the GPR and V_1 variants, that is, numerically investigate whether substituting the RBF kernel k_{rbf} with the VB kernel k_{vb} produces similar results (see Table 3.1). We arrive at the conclusion that the use of k_{rbf} achieves the same forecasting quality, provided that the parameter ε is well tuned.

As discussed in Chapter 2, a key difference between the GP regression and KAF is the tuning procedure:

- for KAF, the tuning happens inside the construction of the variable-bandwidth kernel, that is, estimation of the manifold dimension, sampling density and

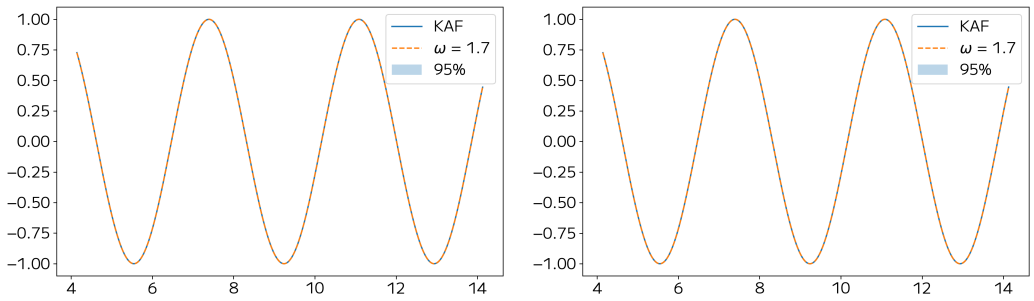


Figure 3.1: Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 0.001$.

bandwidth itself are all derived from the input data $\{x_n\}_{n=1}^N$ (which can be thought of as an inner-loop optimization);

- for GPR, the hyperparameters (e. g. ε in k_{rbf}) are typically tuned via an outer loop, that is, an optimization algorithm is applied to the loss function of the following form:

$$J(\varepsilon) = \|z(X_{\text{tune}}; \varepsilon) - y_{\text{tune}}\|, \tag{3.3}$$

where ε denotes a vector of hyperparameters, and X_{tune} and y_{tune} denote data set that has no intersection with the train set.

Harmonic Oscillator

For this data set, in the RBF case we found a suitable parameter $\varepsilon = 0.001$ by hand because the range of acceptable values is very wide. We observe both coordinates for training, and predict x^1.

The results are presented in Figure 3.1. As is clear from the plots, both results are indistinguishable from the ground truth.

Here we note that sparsification of the kernel matrix only played a significant role for the VB kernel, while it had no effect in the RBF case for the chosen parameter value. For VB, the chosen values of the sparsification parameter α_{sparse} is 0.8 for both kernel matrix and kernel field. An example of poorly tuned sparsification parameter is depicted in Figure 3.2.

On a different note, even when forecasting was not as good, the conditional variance of KAF was not producing meaningful results. This relates to the discussion at the end of Section 2.3.1.

Lorenz '63

The results are presented in Figure 3.3. As before, we observe all coordinates and predict x^1. Again, we tune the sparsification parameter for the VB kernel (0.8 for kernel matrix and 0.6 for kernel field), and the ε parameter for RBF (1.0). As

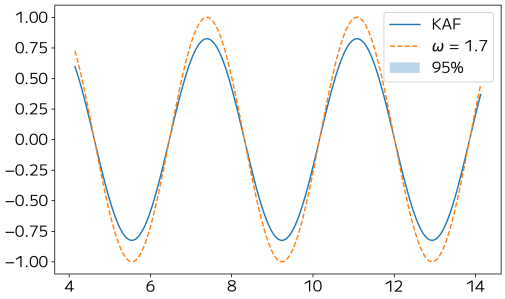


Figure 3.2: The VB kernel with a poorly-tuned sparsification parameter.

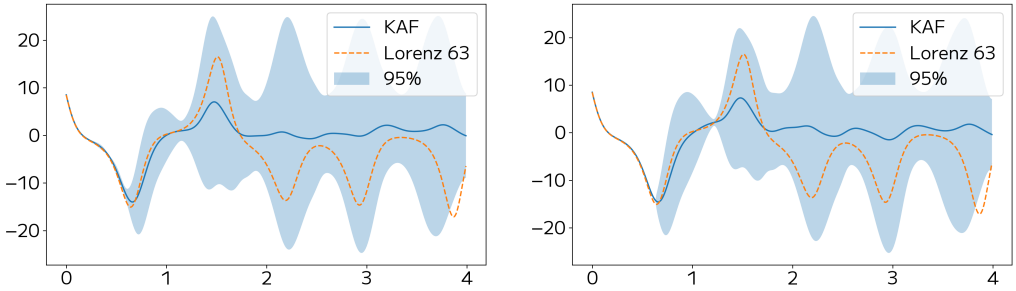


Figure 3.3: Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 1.0$.

with the harmonic oscillator, sparsification had no effect on the forecasting skill in the RBF case, but unlike that model, it would not deteriorate forecasting with the VB kernel as much Figure 3.4.

Again, we see no visible difference between the two kernels. The conditional variance, however, clearly provides useful uncertainty quantification for this dynamical system.

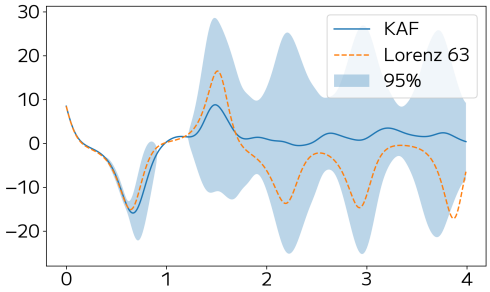


Figure 3.4: The VB kernel with a poorly-tuned sparsification parameter.

El Niño–Southern Oscillation

The results are presented in Figure 3.5. The RMSE is computed using 2351 points, since the total available number of the points in the testing chunk was 2400 and $p_{\max} = 50$.

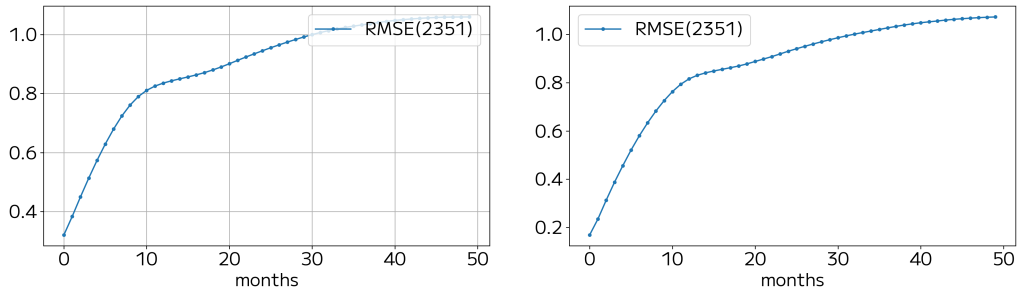


Figure 3.5: Result of applying the VB (left) and RBF (right) kernels. For RBF, $\varepsilon = 300\,000$.

3.3.2 *Experiment II*

Experiment II juxtaposes the standard GP regression with the regressor after bi-stochastic normalization (see Table 3.1).

Harmonic Oscillator

For HO, we report no difference between regression with and without normalization. As in Experiment I, both methods forecast with no error.

Lorenz '63

The results are presented in Figure 3.6.

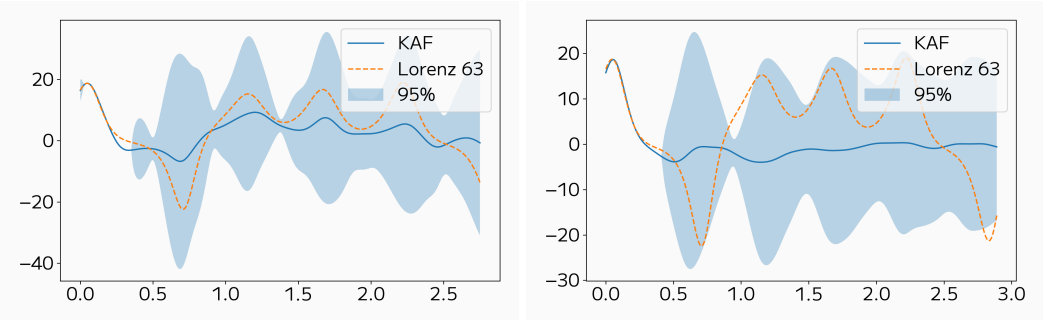


Figure 3.6: Result of applying regression without (left) and with (right) normalization.

El Niño–Southern Oscillation

The results are presented in Figure 3.7. As can be seen from the RMSE plots, the results vary insignificantly.

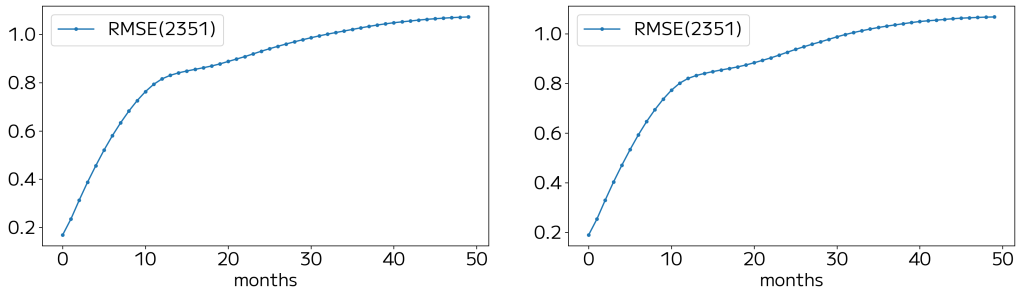


Figure 3.7: Result of applying regression without (left) and with (right) normalization.

3.3.3 *Experiment III*

Experiment III compares the standard GP regression and the spectrum-truncated version (see Table 3.1).

Harmonic Oscillator

Similar to two previous experiments, there is no difference between GP regressor with and without spectrum-truncation.

El Niño–Southern Oscillation

The results are presented in Figure 3.8. As can be seen from the RMSE plots, the results are essentially the same. This should come as no surprise, since the spectrum decays very quickly (Figure 3.9).

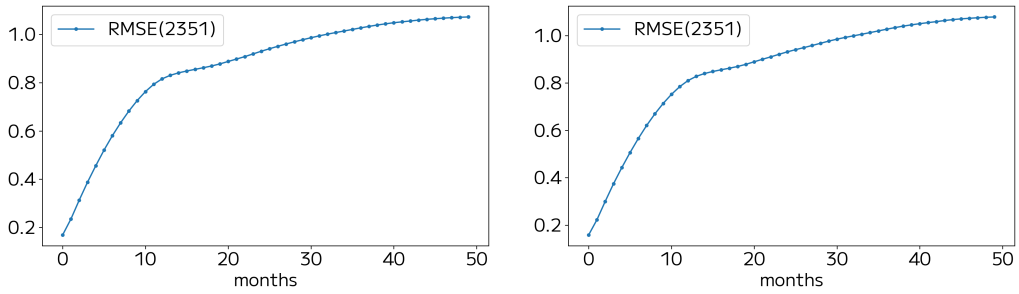


Figure 3.8: Result of applying regression without (left) and with (right) truncation.

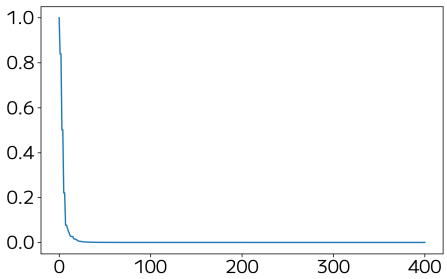


Figure 3.9: Spectrum of the kernel matrix, evaluated on ENSO dataset (first 400 values).

CHAPTER 4

PARAMETRIC EXTENSION OF THE KAF

4.1 INTRODUCTION TO PARAMETRIC DEPENDENCE

Many dynamical systems are parametrically-dependent, i. e. have one or several parameters that define its behavior. For simplicity, we limit ourselves to one-parameter models, with the belief that ideas outlined here can be extended to several parameters. In this chapter, we consider a parametric extension of the KAF with two settings in mind: (1) the data is given in the form of multiple trajectories, each of which is obtained for a fixed *and known* parameter value; (2) the data is given in the same form, however, the actual parameter values are unknown. We will refer to case (1) as *explicit* and case (2) – *implicit*. It is assumed that the parameter value does not change throughout one time-series, and it is always known which time-series correspond to different parameters, even in the implicit case.

Naturally, our goal in both cases is to predict evolution of the dynamical system for a given parameter value (whether one that is present in the training data set, or one that lies between the smallest and the largest values – keeping in mind that we deal with the one-dimensional case only), and starting from an unseen initial point. In the explicit setting, in the best-case scenario, one would hope that having a pair of the form $(x_0; \lambda)$, where x_0 is the initial point, and λ is a parameter value, would suffice. However, this would be impossible for the implicit case, except in certain trivial cases (such as when a parameter does not affect the system's behavior). Thus, in case (2), we *must* use additional information in order to make predictions. Perhaps the most natural and straightforward methodology for such a task is delay embedding; we devote our methodology in case (1) entirely to this approach and its description is contained in Section 4.5.1.

4.2 ORGANIZATION AND SUMMARY OF THE RESULTS

The chapter is organized as follows. We introduce a set of test problems in Section 4.3 which are used throughout the chapter to numerically test proposed approaches to parametric extensions of the KAF. These problems are comprised of two subsets:

- contolled settings which we can analyze before applying the KAF, and by comparing our intuition with the actual results of the numerical experiments, draw conclusions about the forecasting skill;
- settings that are closer to the real-world dynamical systems which serve as a test of applicability of the KAF parametric extensions.

In Section 4.4, we study the explicit parameter case (1); the same is then done for the implicit case (2) in Section 4.5. We explore the various options for extending KAF as a forecasting technique for parametrically dependent dynamical systems. The proposed extension are then applied to a suite of test problems, highlighting some advantages and drawbacks of each approach. In particular, even though the delay embedding lacks a rigorous mathematical guarantee for successful applications in forecasting (as discussed in Section 4.5.1), it is indeed an efficient practical tool. Furthermore, we found that certain cases are easier to tackle: generally, if the location of the manifold or attractor changes in the phase space as the parameter varies, then the forecasting skill is much higher compared to those cases when only the topology changes (like going from a chaotic attractor to a limit cycle of small period), or only the velocity of the oscillations changes. We observe that forecasting in the former cases (with shifts in the phase space) is sometimes even possible in the implicit case (2) *without* the use of delay embedding (demonstrated by d-KAF (4.6) and cc-KAF (4.7)).

Section 4.6 concludes with a summary of the results and an overview of the possible further directions.

4.3 TEST PROBLEMS

Here we introduce five test problems, stemming from two dynamical systems: the harmonic oscillator (HO) and Lorenz '63 (L-63). The former is a simple periodic two-dimensional oscillator that allows us to study the proposed parametric extensions of KAF in a controlled setting, while the latter, being, perhaps, the most well-known chaotic dynamical system, sheds some light on how such extensions would work on idealized, i. e. noiseless, chaotic systems. Introduced in 1963 by Edward Lorenz [64], the L-63 system still serves today as a prototypical example for testing numerical methods on chaotic systems, as it was proven in 1999 [106] that the system does exhibit chaos for certain parameter values, referred to as the classical values (see eq. (4.3) below).

4.3.1 *Harmonic oscillator*

The continuous-time harmonic oscillator dynamical system is defined as follows:

$$\begin{aligned} x(t) &= E\cos(\kappa(t+\alpha)), \\ y(t) &= E\sin(\kappa(t+\alpha)), \end{aligned} \tag{4.1}$$

with three parameters: amplitude E, frequency κ and phase shift α.

Experiment	Training values	Testing, seen	Testing, unseen
(HO-AMP)	$E = 1.0, 2.0, 3.0, 4.0$	$E = 2.0$	$E = 2.5$
(HO-FREQ)	$\kappa = 1.3, 1.6, 1.9, 2.1$	$\kappa = 1.6$	$\kappa = 2.0$
(HO-PH)	$\alpha = 0.0, 0.2, 0.4, 0.8$	$\alpha = 0.0$	$\alpha = 0.6$

Table 4.1: Parameter values used to generate training and testing data sets for HO experiments.

Fixing two of the parameters and varying the third, defines a test problem each:

(HO-AMP) varying amplitude E changes the location of the manifold on which data lives in the phase space, which is, arguably, the easiest mode for forecasting a dynamical system with parametric dependence;

(HO-FREQ) varying frequency κ does not change the manifold, and so, clearly, would make it impossible to distinguish between two different frequencies, but when sampled with equal time intervals, this problem in the discrete setting *is* amenable to forecasting;

(HO-PH) varying phase shift α is the hardest, as it tests the ability of the forecaster to shift the initial condition with data lying on the same manifold and sampled with equal intervals.

For simplicity, whenever the two of three parameters are fixed, we let them be as follows:

$$E = 1, \qquad \kappa = 1, \qquad \alpha = 0.$$

The setup for all three experiments is the same: we generate training data set by sampling system (4.1) with a constant time-step $\Delta t = 0.01$, using 3000 steps for each of the parameter values. For example, for (HO-AMP) we produce a total of 12000 data points, with 3000 for each of the values $E = 1, 2, 3, 4$. For each experiment, we also generate two testing data sets: one with a parameter value

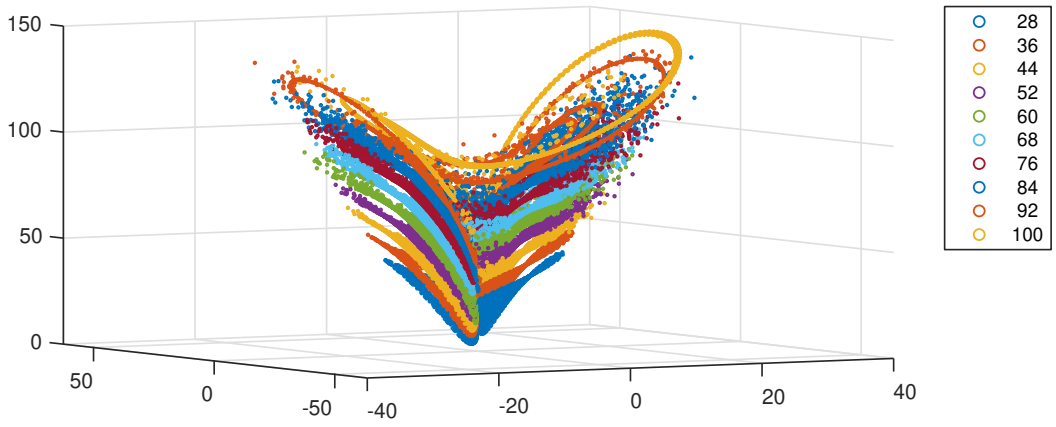


Figure 4.1: Lorenz '63 attractors resulting from various values of ρ.

that was in the training set, and one for a new parameter value. All parameter values for the HO set of experiments are summarized in Table 4.1.

In the explicit case, these three problems provide insights into how one might go about choosing a kernel, and which of the proposed in Section 4.4 approaches are better suited. In the implicit case, they allow us to theorize when delay embedding might work, and when it certainly *must* fail.

4.3.2 *Lorenz '63*

The L-63 model is a chaotic, ergodic system of differential equations with three parameters σ, ρ and β:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z. \end{aligned} \tag{4.2}$$

We define two problems:

(L-RHO) varying parameter ρ, the attractor shifts in the phase space along the z-coordinate (simultaneously getting more spread out and undergoing bifurcations transitioning to limit cycle behavior at $\rho \approx 100$), see Figure 4.1;

(L-SIGMA) varying parameter σ, the attractor changes its behavior from chaotic to periodic, see Figure 4.2.

The former setting is somewhat analogous to problem (HO-AMP), and thus is expected to be easier than the latter, which is a mixture of problems (HO-AMP) and (HO-FREQ).

The so called *classical* parameter values are:

$$\sigma = 10, \qquad \rho = 28, \qquad \beta = 8/3. \tag{4.3}$$

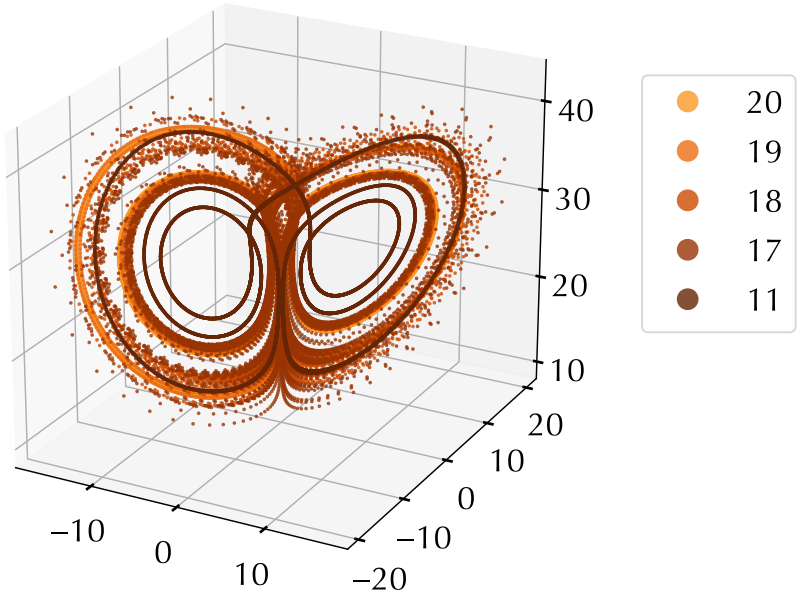


Figure 4.2: Lorenz '63 attractors resulting from various values of σ.

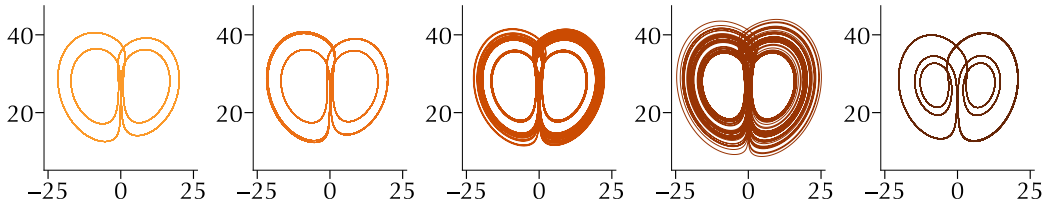


Figure 4.3: Same as Figure 4.2 (with same coloring), but 2-d projections onto $3x = 5y$ plane.

It is proven that a chaotic attractor exists for these values.

Experiment	Training values	Testing, seen	Testing, unseen
(L-RHO)	$\rho = 28 + 8k, \quad k = 0, \ldots, 9,$	$\rho = 28$	$\rho = 48$
(L-SIGMA)	$\sigma = 20, 19, 18, 17, 11$	$\sigma = 18$	$\sigma = 14$

Table 4.2: Parameter values used to generate training and testing data sets for L-63 experiments.

In the (L-RHO) experiment, as with the HO problems, we set two of the three parameters to these values except the one that is being changed, i. e. ρ. In the (L-SIGMA) experiment, we deviate slightly from the classical values (4.3):

$$\rho = 28, \qquad \beta = 1.1,$$

and the values of σ correspond to a limit cycle of period 2, period 4, chaos, chaos, and a limit cycle of period 3, respectively (Figure 4.3).

All parameter values used to generate training data sets, along with the values that generate testing (seen and unseen) data sets, are provided in Table 4.2.

Numerical integration is performed using the `RK45` method, as implemented in the `Python` library `SciPy` [109]. We use default tolerance values (`rtol = 0.001` and `atol = 1e-06`), but set the `max_step` parameter to 0.01 so that the solution is sampled at equal time intervals. Before any integration run, we perform the following steps:

1) set a seed for pseudo-random number generator (`numpy.random.seed`);
2) compute the coordinates of one of the two critical points, p_+:

$$p_\pm = \left(\pm\sqrt{\beta(\rho-1)},\ \pm\sqrt{\beta(\rho-1)},\ \rho-1\right)^\top;$$

3) sample normally-distributed random variables $\xi_i \sim \mathcal{N}(0, 0.01)$, for $i = 1, 2, 3$, with zero mean and 0.1 standard deviation, and add them to the coordinates of p_+ to obtain the spin-up initial conditions:

$$\begin{pmatrix} x_{\text{ic}} \\ y_{\text{ic}} \\ z_{\text{ic}} \end{pmatrix} = p_+ + \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}$$

4) spin-up L-63, i. e. numerically integrate system (4.2) (using the same solver parameters) for time t from 0 to 3000;
5) save coordinates of the last integration step as the new initial conditions.

4.4 EXPLICIT PARAMETER VALUES

4.4.1 *Fixed Parameters; One Time Series*

For ease of exposition, we provide a brief recap of the KAF formulation here.

Suppose the dynamical system is given by

$$\omega_{n+1} = \Phi(\omega_n), \tag{4.4}$$

where $\Phi\colon \mathcal{X}\times\mathcal{Y} \to \mathcal{X}\times\mathcal{Y}$, with the space of observations $\mathcal{X}$ and the latent space $\mathcal{Y}$. We work within the discrete-time dynamics setting, but also note that Φ can come from an appropriately discretized continuous-time dynamical system, i. e. it can be set to an evolution map $\Upsilon_{\Delta t}$ over time period Δt. We also assume the system is ergodic, and denote the invariant measure μ.

Let $\omega = \{\omega_n = (x_n, y_n)\}_{n=1}^N$ be a sequence in $\mathcal{X} \times \mathcal{Y}$ generated by (4.4). We are given data $d(\omega)$, comprised of two sequences:

$$\{x_n\}_{n=1}^N \subset \mathcal{X} \quad \text{and} \quad \{f_n = F(x_n, y_n)\}_{n=1}^N \subset \mathbb{R},$$

where the latter are computed from ω via mapping $F\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Our goal is to predict $F(x_q, y_q)$ for some number of discrete-time iterations $q \in \mathbb{N}$, given unseen initial condition x and data $d(\omega)$. Thus, we view the data-driven predictor as a map $Z_{q,d}\colon \mathcal{X} \to \mathbb{R}$ which takes initial condition x as input. Note, however, that it necessarily depends on data $d(\omega)$, which we simply denote as d, and also takes q as an input parameter.

Remark. Here we remind the reader that, even if we had access to the true dynamics Φ and response F, we would not be able to compute $F(x_q, y_q)$ as it would require the knowledge of the full phase-space initial condition (x, y), and not just the x part. This remark remains valid even in the case when F depends non-trivially only on the x variable. Such settings where the initial condition is unknown or only partially known, and the model is also known, can be linked to problems that data assimilation is concerned with — in the case of dynamical systems, and to the field of inverse problems — in general [91].

The KAF predictor has the following form:

$$\begin{aligned} Z_{q,d}(x) &= \frac{1}{N} \sum_{n=1}^{N} p(x, x_n; d) f_{n+q}(d), \\ p(x, x_n; d) &= \sum_{j=1}^{\ell(q)} \frac{\psi_j(x)\phi_j(x_n)}{\lambda_j^{1/2}}. \end{aligned} \tag{4.5}$$

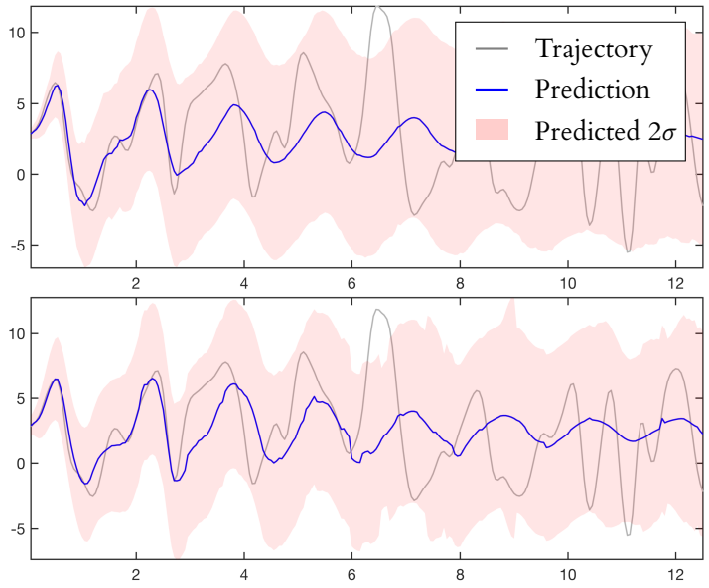
Here eigenpairs ϕ_j, λ_j, as well as Nyström extensions ψ_j are computed from data d; and $\ell(q)$ is the spectrum truncation number. We include explicit data dependence in the formula above so that we may emphasize the following: $p(x, x_n; d)$ does not involve knowledge of the time-ordering of the data, but $f_{n+q}(d)$ does. The weighting kernel $p(x, x_n; d)$ determines how much weight to attach to a time-series initialized at point x_n, according to its proximity to x, the desired initial point.

4.4.2 *Fixed Parameters; Multiple Time Series*

We now assume we are given *multiple* time-series $\{d^{(m)}\}_{m=1}^M$, each initialized by an initial point $x_0^{(m)}$, all of which are drawn independently at random from μ; we again label the union of these time-series by d. We also write

$$\left\{x_n^{(m)}\right\}_{n=1}^N \subset \mathcal{X} \quad \text{and} \quad \left\{f_n^{(m)}\right\}_{n=1}^N \subset \mathbb{R},$$

for the resulting constituent time series within $d^{(m)}$. For simplicity, and w. l. o. g., we suppose that the number of points within each time-series is constant and equal to N.



(a) d-KAF

(b) cc-KAF

Figure 4.4: Comparison of d-KAF (4.6) and cc-KAF (4.7).
The constituent time series are from the same larger dataset of chaotic L-63, here $N^{(m)} = 4000$, $M = 10$ and $N = 40000$.

In this setting two natural predictors suggest themselves. The first is based on straightforward Monte Carlo averaging of (4.5):

$$Z_{q,d}(x) = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{n=1}^{N} p\left(x, x_n^{(m)}; d^{(m)}\right) f_{n+q}^{(m)}\right). \tag{4.6}$$

The second one takes the following form:

$$Z_{q,d}(x) = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{n=1}^{N} p\left(x, x_n^{(m)}; d\right) f_{n+q}^{(m)}\right). \tag{4.7}$$

For the reason outlined below, we will refer to these as *diagonal* and *cross-correlated* KAF, respectively (and abbreviate as d-KAF and cc-KAF).

The difference here is subtle but significant: in the first case, the weighting kernel p only depends on each respective individual trajectory $d^{(m)}$, whereas in the second case it depends on all of the data d at once, exploiting the fact that the weighting kernel p does not require time-ordered data.

To present the difference in a more lucid way, we again refer to the connection with GP regression, and, in particular, a formula from Chapter 2:

$$Z_{q,d}(x) = g(x)^\top G_\ell^\dagger f(q).$$

Starting from d-KAF (4.6), we rewrite it in the matrix form:

$$\frac{1}{N}\sum_{n=1}^{N} p\left(x, x_n^{(m)}; d^{(m)}\right) f_{n+q}^{(m)} =$$
$$\begin{bmatrix} g(x, x_1^{(m)}) \\ \vdots \\ g(x, x_N^{(m)}) \end{bmatrix}^{\top} \begin{bmatrix} g(x_1^{(m)}, x_1^{(m)}) & \cdots & g(x_1^{(m)}, x_N^{(m)}) \\ \vdots & \ddots & \vdots \\ g(x_N^{(m)}, x_1^{(m)}) & \cdots & g(x_N^{(m)}, x_N^{(m)}) \end{bmatrix}^{\dagger,\ell} \begin{bmatrix} f_{1+q}^{(m)} \\ \vdots \\ f_{N+q}^{(m)} \end{bmatrix} =$$
$$g^{(m)}(x)^{\top} \left(G^{(m)}\right)^{\dagger,\ell} f^{(m)}(q),$$

where we use notation $A^{\dagger,\ell}$ to denote pseudoinverse of A that is taken after truncating the eigendecomposition to ℓ eigenpairs, and $f^{(m)}(q) = \left(f_{1+q}^{(m)}, \ldots, f_{N+q}^{(m)}\right)^{\top}$.

For exposition purposes, let $M = 2$, the formula for the d-KAF is then as follows:

$$Z_{q,d}(x) = \frac{1}{2}\left(g^{(1)}(x)^{\top}\left(G^{(1)}\right)^{\dagger,\ell} f^{(1)}(q) + g^{(2)}(x)^{\top}\left(G^{(2)}\right)^{\dagger,\ell} f^{(2)}(q)\right) =$$
$$\frac{1}{2}\begin{pmatrix} g^{(1)}(x) \\ g^{(2)}(x) \end{pmatrix}^{\top} \begin{pmatrix} \left(G^{(1)}\right)^{\dagger,\ell} & 0 \\ 0 & \left(G^{(2)}\right)^{\dagger,\ell} \end{pmatrix} \begin{pmatrix} f^{(1)}(q) \\ f^{(2)}(q) \end{pmatrix}.$$

If the truncation number ℓ is equal to N (which means no truncation), then we may further simplify it to:

$$Z_{q,d}(x) = \frac{1}{2}\begin{pmatrix} g^{(1)}(x) \\ g^{(2)}(x) \end{pmatrix}^{\top} \begin{pmatrix} G^{(1)} & 0 \\ 0 & G^{(2)} \end{pmatrix}^{-1} \begin{pmatrix} f^{(1)}(q) \\ f^{(2)}(q) \end{pmatrix}.$$

This is due to the fact that for two positive-definite symmetric matrices $A_1 = U_1\Lambda_1 U_1^{\top}$ and $A_2 = U_2\Lambda_2 U_2^{\top}$, the eigendecomposition of the block-diagonal matrix comprised of A_1 and A_2 is defined by the formula:

$$\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} = \begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix} \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix}^{\top}.$$

For the same $M = 2$ case, the cc-KAF formula has the following form:

$$Z_{q,d}(x) = \begin{pmatrix} g^{(1)}(x) \\ g^{(2)}(x) \end{pmatrix}^{\top} \begin{pmatrix} G^{(1)} & G^{(1,2)} \\ G^{(2,1)} & G^{(2)} \end{pmatrix}^{\dagger,\ell} \begin{pmatrix} f^{(1)}(q) \\ f^{(2)}(q) \end{pmatrix}.$$

Here the off-diagonal matrices $G^{(m_1,m_2)}$, $m_1 \neq m_2$, have cross-correlation terms of the form $g\left(x_i^{(m_1)}, x_j^{(m_2)}\right)$, with $i, j \in \{1, \ldots, N\}$. Note, in particular, that the

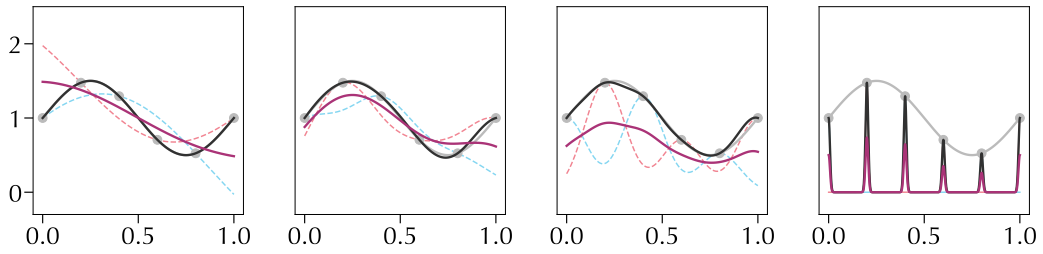


Figure 4.5: Comparison of the average (purple) of two GP regressors each trained on half points vs one trained on all (black), ordered by decreasing length scale.
Pink and light blue dashed lines are for two GP regressors trained on $(0.2, 0.6, 1.0)$ and $(0.0, 0.4, 0.8)$, respectively. In gray: the ground truth function $1 + 0.5\sin(2\pi x)$. All four GP regressors use $\exp(-\|x-y\|^2/\varepsilon)$ kernel, where $\varepsilon = 1.0$, 0.25, 0.15, and 0.01, from left to right.

factor $1/2$ disappeared: this is because the total number of points is $2N$, and so the eigenvector matrix satisfies $\Phi\Phi^\top = 2N \cdot I_{2N\times 2N}$, and thus the normalization $\frac{1}{2N}$ is needed.

Since G is symmetric, the off-diagonal matrices obey $G^{(1,2)} = \left(G^{(1,2)}\right)^\top$. These matrices contain correlations between $m = 1$ and $m = 2$ trajectories; thus, if the two trajectories are very dissimilar (say, lie in different lobes of an attractor, such as L-63), then the two methods should differ in approximately a factor of 2.

In an attempt to interpret this unusual conclusion from the GP regression point of veiw, suppose we take the trivial identity kernel $g_{\mathrm{id}}(x, y) = \delta(x - y)$, and let $N = 3$, $M = 2$. Suppose also that the space is the interval $[0, 1]$, and relabel points so that they have continuous indices:

$$\begin{aligned} x_1 &= x_1^{(1)}, & x_2 &= x_1^{(2)}, \\ x_3 &= x_2^{(1)}, & x_4 &= x_2^{(2)}, \\ x_5 &= x_3^{(1)}, & x_6 &= x_3^{(2)}. \end{aligned}$$

Evaluating posterior GPs z_1 and z_2 on x_1 would yield y_1 and 0, respectively, hence the average of the two is $y_1/2$. However, if we use all four points to construct $z^*(x) = g_{\mathrm{id}}(x)^\top G_{\mathrm{id}}^{-1} y$, then $z^*(x_1) = y_1$. This effect is further demonstrated in a series of experiments, where we take the squared exponential kernel and let $\varepsilon \to 0$ (Figure 4.5).

We now turn to testing whether both d-KAF and cc-KAF give similar answers in large data limits. The first is likely justifiable and if they are similar it helps to justify the second. If they do give similar answers then note that the latter is better suited to the next section in which we try to learn parameter-dependence. Results of this experiment are shown in Figure 4.4.

Finally, we note that, using, as before, Arnoldi iterations, d-KAF requires $\mathcal{O}(MN^2)$ evaluations of the kernel function, and $\mathcal{O}(\ell N^2)$ multiplications to obtain

the ℓ-truncated pseudoinverse; whereas cc-KAF requires $\mathcal{O}(M^2N^2)$ evaluations and $\mathcal{O}(\ell M^2N^2)$ multiplications.

4.4.3 *Variable Parameters; Multiple Time Series*

Now imagine the data is generated by a parameter $\lambda \in \Lambda$ dependent (possibly stochastic) dynamical system and let $\mathcal{X}_\Lambda = \mathcal{X} \times \Lambda$. Let $\omega = \{\omega_n = (x_n, y_n)\}_{n=0}^{N-1}$ be a sequence in $\mathcal{X} \times \mathcal{Y}$ generated by $\omega_{n+1} = \Phi(\omega_n, \lambda)$. We write the evolution of this (possibly stochastic) dynamical system as

$$\begin{aligned} \omega_{n+1} &= \Phi(\omega_n, \lambda_n; \zeta_n), \quad \zeta_n \overset{i.i.d.}{\sim} \nu \\ \lambda_{n+1} &= \lambda_n. \end{aligned} \tag{4.8}$$

Assume (for simplicity) that

$$\mu'(d\omega, d\lambda) = \mu(d\omega|\lambda)\,\mathrm{Leb}_\Lambda(d\lambda),$$

where Leb_Λ is Lebesgue measure on Λ, normalized to a probability (uniform measure on Λ). We implicitly assume that conditional measure $\mu(d\omega|\lambda)$ enjoys continuity properties with respect to λ; this is certainly not necessarily the case in all situations, but in the case of stochastic dynamics settings can be developed where it does hold. This assumption is also true for some dissipative ODEs: in particular, it is true (albeit for small perturbations of the classical parameters) for L-63, with measure $\mu(d\omega|\lambda)$ being the invariant measure defined on the system's attractor [106].

We then generate data in $\Omega \times \Lambda$ by choosing λ uniformly at random from Λ and then generating a time-series from the dynamical system (4.8) at this value of λ and generating data from it as in the previous sections, but with $\mathcal{X}$ replaced by $\mathcal{X}_\Lambda$, and x by (x, λ). We thus use a weighting kernel $q\colon \mathcal{X}_\Lambda \times \mathcal{X}_\Lambda \mapsto \mathbb{R}$. Previous kernel was based on $\|x - x'\|^2$, i. e. Euclidean norm. In the parameter-dependent setting a number of ideas are natural. The first is to set $z = (x, \lambda)$ and consider norm based on

$$\frac{1}{2}\langle z, Az\rangle$$

and then A is a parameter to be chosen to put the two contributions on the same scale. The second is to seek a product kernel w. r. t. x and λ separately with λ being computed analytically using large data asymptotics and uniform distribution, and then x being as before but with λ dependent variable bandwidth.

We may choose λ uniformly at random M times from Λ and then generate M time-series at M different values of λ, noting that λ does not evolve along the dynamical system. We may then use formula (4.7) noting that now it gives a function of both initial condition and parameter:

$$Z_{q,d}(x, \lambda) = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{n=0}^{N-1} p\left((x, \lambda), \left(x_n^{(m)}, \lambda^{(m)}\right); d\right) f_{n+q}^{(m)}\right). \tag{4.9}$$

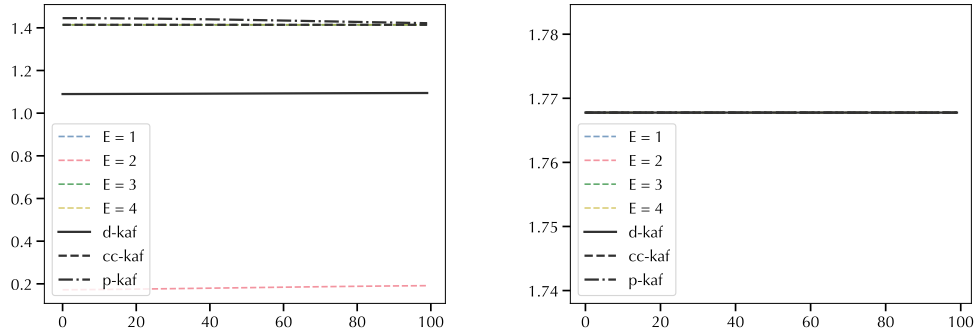


Figure 4.6: RMSE for (HO-AMP) test problem, forecasting for $E = 2.0$ and $E = 2.5$.

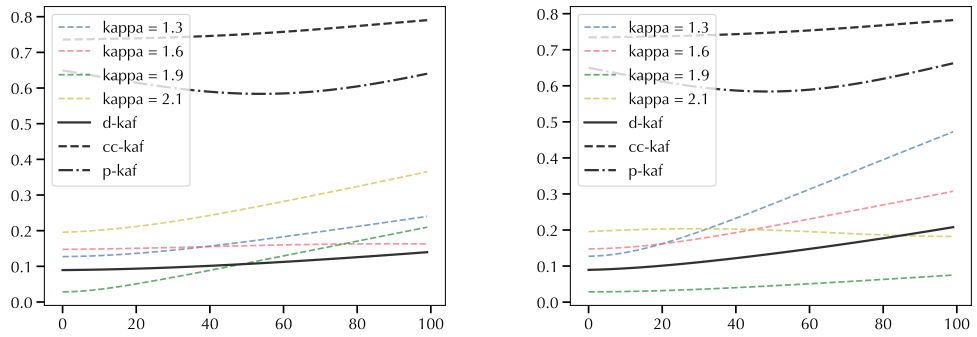


Figure 4.7: RMSE for (HO-FREQ) test problem, forecasting for $\kappa = 1.6$ and $\kappa = 2.0$.

4.4.4 *Case Studies*

We study the above ideas by comparing the predictors (4.6), (4.7), (4.9) on a dataset containing different parameter regimes. Results are demonstrated on the test problems outlined in the previous section, with single parameter variation in each case.

4.4.5 *Harmonic Oscillator*

Here we present results of the three groups of experiments outlined in Section 4.3.1.

We train individual regressors for each of the timeseries, and a d-KAF, cc-KAF and p-KAF regressors.

For each of the three cases we use two evaluations: one for a previously seen parameter value (i. e. one that is in the training set), and one for an unseen parameter value.

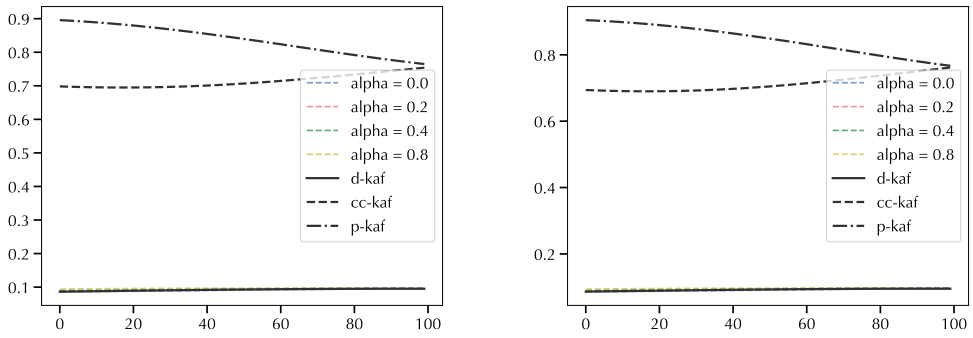


Figure 4.8: RMSE for (HO-PH) test problem, forecasting for $\alpha = 0.0$ and $\alpha = 0.6$.

The results for (HO-AMP), (HO-FREQ) and (HO-PH) are presented in Figures 4.6, 4.7 and 4.10, respectively.

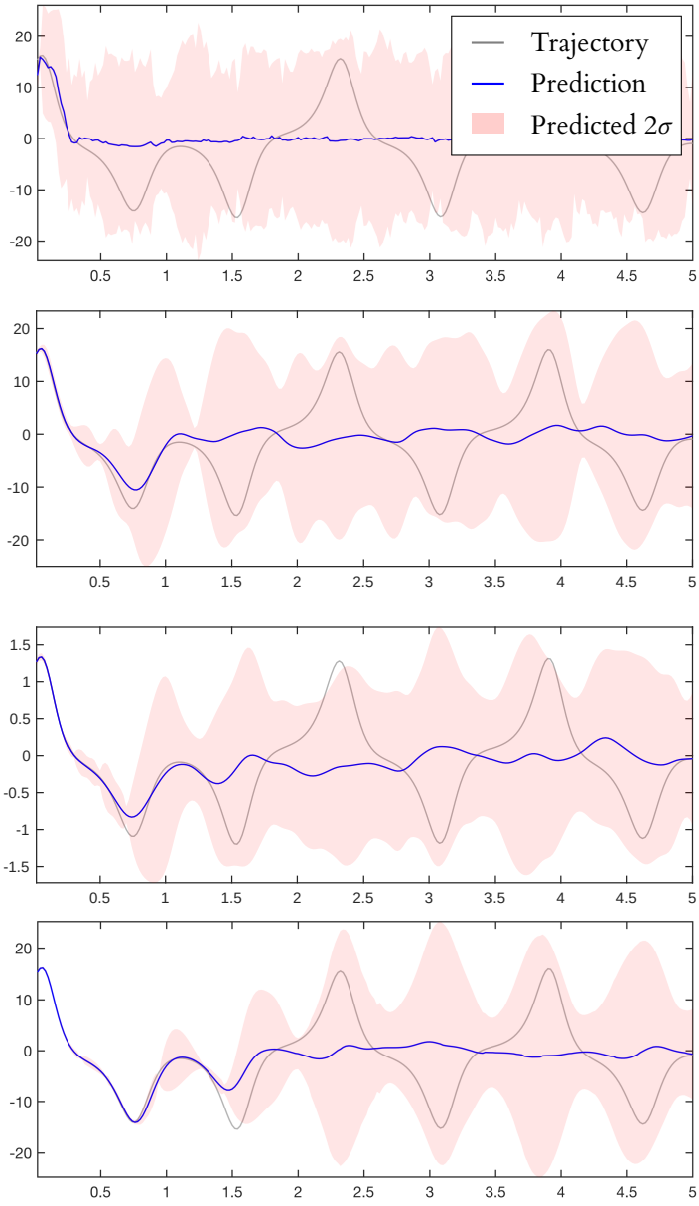
4.4.6 *Lorenz '63*

We compare the following variants of KAF applied to these multiple time series, ordered from least to best expected predictive skill:

1. d-KAF (4.6): the average of $M = 10$ KAF predictors trained separately on each time series,
2. cc-KAF (4.7): KAF trained all at once on the entire group of time series with no parameter information, namely, $H(\omega, \lambda) = [x, y, z]^\top$,
3. p-KAF (4.9): cc-KAF trained on parameter-augmented coordinates (mean centered and normalized), namely $H(\omega, \lambda) = [x, y, x, \lambda]^\top$,
4. KAF (4.5): the ideal scenario of training KAF entirely on the same fixed parameters (one time series) as the initial data.

Results are shown in Figure 4.1.

As is expected, p-KAF produces results that are slightly worse but similar to the standard KAF. Perhaps, what is unexpected is that p-KAF and cc-KAF produce similar results. For Lorenz '63 this can be simply explained by the fact that the attractor physically lies in different parts of the phase space (i. e. it moves up in z-direction as *rho* increases), so Euclidean distance in the phase space provides enough information; in other words, the weights assigned to samples $f_{n+q}^{(m)}$ in eq. (4.7), for $m \neq m(\rho)$, are rather small. This is in contrast to d-KAF, where there are no cross-correlation blocks $G^{(m_1,m_2)}$, $m_1 \neq m_2$, and so the trajectory converges to the prior mean, zero, much faster.



(a) d-KAF

(b) cc-KAF

(c) p-KAF

(d) standard KAF
$\ell = 1000$

Figure 4.9: Variable parameters; multiple time series prediction of $x_0 \approx 15$.

Ordered from least to best predictive skill from top to bottom, are plots of d-KAF (4.6), cc-KAF (4.7), and p-KAF (4.9). The bottom plot, included for illustrative purposes, depicts the ideal scenario of fixed parameters and single time series for training a standard KAF predictor (4.5).

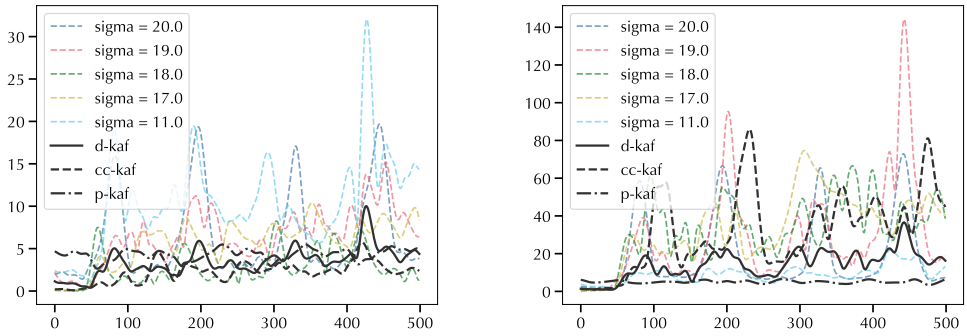
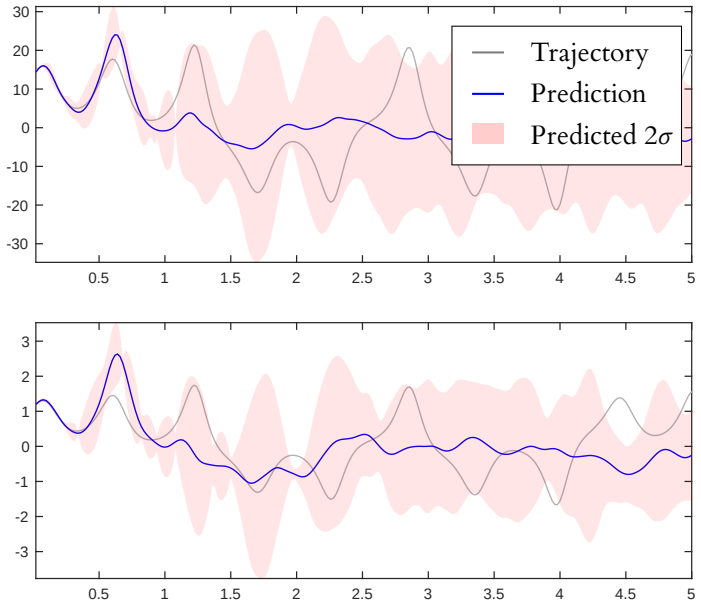


Figure 4.10: RMSE for (L-SIGMA) test problem, forecasting for $\sigma = 18.0$ and $\alpha = 14.0$.



(a) cc-KAF

(b) p-KAF

Figure 4.11: Comparison of cc-KAF (4.7) and p-KAF (4.9), for parameter value $\rho = 48$ which not in the training set; both have $\ell = 1000$ eigenpairs.

4.5 IMPLICIT PARAMETER VALUES

4.5.1 *Delay Embedding*

The main technique that allows us to extend KAF to the parametric setting is called *delay embedding*, a widely deployed technique based on a result proved by Floris Takens in 1981 [104]. In this subsection we state the main theorem, provide brief explanation of the use in the discrete dynamics case, introduce notation for delay embedding and discuss heuristics and computational aspects of using it in practice. The original Takens' embedding theorem is formulated below.

Theorem 4.1 (F. Takens [104]). *Let M be a compact manifold of dimension m. For pairs (ϕ, y), $\phi\colon M \to M$ a smooth diffeomorphism and $y\colon M \to \mathbb{R}$ a smooth function, it is a generic property that the map $\Phi_{(\phi,y)}\colon M \to \mathbb{R}^{2m+1}$, defined by*

$$\Phi_{(\phi,y)}(x) = \left(y(x), y(\phi(x)), \ldots, y(\phi^{2m}(x))\right)$$

is an embedding; by "smooth" we mean at least C^2.

For our purpores, this theorem implies that, given a discrete dynamical system, defined by iterations $x_{n+1} = \phi(x_n)$, it is possible to recover full dynamics by looking at one-dimensional *observations* of the trajectory, provided enough delay-points are used (twice as many as the dimension of a manifold on which dynamics lie).

A few practical issues arise here. First of all, chaotic attractors are not compact manifolds, for they have a fractional dimension. This poses a problem, however, there have been many generalizations of Theorem 4.1, some of which, in particular, replace the smooth manifold with a set of arbitrary box-counting dimension. Moreover, in practice, delay embedding tends to work well, which is confirmed by a vast number of studies over the period of several decades.

Second, we are not necessarily guaranteed that the dynamics "visits" the whole manifold M; thus, a time-series of observations $\{y(x_n)\}_{n=1}^{N}$, no matter how big N is, will not allow us to recover the whole manifold. This is circumvented by an assumption we make that we only consider ergodic systems with compact attractors, so any long enough trajectory is sufficient to cover the whole attractor. It is true, however, and just like in the normal KAF setting of Chapters 1 and 2, if the unseen initial condition does not lie on such attractor, the prediction skill is not expected to be high.

"Recovering dynamics" in this context means that the new, reconstructed dynamics in $\mathbb{R}^{2m+1}$ are diffeomorphically equivalent to the original dynamics. This, by itself, does not provide a forecasting technique, but applying any forecasting method on delay-embedded vectors *does*. A delay-embedded vector of

delay-embedding dimension b is defined as a stack of $(b-1)$ *previous* iterations of the dymical map, plus the current one:

$$x_n^{(b)} = [x_{n-b+1}, x_{n-b+2}, \ldots, x_n]^\top .$$

If the dynamics (or observations) are one-dimensional, then $x^{(b)} \in \mathbb{R}^b$; but if each of the time-series vectors x_n is s-dimensional, then $x^{(b)} \in \mathbb{R}^{sb}$.

Obviously, taking a time-series $\{x_n\}_{n=1}^N$ and turning it into a delay-embedded one means that the total number of points reduces to $N - b$; this usually poses no problem since $N \gg b$. Here we note that finding a good dimension b is mostly a heuristic exercise: in principle, Theorem 4.1 guarantees that $b = 2m + 1$ should be enough for a manifold of dimension m. This means that employing a higher embedding dimension should do no harm, after the $2m + 1$ dimension is reached. In practice, however, observations are inherently noisy (due to numerical errors of ODE intergration), so then a *delay embedding window* $T_{\text{emb}} = b\Delta t$ becomes important. The dimension is still required to be at least $2m + 1$, but too long of a window T_{emb} might yield worse results. To complicate matters further, if the window is too short, data-driven forecasts also become worse, even when the manifold dimension is known, and the $2m + 1$ requirement is satisfied. A possible explanation of this phenomenon is that many T_{emb} intervals of the trajectory are too similar to each other, hence the kernel does not provide sufficient separation, and the computations are poorly conditioned.

Heuristically, however, numerical experiments we conducted on test problems outlined in Section 4.3, as well as with other dynamical systems not covered in this chapter and whose stable manifold dimension is explicitly known, suggest that the best results from using delay embedding are achieved when two factors are satisfied:

- the delay-embedding dimension b is appoximately between $2m$ and $4m$, and
- the delay-embedding window T_{emb} is large enough to observe significant variation (in terms of the sensitivity of the chosen kernel), yet small enough to not accumulate large errors from the numerical integration.

Similarly, theory tells us that an observation function $y\colon M \to \mathbb{R}$ whose codomain is the real line, is enough, so higher dimensions should *not* deteriorate the embedding accuracy. In practice, however, it happens that observing just one coordinate is sometimes more beneficial than observing the full state, at least for the purposes of using KAF (as outlined below, in the harmonic oscillator subsection).

The computational cost of using delay embedding in the context of kernel methods depends on the choice of the kernel. For the squared exponential kernel $k(x, y) = \exp(\|x - y\|^2/\delta)$, or any other kernel that uses Euclidean norm as

distance, computation of each element of the kernel matrix from a pair of points x_i and x_j grows linearly from $\mathcal{O}(s)$ to $\mathcal{O}(sb)$. Thus, total complexity of computing one kernel matrix from N data points is $\mathcal{O}(sbN^2)$ (counting multiplications and evaluations of the exponential function only). Here we assume $N \gg b$, so we ignore changes in the total number of points.

As noted above, the embedding dimension b needs to be determined experimentally. For such procedure, efficient implementation of the delay embedding involves computation and storage of pairwise distances (p-d) $p_{ij} = \|x_i - x_j\|^2$. This increases the memory cost by $\mathcal{O}(N^2)$ for dense matrices. It can be further reduced by the use of sparse matrices, setting values $p_{ij} \ll 1$ to zero. Storing this additional matrix is not a significant challenge, though, since even for $N = 10^5$ points the total size of the upper-triangular part of the p-d matrix (its diagonal is zero, and it is symmetric) is ~4.7 GB. After the p-d matrix (p_{ij}) is computed, the new distances $\left(p_{ij}^{(b)}\right)$ for embedding dimension b are computed using a simple formula:

$$p_{ij}^{(b)} = p_{ij} + p_{i-1,j-1} + \cdots + p_{i-b+1,j-b+1},$$

and i and j range from b to N, thus $\left(p_{ij}^{(b)}\right) \in \mathbb{R}^{(N-b)\times(N-b)}$.

4.5.2 *KAF with Delay Embedding*

Delay embedding can be a useful tool for general time-series prediction, improving the forecasting skill even when full state of the system is observed and the system is Markovian (as can be seen from some of the results in Section 4.5.3). The obvious downside of using delay embedding is using a portion of trajectory instead of one point as the initial condition. In the context of forecasting parameter-dependent dynamical systems with hidden parameters, however, it is justified because it provides additional information to distinguish (x_0, λ_a) from (x_0, λ_b).

The most straight-forward approach of applying delay embedding to forecasting with a kernel method like KAF is to simply treat the newly formed delay-embedded vectors $x^{(b)}$ as coming from a phase space of an augmented dynamical system:

$$x_{n+1}^{(b)} = \Phi^{(b)}\left(x_n^{(b)}\right).$$

In the case of a kernel that uses Euclidean distance, this means that the L^2-distance is computed between two trajectory pieces that $x_i^{(b)}$ and $x_j^{(b)}$ represent. Obviously, there are other options, for example, using an approximation of

the H^1 norm, or approximation of the curvature of $x_i^{(b)} - x_j^{(b)}$, which, in the arc-length parametrization, is essentially the H^2 seminorm:

$$|x(\cdot) - z(\cdot)|_{H^2}^2 = \int_{[0,T]} |x''(t) - z''(t)|^2 dt,$$

and so on. We note that these choices cannot be made for a general forecasting problem, and would need to be made for a particular problem.

It is also important to note that Theorem 4.1, despite providing a solid theoretical ground for the use of delay embedding in the context of dynamical systems forecasting, does not guarantee success. Indeed, two attractors corresponding to parameter values λ_a and λ_b can be diffeomorphically equivalent yet the dynamics will differ. In practice, however, delay embedding demonstrates excellent separation of the various dynamics. Moreover, as demonstrated in the harmonic oscillator example, even in the case when the two manifolds corresponding to different parameter values are exactly the same, KAF with delay embedding still gives good results.

4.5.3 *Case Studies*

Similar to the explicit parameter case, we test KAF with delay embedding against pure KAF on the HO and L-63 data set. In each case, we conduct one experiment for each dynamical system.

4.5.4 *Harmonic Oscillator*

The results are presented in Figure 4.12. Here we only run the (HO-FREQ) experiment. Top row represents the case where we observe full state and only predict x coordinate, and the bottom row is observing and predicting only x coordinate. Left and right columns depict cases where the parameter is out-of-sample and from the training set, respectively.

4.5.5 *Lorenz '63*

The results are presented in Figure 4.13. Here we only run the (L-RHO) experiment. Top row represents the case where we observe full state and only predict x coordinate, and the bottom row is observing and predicting only x coordinate. Left and right columns depict cases where the parameter is out-of-sample and from the training set, respectively.

4.6 CONCLUSION AND FURTHER DIRECTIONS

Among the possible further directions we indicate the following:

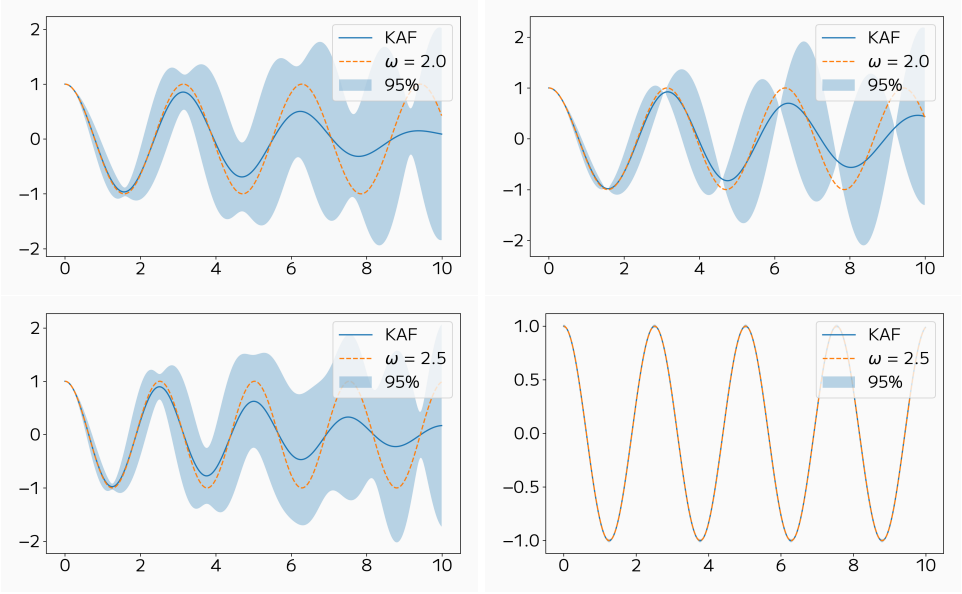


Figure 4.12: Applying KAF to HO with delay-embedding dimension 24.

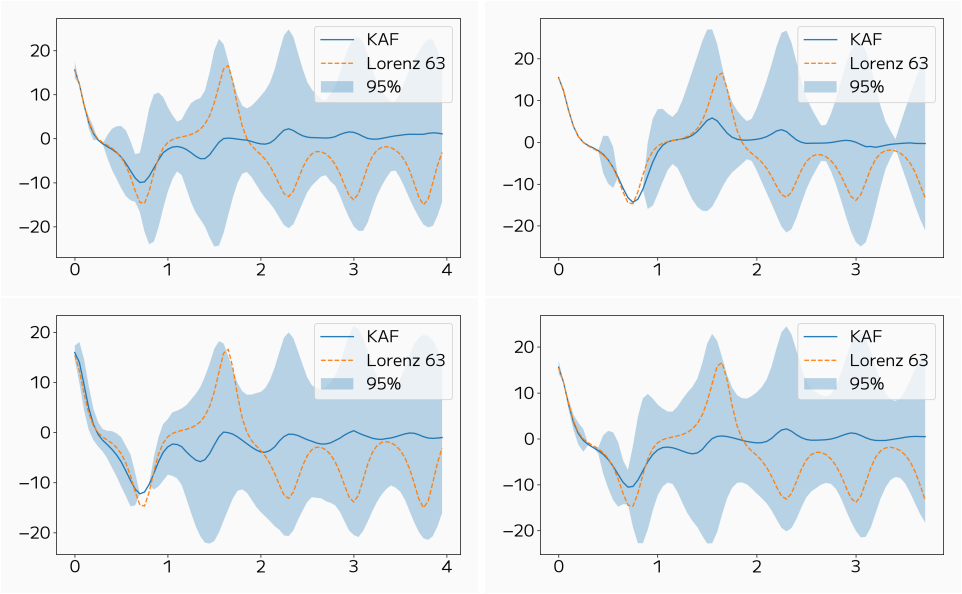


Figure 4.13: Applying KAF to L-63 with delay-embedding dimension 5.

1. rigorous foundation for the use of delay embedding, with two important questions to answer: how to distinguish the cases when it is applicable to the problem at hand, and – in those cases when it is indeed applicable – what the optimal delay-embedding dimension and window are;

2. design of a kernel in the joint $\mathcal{X} \times \Lambda$ space which could exploit the geometry of the family of attractors;

3. development of a hybrid approach, when part of the data set is given with explicit parameter values, and part is given without such values.

As mentioned in the introduction to the chapter, our methodology relies on the assumption that dynamics, and the attractor when one exists, vary smoothly with parameter. And even though the L-63 experiments included bifurcations, they were not significant enough to rule out useful application of the KAF, but would merely deteriorate the forecasting skill. However, we expect that one can easily find an example of parametric dynamics which would render our approach useless. A likely setting, which arises frequently, is where the global attractor is only upper semi-continuous [103]; this happens, for example, if one of the lobes of a chaotic attractor disappears. An entirely new way of dealing with such situations would be needed.

Part II

DATA-DRIVEN MODEL AUGMENTATION

In the second part of the thesis, we consider two particular examples of model augmentation via data gathered from dynamical systems. The first is a problem of finding closures for multiscale systems. Here we propose several approaches and demonstrate them on a few examples. The second is a case of applying data assimilation techniques to epidemiological models on graphs (motivated by the COVID-19 pandemic).

CHAPTER 5

CLOSURES FOR MULTISCALE SYSTEMS USING KERNEL METHODS

One of the main challenges that arises in forecasting multiscale dynamical systems is dealing with numerical integration with vastly different time scales. The size of a time-step required for stable and accurate integration is controlled by the fastest process exhibited by a system. However, often one is only interested in the macro-scale processes, for which the said time-step may be set to orders of magnitude larger. As a result, numerical investigation of such dynamical systems is constrained by the smallest time-step needed to resolve the fast process, even though there is little practical benefit from obtaining accurate trajectories of the fast process.

A different approach is to try and construct a *closure* for the fast process' contribution to the behaviour of the slow process. Consider the following ordinary differential equation (ODE):

$$\begin{aligned} \dot{x} &= f_1(x) + f_2(y), \\ \dot{y} &= \frac{1}{\varepsilon} g(x, y), \end{aligned} \tag{5.1}$$

where x and y are some vectors, and $\varepsilon \ll 1$. Deriving a closure then constitutes finding another function such that the x-subsystem can be integrated in time independently:

$$\dot{X} = f_1(X) + C(X), \tag{5.2}$$

with $X \approx x$ in some sense. A typical objective in obtaining the closure C is to match statistics of the resulting system $X(t)$ with the true statistics of $x(t)$. These questions have been extensively studied in the literature [77].

This chapter is devoted to the possibility of learning such closures in a data-driven way using kernel methods. It also serves as a bridge between first and second parts of the thesis, as it involves both data-driven forecasting with kernel methods and obtaining kernel-derived closures. In Section 5.1, we introduce the model problem used for studying closures, and outline several strategies of constructing kernel closures. In Section 5.2, we test the forecasting ability of the proposed methods numerically, and compare them to other closure methods, along with the pure data-driven forecasting.

5.1 KERNEL-BASED CLOSURES

A natural setting for seeking closures is provided by a linear variant of the model (5.1) that exhibits averaging:

$$\begin{aligned} \dot{x} &= f_1(x) + By, \\ \dot{y} &= \frac{1}{\varepsilon} g(x, y), \end{aligned} \tag{5.3}$$

where now B is linear, and invoking the averaging principle, for a sufficiently small ε, we may write $X \approx x$ with

$$\begin{aligned} \dot{X} &= f_1(X) + C(X), \\ C(X) &= \int_{\mathcal{Y}} By \, d(\mu_x y), \end{aligned} \tag{5.4}$$

and μ_x is the invariant measure of the y variable, with a frozen x. These assumptions guarantee existence of such closure, and therefore justify the data-driven approach to finding one. Details of the underlying theory may be found in Pavliotis and Stuart [77].

5.1.1 *The Model*

In this chapter we focus our attention on a chaotic dynamical system, colloquially known as Lorenz '96 multiscale [66], which we will simply abbreviate to L-96. Following the notation established in [33], the L-96 equations model K slow variables $\{x_k\}_{k=1}^{K}$ coupled to JK fast variables $\{y_{j,k}\}_{j,k=1,1}^{J,K}$ with evolution given as follows:

$$\begin{aligned} \dot{x}_k &= -x_{k-1}(x_{k-2} - x_{k+1}) - x_k + F_x + \frac{h_x}{J} \sum_{j=1}^{J} y_{j,k}, \\ \dot{y}_{j,k} &= \frac{1}{\varepsilon}\Big(-y_{j+1,k}(y_{j+2,k} - y_{j-1,k}) - y_{j,k} + h_y x_k\Big), \\ & x_{k+K} = x_k, \quad y_{j,k+K} = y_{j,k}, \quad y_{j+J,k} = y_{j,k+1}. \end{aligned} \tag{5.5}$$

This is of the form (5.3). Here k ranges from 1 to K and j ranges from 1 to J, thus, there are $(J+1)K$ equations in total. The periodic boundary conditions link x and y variables in such a way that if we were to represent coupling between the variables they would form two circles (Figure 5.1).

On the assumption that the y-variables, with x frozen, are ergodic, the averaging principle shows the existence of a function $C\colon \mathbb{R}^K \to \mathbb{R}^K$ such that, for small ε, the x variables are approximated by $X = (X_1, \ldots, X_k)$ solving

$$\begin{aligned} \dot{X}_k &= -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F_x + \frac{h_x}{J} C_k(X), \quad k = 1 \,..\, K, \\ & X_{k+K} = X_k, \end{aligned} \tag{5.6}$$

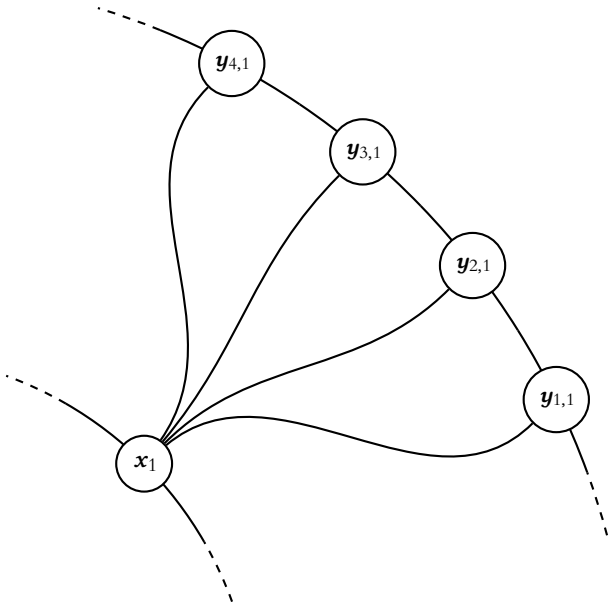


Figure 5.1: Coupling diagram for L-96.

Each line represents a mutual dependence of the connected variables. Dependence of x on y means that y appears in the right-hand side of an ODE for $\dot{x}$. Each variable also depends on itself, but for clarity we do not picture those.

with the same periodic boundary conditions as before, and $C_k\colon \mathbb{R}^K \to \mathbb{R}$ denoting the k^{th} component of a vector-valued function C. This system is of the form (5.4). Since system (5.5) is index-shift-invariant, it is clear that the closure C_k, if it exists, satisfies $C_{k+1}(X) = C_k(\pi X)$ where π shifts the vector indices by adding one unit, invoking periodicity at the end points. Furthermore, when J is large, empirical evidence [33, 115] suggests that there is a function $c\colon \mathbb{R} \to \mathbb{R}$ such that the approximation $C_k(X) = c(X_k)$ is a good one.

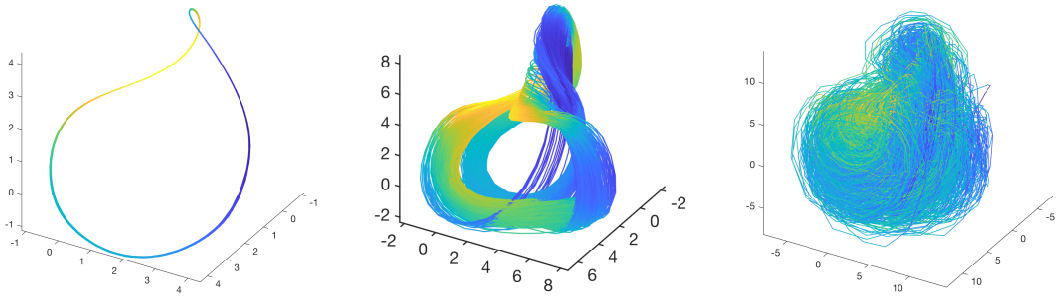
For the numerics that follow a key point to appreciate is that for small ε the variables x in (5.5) exhibit (approximately) Markovian behavior, and this behavior is deterministic and governed by X. However, by tuning F_x, different responses arise in the deterministic variable. In the following we fix parameters ε^{-1}, K , J, h_x, h_y throughout all our experiments as follows:

$$\varepsilon^{-1} = 128,\ K = 9,\ J = 8,\ h_x = -0.8,\ h_y = 1.0. \tag{5.7}$$

We then choose F_x as a bifurcation parameter, and distinguish three cases as follows:

periodic	quasiperiodic	chaotic
$F_x = 5.0$,	$F_x = 6.9$,	$F_x = 10.0$.

Figure 5.2 demonstrates the three responses within system (5.5) resulting from these parameter choices.



(a) Periodic (b) Quasiperiodic (c) Chaotic

Figure 5.2: L-96 regimes of increasing complexity (left to right).
Phase portraits show (x_1, x_2, x_3) coordinates shaded by x_4. The parameter F_x takes values 5.0, 6.9 and 10.0 respectively, from left to right, and all other parameters are as in (5.7).

5.1.2 *Closure designs*

Here we introduce three approaches to constructing a closure. All three have a common theme of collecting data from numerical integration of some ODE and using it to learn the closure $c\colon \mathbb{R} \to \mathbb{R}$. Once the closure is obtained, it is used for numerical integration of the closed system.

As discussed above, for L-96 we seek the closure c in the form of a function taking X_k as input and mapping it to an approximation of $\sum_{j=1}^{J} y_{j,k}$, which can then be multiplied by $h_x J^{-1}$ to obtain the forcing term. Thus, in each case the training data comes in the following form:

$$\left\{\left(x_k(t_i), \sum_{j=1}^{J} y_{j,k}(t_i)\right)\right\}, \quad k = 1\,..\,K. \tag{5.8}$$

Note that index-shift invariance allows us to collect K data point pairs in one integration step.

Since numerical evidence suggests that L-96 is ergodic [33], it is important to take a constant time-step Δt for numerical integration, as it will guarantee sampling from the ergodic measure. In practice, we use Runge–Kutta 4(5) scheme (from the `Python` library `SciPy` [109]), with adaptive step size, however, interpolation of 5^{th} order is then used to reconstruct the solution at Δt-apart points.

In most cases, the amount of data points collected is too big for computationally-efficient use of a kernel method, and is overwhelming for a learning a one-dimensional function, so we choose a subset of training data uniformly at random to obtain 500–800 points (Figure 5.3). Since we assume a Markovian form of the closure (i. e. it only depends on the current value of the variable), we do not need to keep track of the ordering of the points.

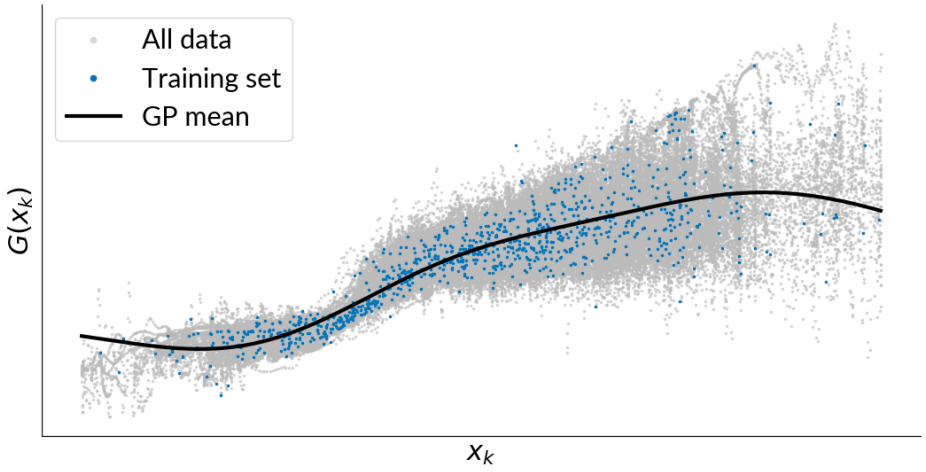


Figure 5.3: Gaussian Process Regression.

As our closure model, we use a standard GP regression with the following kernel:

$$k(x, y) = A \exp\left(-\frac{\|x - y\|^2}{2\delta^2}\right) + \alpha,$$

where $A > 0$ is a magnitude pararmeter, $\delta > 0$ is a length-scale parameter, and $\alpha > 0$ is a white-noise parameter. The parameters' initial values, bounds and typical posterior values are outlined in Table 5.1.

Parameter	Initial values	Bounds	Typical Posterior
A	1	$[10^{-5}, 10^{5}]$	0.8, 1.1
δ	3	$[10^{-10}, 10^{6}]$	0.01, 0.03
α	1	$[10^{-10}, 10^{5}]$	0.1, 0.5

Table 5.1: GP parameter bounds, initial and typical posterior values.

The GP regression is carried out using `Python` library `Scikit-learn` [79]. It implements Algorithm 2.1 of Rasmussen and Williams [85] to fit the GP to the data, which also outputs log-marginal likelihood; the latter is then used to tune the parameters via `L-BFGS-B` routine [22, 121], as implemented in `SciPy` [109].

After tuning parameters and fitting a GP to the data, we use the posterior GP's mean as the closure c in (5.6). As our benchmark, we integrate this system numerically, and compare the density of X_k to the density of x_k of the original system (5.5). The same remark about constant step-size applies here; furthermore,

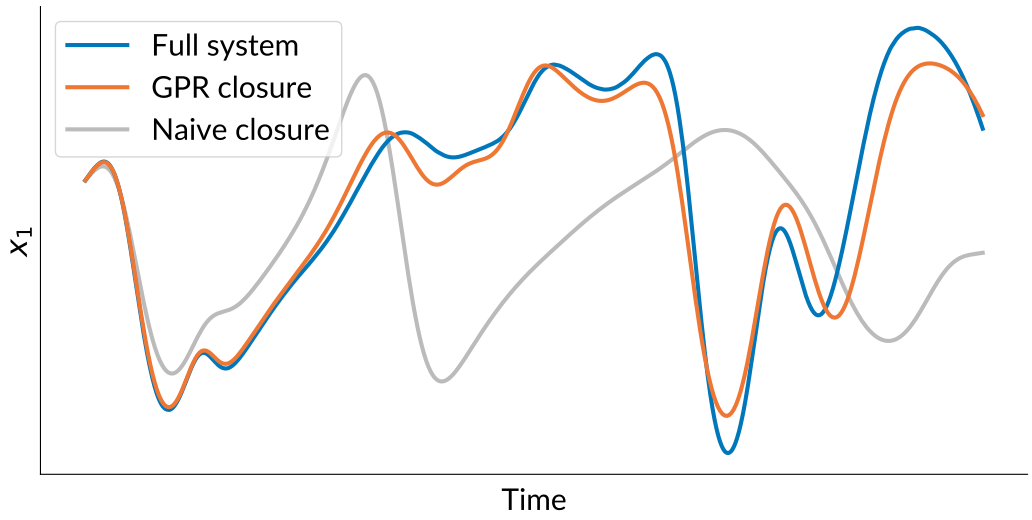


Figure 5.4: L-96 numerical simulations.

Depicted in this plot are sample numerical simulations of a full L-96 system (5.5) (blue), closed system (5.6) with an offline closure, and closed system (5.6) with a naive closure $c(X_k) = h_y J X_k$.

since both systems are index-shift invariant, the density of X_k is the same for any $k = 1..K$, which again allows us to build empirical densities faster. We use kernel density estimation for comparison and plotting (see Figure 5.5 for an example).

Offline

The most staightforwad approach, which we call *offline*, is to first obtain data pairs of the form (5.8), for example, by numerically integrating the full model (5.5) for a short period of time, and then use them to perform the GP regression with parameter tuning.

Filtered

The second approach, called *filtered*, starts off exactly like the offline one, but instead of using the closed model (5.6) with learned closure c, we leave one section of the fast variables to provide behavior that is close to the real system (5.5):

$$\begin{aligned}
\dot{X}_k &= -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F_x + \frac{h_x}{J} c(X_k), \quad k = 2..K, \\
\dot{X}_1 &= -X_K(X_{K-1} - X_2) - X_1 + F_x + \frac{h_x}{J} \sum_{j=1}^{J} y_j, \\
\dot{y}_j &= \frac{1}{\varepsilon}\Big(-y_{j+1}(y_{j+2} - y_{j-1}) - y_j + h_y X_1\Big), \\
X_{k+K} &= X_k, \quad y_{j+J} = y_j.
\end{aligned} \tag{5.9}$$

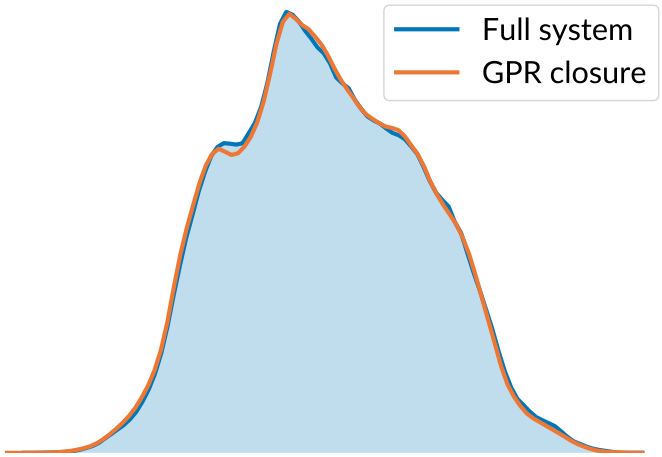


Figure 5.5: Probability density functions of L-96 and the system with offline closure.

Note that since we only have one section of the fast variables, we modify the periodic boundary condition for y to make it loop onto itself.

This approach allows for a data assimilation algorithm to be run in parallel, in case data from a real-world system is gathered for the fast variables. This also allows one to investigate the significance of any particular subset of the fast variables and, potentially, answer some optimal design questions.

Online

Finally, the *online* approach consists of many small iterations of building the closure. Here we will have a series of closures c_i instead of one. We first must choose an initial closure c_0; for L-96, we would simply choose a linear closure $c_0(X) = h_x h_y$, which can be obtained from the fast subsystem by setting all $y_{j,k} \equiv$ const and equal to each other. After a short period of time, the data produced in such fashion is then used to train a GP regressor, and a new closure $c_1(X)$ is set to be the mean of that regressor. The process then continues, and one could stop when c_i and c_{i+1} are close in some norm.

5.2 FORECASTING COMPARISONS

5.2.1 *Conditional Expectation and Variance*

We aim to predict the x_1 variable from historical data of a long trajectory of x alone. Thus the observation and observable maps are $\Pi(\omega) = x$, $F(\omega) = x_1$. We will also use $F(\omega) = x_1^2$ when estimating conditional variance. By tuning the scalar parameter F_x (not to be confused with function F) as outlined in the

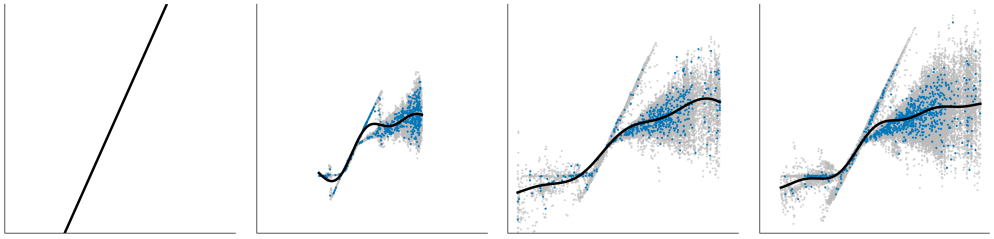


Figure 5.6: Online GPR iterations: $i = 0$ through $i = 3$.

preceding subsection we can obtain periodic, quasiperiodic and chaotic responses in the averaged variable X. It is intuitive that the ability of the KAF to track the true trajectory of the slow variables decreases with increasing complexity; in other words, predictions in the periodic case should be the most accurate whilst those in the chaotic case present a significant challenge. In the experiments that follow the size and sampling interval of the source (training) data remain fixed at $(40000, 0.05)$ and the out-of-sample (test) data set is fixed at $\hat{N} = 7000$.

The space of observables $\mathfrak{X}$ in the current example is the space of all slow variables. Since, under the small-ε limit, an ODE closure of the slow dynamics is obtained, the variable x behaves (approximately) like a deterministic Markov process, and the expectation in (2.6) disappears; the predictor is expected to track the actual trajectory $x_1(t)$. To see this another way, note that simply knowing the initial values of the x-variables (recall that $\mathfrak{X}$ is precisely all x-variables) and the closure $C(X)$ in equation (5.6), we are able to predict x_1 (or indeed, any x_k) exactly, given the initial conditions for all x-variables.

However this picture is greatly affected by the sensitivity of the system to initial conditions and sampling errors due to high dimensionality of the attractor. We now describe how these predictions work in practice, in the three regimes shown in Figure 5.2. We display our results in Figure 5.7, where x_1 and standard deviation bands are predicted and compared with the true signal starting from the same point. The long-term predictability in each regime is constrained by the complexity of the underlying Markovian, deterministic, slow dynamics. In the periodic regime, since chaos is absent in the slow variables, a perfect predictor is obtained via the partially observed dynamics; one interpretation of why this occurs is because the eigenfunctions of the Koopman operator lie in a finite span of the diffusion coordinate observables [9]. Observe that Z_τ remains in phase, and the forecast variance is negligible, for long lead times up to the length of the entire out-of-sample trajectory ($\tau = 350$). The quasiperiodic trajectory is tracked imperfectly, but with significant accuracy over the same range of times; errors are visible mainly around the extrema of x_1 as suggested by the phase portrait; the conditional variance reflects the significant accuracy present. Prediction in the fully chaotic regime only tracks the trajectory, however, until a lead time of approximately 1 time unit, exhibiting behaviour at long lead times

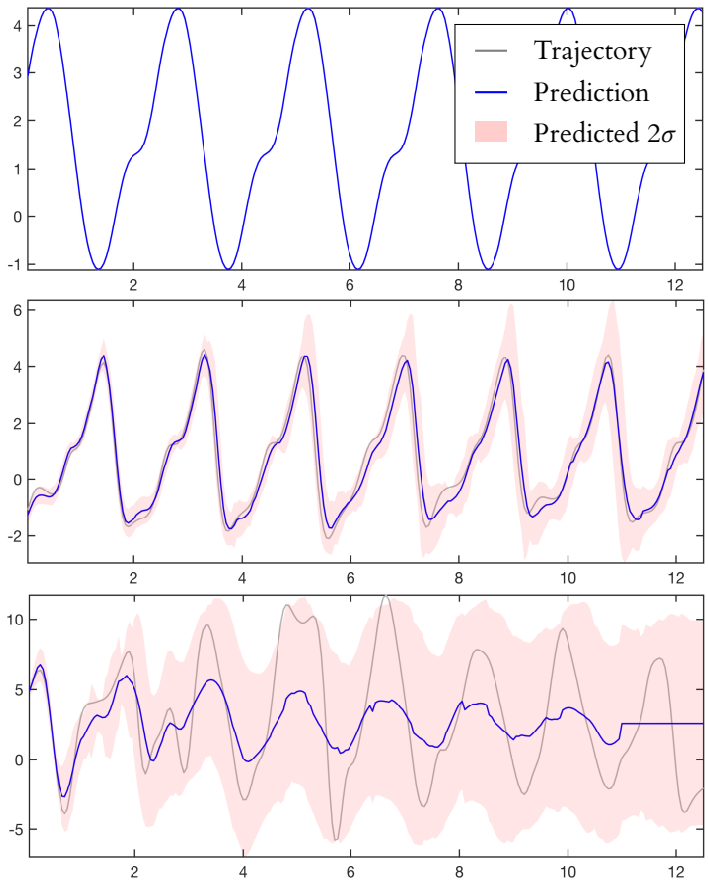


Figure 5.7: Predictability: periodic, quasiperiodic and chaotic regimes.

Prediction $Z_\tau(x)$ of observable $F(\omega) = x_1$ across 3 different regimes. In each figure grey is the true trajectory, blue the predictor using KAF, and pink gives two standard deviations confidence bands, computed using the conditional variance. The parameter F_x takes values 5.0, 6.9 and 10.0 respectively, from top to bottom, and all other parameters are as in (5.7). In the first, periodic response regime, the trajectory is predicted almost perfectly and this accuracy is reflected in the narrow confidence bands. In the second, quasiperiodic response regime, the trajectory is predicted very well, but with growing error reflected accurately in the slowly growing confidence bands. In the third, chaotic response regime, the predictive capability is lost due to sensitivity to initial conditions and this is reflected in the rapidly growing confidence bands and in the convergence of the predictor to a constant, for large τ.

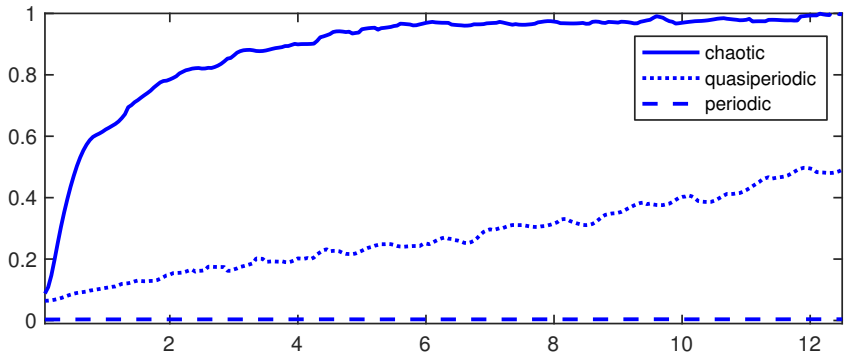


Figure 5.8: RMSE.

This figure depicts the RMSE of the predictor $Z_\tau(x)$ for (5.5), for different F_x, as a function of τ. The parameter F_x takes values 5.0, 7.1 and 10.0 respectively, from smaller to larger error, corresponding to periodic, quasiperiodic and chaotic response; all other parameters are as in (5.7).

which is somewhat similar to that seen in the previous, homogenization, section in which the predicted variable behaved as if drawn from a Markov stochastic process. In particular the long-term predictor in the chaotic regime converges to a constant by construction, assuming mixing, and this is consistent with the inherent unpredictability of chaotic dynamics. It is notable that the size of the conditional variance, and the resulting confidence bands, is a useful guideline as to the pathwise accuracy of the data-driven predictor. The observations about the predictability of the system by KAF methods are also manifest in Figure 5.8 which shows the RMSE in each of the periodic, quasiperiodic and chaotic regimes.

We mention that in the quasiperiodic case the presence of multiple attractors (or multiple lobes of the same attractor), and resulting intermittent switching between these attractors, leads to a loss of predictability that is significant on time-scales much longer than those shown here. For the figure shown here we have ensured that training points and out-of-sample points are gathered from the same (part of the) attractor to maintain accuracy. We train using two different trajectories to gather ample training data.

Recall that at each lead time τ along the horizontal axis there is a potentially different number of eigenfunctions $\ell(\tau)$ used in the data-driven method. In the chaotic regime the optimal $\ell(\tau)$ tends to 1 for large times whilst ℓ fluctuates around 50 in the quasiperiodic regime; we obtain $\ell \approx 9$ for all τ in the periodic regime.

5.2.2 *Comparison Of Data-Driven And Model-Data-Driven Prediction*

The previous subsection concerned purely data-driven prediction of variable x from (A), using only data in the form of a time-series for x. In this subsection we

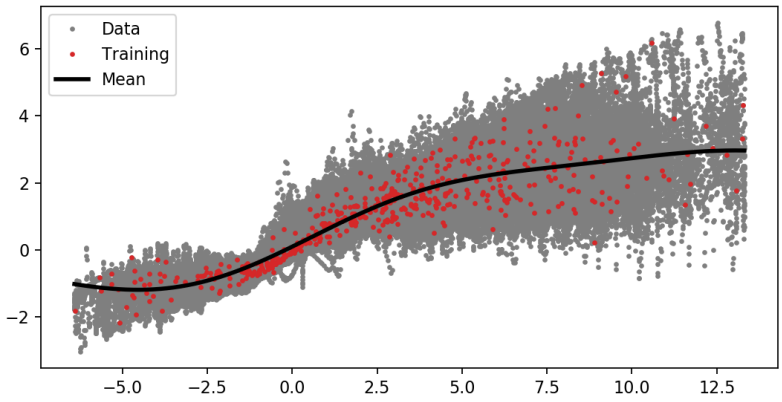


Figure 5.9: Mean of Gaussian process regression as a closure.

Function c_{GP}, and data used to determine it, from data generated by (5.5) with parameters as in (5.7) and $F_x = 10.0$.

provide comparison with a different forecasting technique based on a combination of model and data-driven prediction, using data in the form of a time-series for (x, By). Knowledge of By enables the use of *Gaussian process regression* (GPR) [85] to approximate $v(\cdot)$ by $v_{GP}(\cdot)$ in (A0). Our approach is motivated by the paper [33] which looked at finding such closures for the L-96 model in form (5.5). When applied to (5.5) the methodology leads to an approximate closure for the slow variable X which takes the form

$$\dot{X}_k = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F_x + h_x c_{GP}(X_k), \quad k \in \{1, \ldots, K\}, \tag{5.10}$$

subject to periodic boundary conditions $X_{k+K} = X_k$. This should be compared with (5.6), which arises from application of the averaging principle; note that, in addition, we have invoked the hypothesis that $C_k(X)$ can be well-approximated by function of $c(X_k)$, as discussed directly after (5.6); and we will determine an approximation c_{GP} for c by GPR.

Explicit details of the procedure we use to build a GP closure are described in Section 5.1.2; here we observe that for training we use tuples $\{x_k(t_n), (By)_k(t_n)\}_{n=1}^N$, over all $k = 1, \ldots, K$. See Figure 5.9 to see the data used (red random subsamples, without replacement, of the total grey data set), and an approximate GP closure c_{GP} determined from that data.

Once we have the closed model appearing in (5.10) we may use it to predict the variable x appearing in (5.5), and we may compare that prediction with the one made by KAF. Figure 5.10 shows the result of doing so. It shows that the KAF approach is superior in the periodic and quasiperiodic settings, but that for predictions of the trajectory itself the model-data based predictor (5.10) is superior to KAF in the chaotic case. Note that the model-data based predictor

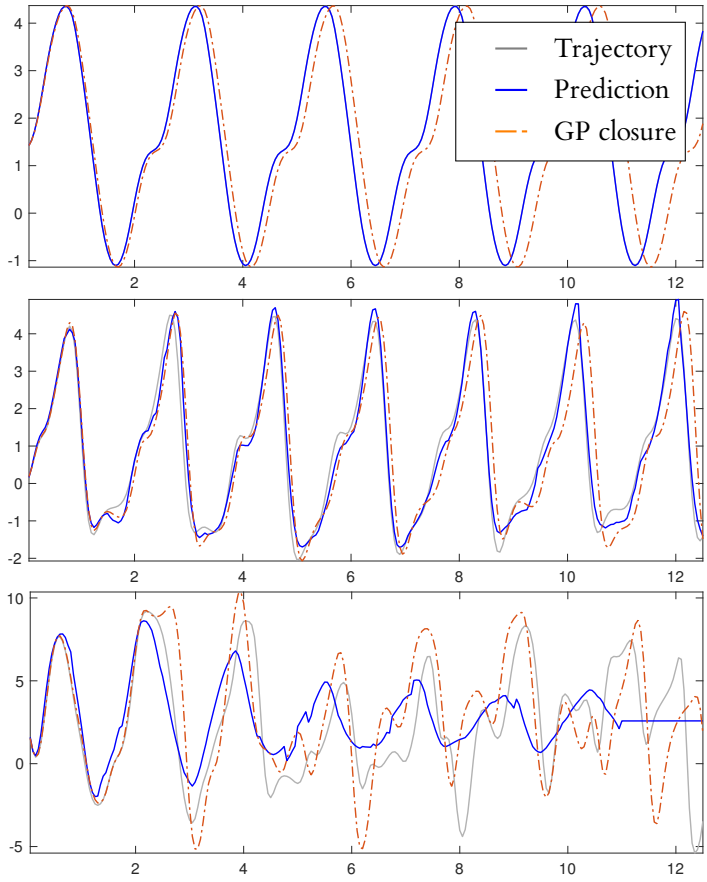


Figure 5.10: Comparison of data-driven and model-data-driven prediction.
The true trajectory is shown in grey, the KAF data-driven prediction in blue and the model-data-driven predictions based on (5.10) in dotted-red; the periodic, quasiperiodic, and chaotic regimes are considered in turn.

has access to more data than does the KAF, and requires model knowledge; the KAF is entirely data-driven.

We now dig a little deeper into the comparison. We do this in a systematic way in the periodic, quasiperiodic and chaotic regimes. In each of these three cases we show four RMSE error curves, labelled as follows: a) the standard KAF based on x data alone; b) an enhanced KAF using (x, By) data, the same data used to train the ODE (5.10); c) a prediction using the ODE (5.10); d) a KAF prediction trained on X data alone, generated by the ODE (5.10). Figure 5.11 shows that KAF a) is the ideal predictor in the periodic regime and is near-ideal in the quasiperiodic regime; on the other hand, the ODE (5.10) predictor c) is ideal for short-term predictability in the chaotic case. Augmenting observations with By within KAF, as in b), gives errors similar to those arising from a), when observing x alone; thus knowledge of By provides little extra information. In the chaotic case, the RMSEs of KAF trained on x, a), and on X, d), are very

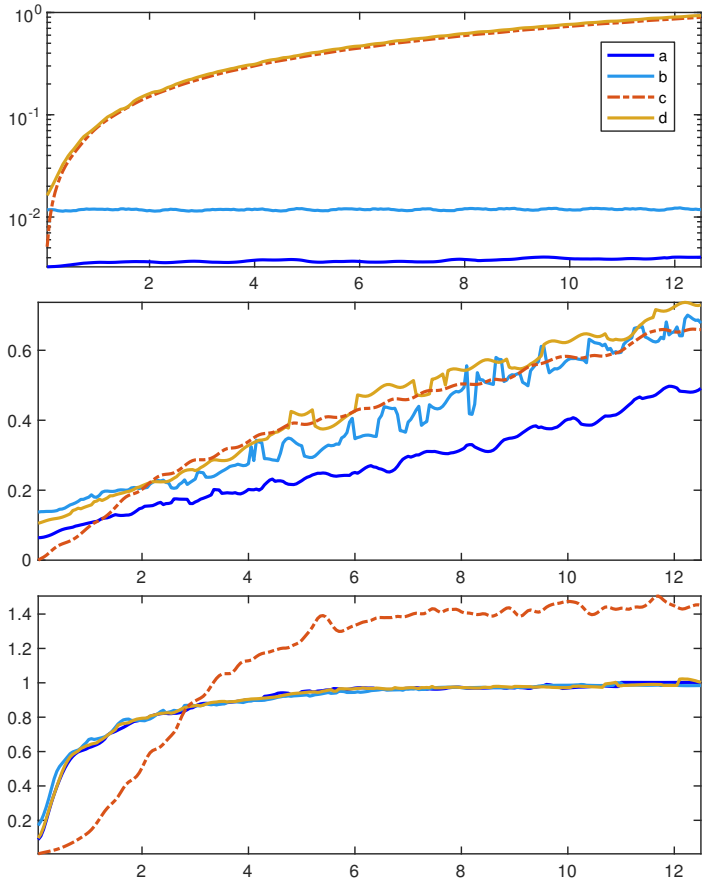


Figure 5.11: RMSE comparison for the four cases a)–d) described in the text, in the periodic, quasiperiodic, and chaotic regimes.

In the periodic regime, KAF (a) is an ideal predictor with negligible growth in error (note the logarithmic scale). In the quasiperiodic response regime, the growth in RMSE with KAF (a,b) is significantly slower than that of the GP-based ODE prediction (c). In the chaotic response regime, the GP-based ODE prediction (c) is more accurate in the near term, yet KAF error stabilizes as the prediction converges to the conditional mean.

close, confirming that the ODE (5.10) for X captures the invariant measure of the approximately Markovian variables x as intended.

We emphasize the difference between averaging and homogenization here: in the averaging case observing the fast variables adds nothing to our prediction because there is a closed system determined only by the slow variables (see Figure 5.11, graphs a) and b) in all three plots). By contrast, in the homogenization case, observing the fast variables improves short-term predictions because it provides further information about the driving stochastic process entering the homogenized limit (see Figure 5.12).

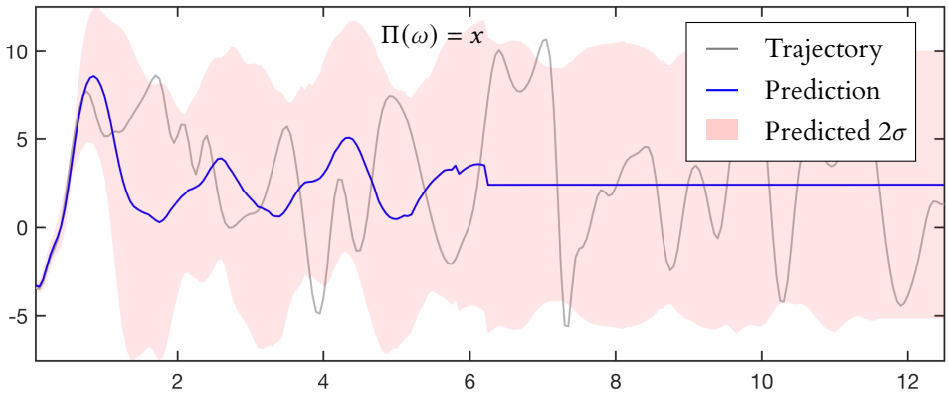


Figure 5.12: Prediction in non-Markovian regime.

Here $F_x = 10$, $\varepsilon = 1$. As expected, non-Markovian regime has a much shorter accurate trajectory predictability, followed by rapid convergence of the conditional mean to a constant. Note, however, that the uncertainty prediction bands contain the true trajectory for all time.

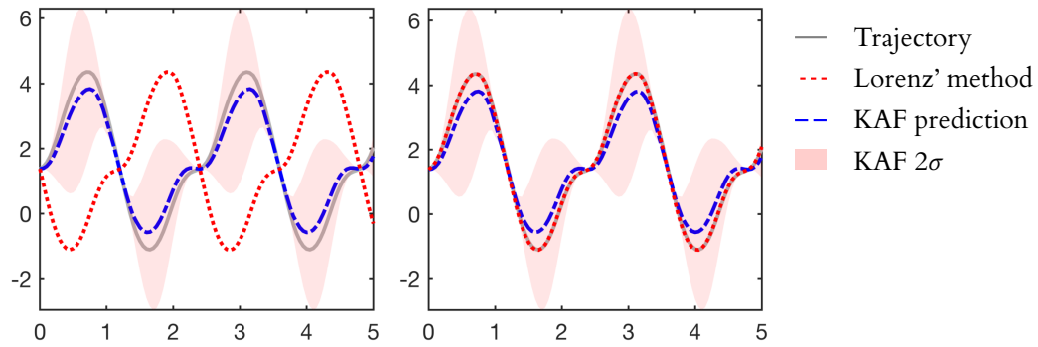
5.2.3 *Non-Markovian regime*

In the preceding subsections we studied predictors for x, based only on time-series data in the x coordinate, for the equation (5.5). We studied the scale-separated regime where $\varepsilon \ll 1$ and x is approximately Markovian and deterministic – it is approximately governed by an ODE. Here we study the behavior of identical predictors when $\varepsilon = 1$; the system (5.5) then no longer exhibits averaging and x is no longer Markovian because there is no scale-separation between x and y. This experiment is conducted with $F_x = 10$. Because of the lack of Markovian behaviour we expect rapid loss of predictability in time, when $\Pi(\omega) = x, F(\omega) = x_1$. The resulting conditional mean and variance, shown in Figure 5.12, confirms this intuition. Indeed the conditional mean is out of phase with the truth at lead time $\tau = 1$, and this is also reflected in the large growth of the conditional variance. Furthermore, the conditional mean tapers to a constant at $\tau = 6$, twice as quickly as it does in the $\varepsilon \ll 1$ setting in which this tapering occurs at $\tau \approx 11$ (Figures 5.7,5.10).

5.2.4 *Comparison with Lorenz' method*

We illustrate the advantage of KAF over Lorenz' original method of analog forecasting (5.11), which can produce predictions that are discontinuous with respect to initial condition. In particular, this occurs when data are partially observed from a larger state space, and different states map to identical partial observations. Recall that the Lorenz method is defined in the following way:

$$Z^p(s) = y^{(p)}_{n^\star(s)}, \tag{5.11}$$



(a) $x_1(0) = 1.3736$ (b) $x_1(0) = 1.3799$

Figure 5.13: Lorenz' method (5.11) vs. KAF.

Here $F_x = 5$, $\varepsilon^{-1} = 128$, with only partial observations: $\Pi(\omega) = x_1$. The sensitivity of Lorenz' method to the initial conditions in this regime results in diverging predicted trajectories for nearby initial conditions, separated only by 0.0063. By contrast, KAF, which is continuous with respect to initial condition, shows moderate predictive skill and makes nearly identical predictions for nearby initial conditions.

where $n^\star(s) = \arg\min_n \|s - x_n\|$.

To study this, we observe a single coordinate x_1 of the periodic regime ($F_x = 5$, $\varepsilon^{-1} = 128$) so that the observed data are highly non-Markovian. We select initial conditions that are $\mathcal{O}(10^{-3})$ apart, but are separated in time by integer multiples of the period. Figure 5.13 plots the resulting predictions from Lorenz' method and KAF. Although Lorenz' method is accurate for one initial condition (right), it gives a diverging prediction for a nearly identical point (left). By contrast, KAF is continuous with respect to initial condition and displays theoretically optimal predictive skill (in an RMSE sense) for even highly non-Markovian observation data. This experiment illuminates a key feature of KAF: that it gives consistent predictions that are continuous with respect to initial conditions. Note, also, that KAF uncertainty predictions of a periodic observable are also periodic, and vanish at every half period when predictions intersect the ground truth.

CHAPTER 6

GRAPH-BASED EPIDEMIOLOGICAL MODELS WITH DATA ASSIMILATION

The study presented in this chapter was primarily carried out during the recent COVID-19 pandemic to investigate the possibility of minimizing lockdown effects via running epidemiological models with data of various kinds gathered in real-time. It was motivated by two factors: on the one hand, the surge of attempts at tackling the spread of the virus using connectivity information available via proximity sensors of mobile devices (in place of, or in addition to, the manual contact tracing) — a technical possibility that was not available during the previous global pandemic of the Spanish flu (1918–1920); and on the other hand, successful applications of data assimilation (DA) techniques in a variety of fields of geophysical sciences, such as numerical weather prediction [50], oceanography, land and ice modeling.

During the COVID-19 pandemic, exposure notification apps have been developed to scale up manual contact tracing. The apps use proximity data from mobile devices to automate notifying direct contacts of an infection source. However, the information they provide is limited because users receive only rare and binary alerts, i. e. they only signal that a contact was made with someone who tested positive for the virus. In this work, the *risk network* model is proposed as a new digital approach to epidemic management and control.

Risk network combines a model of disease transmission with (a) proximity and (b) crowd-sourced health data, in a fashion that closely resembles a data assimilation cycle done in weather forecasting. It then provides frequently updated *individual* risk assessments to users. Epidemic spreading is notorious for its exponential growth, and therefore, creates a great challenge for predictions of any meaningful time window into the future. The risk network model, however, does not forecast the epidemic evolution, but rather focuses on current-time estimations of individuals' probabilities of being infected. In other words, it uses historic data up to the present moment to infer information about the disease spreading over the past few days, and then produce a statistical assessment of individual risks at the moment. This is achieved by using a DA backward-and-forward time integration loop akin to a weather prediction cycle. Thus, risk network uses the same data as exposure notification apps but in a more

efficient way. Implemented at scale, it has the potential to effectively control epidemics while minimizing economic and social disruption, as demonstrated in computational experiments.

The chapter is organized as follows. Section 6.1 serves as a brief introduction to the field of epidemiological modeling and elaborates on relevance of the study in general. Sections 6.2 and 6.3 are devoted to the two components of the risk network model: the former introduces the network model, while the latter describes the proposed data assimilation pipeline. In particular, the network model is comprised of (a) the compartmental epidemiological model (SIR-type), (b) the master and reduced master equations (ODEs for probabilities), (c) the closure of said master equations, (d) and the *contact network*, i. e. a graph with time-dependent edges (connections) between nodes (users). All of these are separated into their respective subsections in Section 6.2. The data assimilation section includes descriptions of the DA algorithm itself, how the two types of data are used, and the medical testing strategies.

Numerical simulation of the proposed model necessarily has to rely on a surrogate model of disease spreading and proximity interactions within the population because obtaining real-world data of either kind is nearly impossible for privacy reasons. Such surrogate model is described in Section 6.4: one subsection is devoted to graph generation, and another one to a Markov chain Monte Carlo simulation of the virus transmission.

Finally, in Section 6.5, we conduct an extensive simulation study of the COVID-19 epidemic in New York City (NYC), and discuss the results. In particular, it is shown that the risk network model with diagnostic testing achieves epidemic control with fewer than half the deaths that occurred during NYC's lockdown, while isolating a far smaller fraction of the population (typically only 5–10% of the population at any given time).

This work was published in a peer-reviewed journal [93]. All code written in support of this chapter is publicly available at:

`https://github.com/tapios/risk-networks`

Contributions of the author

D. B. has contributed to (a) the design of the closure of master equations, in particular, devising and numerically testing a handful of candidate closures (Section 6.2.3); (b) design, testing and implementation of multiple DA algorithms and techniques: backward-and-forward time integration loop, sparsification of the covariance matrices for faster computational runtimes (Section 6.3.1); medical testing strategies (Section 6.3.2); intervention strategies (Section 6.3.6); and classification rules (Sections 6.3.4 and 6.3.5); (c) implementation and testing of a large portion of the codebase; and (d) a great number of numerical runs on small-sized networks. The surrogate model described in Section 6.4 is provided

in this chapter for the full scientific picture of the study. Likewise, the conclusive experiments that were run on large-sized networks in Section 6.5 are provided for the scientific richness.

6.1 INTRODUCTION

Until a majority of the global population reaches immunity against continuously evolving virus variants through vaccination or infection, any future epidemics will need to be fought with non-pharmaceutical interventions (NPIs) [17, 45] — as was the case with the past COVID-19 pandemic. They include social distancing, mask usage, and restrictions of mass gatherings. But NPIs such as lockdowns come at catastrophic costs to individuals, economies, and societies, with disproportionate burdens carried by disadvantaged groups [25, 54]. Even if imposed only intermittently and regionally, lockdowns are an inefficient means of epidemic management and control: they isolate much of the population, although even at extreme epidemic peaks, only a small fraction is infectious [53, 81]. If individuals who are at high risk of being infectious could be identified before they infect others, control measures could be made more efficient by targeting them to this high-risk group.

Testing and contact tracing have been discussed and partly implemented as strategies to identify individuals who are at high risk of being infectious [36, 47, 55]: testing determines who is infectious, contact tracing identifies those who may have been exposed through contact with an infectious individual, and this high-risk group is then isolated. However, controlling the COVID-19 epidemic by *testing, contact tracing, and isolation* (TTI) had been complicated by frequent asymptomatic and presymptomatic transmission, which support silent spread, and a short serial interval, the period between the onset of any symptoms in infector and infectee [46, 47, 60, 78]. Even in ideal scenarios, contact tracing needs to identify upward of 75% of infections to achieve epidemic control [36, 78]. Quickly diagnosing such a large fraction of infections and manually identifying exposed individuals requires testing and a contact tracing workforce at a scale that has been challenging to realize in most countries [3, 114].

To scale up the contact tracing component of TTI without a massive expansion of the workforce, exposure notification apps had been developed. They rely on proximity data from smartphones or other mobile devices to identify close contacts between users [8, 23]. If an individual user is identified as being infectious, prior close contacts are notified and can then self-isolate. The exposure notification is deterministic (a user is only notified when potentially exposed), and it only uses nearest-neighbor information on the network of close contacts among users. Exposure notification apps have not seen widespread use, in part perhaps because of early implementation difficulties and privacy concerns but also because they do not provide users with information except in the rare case

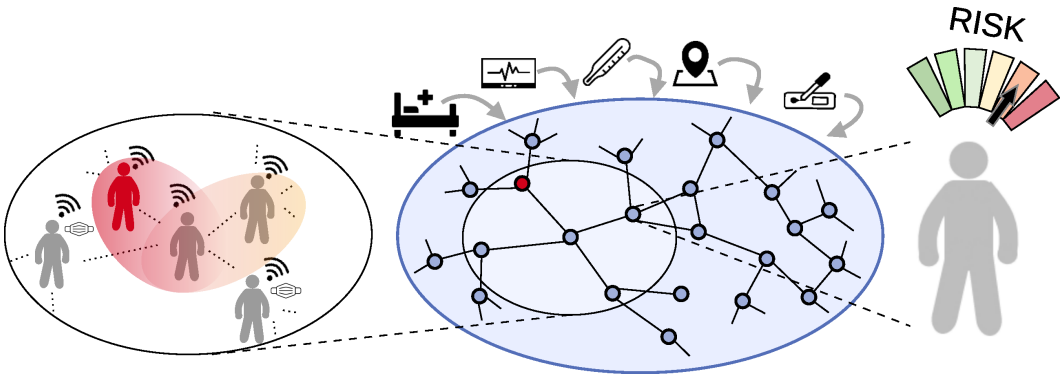


Figure 6.1: Schematic of the personalized risk assessment platform.
Proximity-tracking data from mobile devices is used to assemble a contact network, in which nodes represent individuals and edges represent close contacts between individuals. An epidemiological model defined on the contact network is then fused with diverse health data, including diagnostic tests, hospitalization status, and possibly data such as body temperature readings. The model spreads risk of infectiousness from a positive individual (red) to others, taking into account knowledge about disease progression, the time and duration of contacts, and the use of personal protective equipment (PPE), among other factors. The result of the network DA is an assessment of individual risks, for example, of being infectious, which then can be used to target contact interventions.

when they receive an exposure notification [59]. Nonetheless, where they have been used, these apps have helped prevent the spread of infections [117].

Finally, one prominent approach that stands apart from contact tracing and exposure notification apps is NOVID: a notification framework (implemented as a mobile app for iOS and Android devices) that informs a user about how many connections, or degrees, apart an exposure or infection occured [32, 63]. It uses, in principle, the same two types of anonymous data as the risk network, namely, proximity data to build a graph of users and their connections, and self-reported diagnostic data to register symptomatic, exposed, confirmed cases, and vaccinated users. However, the main difference between NOVID and our approach is that the former lacks any model of disease transmission, thus being unable to predict probabilities of being exposed, leaving the user to guess them based on the propagation "speed" through one's circles of connections (10^{th} degree circle, 9^{th} degree circle etc.). Furthermore, having an epidemiological model as a basis for DA permits one to propagate exposure probabilities even to and from external nodes, i. e. the ones that represent non-users in the general population, and also provide uncertainty quantification.

This study presents a new and more effective way of exploiting the same information on which exposure notification apps rely. Unlike these apps, however, this method provides users with continuously updated assessments of their individual risks. The core idea is to learn about individual risks of exposure and

infectiousness by propagating crowdsourced information about infection risks over a dynamic contact network assembled from proximity data from mobile devices. Instead of the deterministic assessments of exposure notification apps, our approach exploits data from diverse sources probabilistically. Various types of information, including their uncertainties, can be harnessed. For example:

- Diagnostic tests, including sensitive but slow molecular tests, less sensitive but rapid antigen tests, or pooled diagnostic tests [97].
- Serological tests, which indicate a reduced probability of susceptibility when antibodies specific to SARS-CoV-2 (or the causative agent of another targeted disease) are detected.
- Self-reported clinical symptoms, elevated body temperature readings, or other wearable sensor data, which can indicate an elevated probability of infectiousness and virus transmission [16, 83].

Quantification of individual risks is achieved by assimilating data into a model of virus transmission and disease progression. The model represents individual's probabilities via a system of ODEs, where the rates of virus transmission between the nodes, i. e. users, are determined by a dynamic contact network. This network, in turn, is represented by a graph whose edges are assembled from proximity data.

For decision making, periodically updated individual risks of having been exposed or of being infectious take the place of the deterministic assessments in exposure notification apps. The probabilistic network approach propagates data farther along the contact network than contact tracing, consistent with models of disease progression and rates of virus transmission. It harnesses more information than contact tracing, both by being able to include diverse data sources with their uncertainties and by exploiting information inherent in the network structure itself: an individual with many contacts generally is at greater risk of having been exposed than an individual with fewer contacts [67, 75], and such contact rates are available from the proximity data from mobile devices.

The network and the information it contains are dynamically updated in periodic data assimilation (DA) cycles. These cycles resemble the daily DA cycles that weather forecasting centers use operationally [11]. The quantitative information that is provided by the risk network can be used in similar ways as weather forecasts: to inform personal decisions by users based on their desire to avoid risk (in the weather forecasting analogy, staying home rather than going on a mountain in the face of a likely downpour) and to inform public policy when aggregate risk measures indicate that wider mandates are necessary (analogous to evacuating a city to protect lives and avoid overwhelming public health and social infrastructures when a hurricane is likely to make landfall).

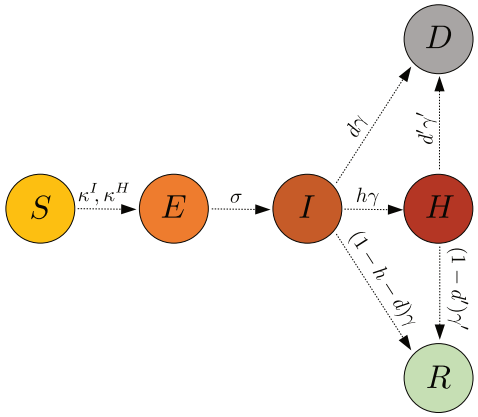


Figure 6.2: Schematic of SEIHRD model [10].
Infected and hospitalized nodes infect susceptible nodes at rates κ^I and κ^H, respectively. After being infected, susceptible nodes become exposed. Exposed nodes become infectious at rate σ. Infected nodes may get hospitalized at rate $h\gamma$, die at rate $d\gamma$, or become resistant at rate $(1-h-d)\gamma$. Once hospitalized, nodes either become resistant at rate $(1-d')\gamma'$ or die at rate $d'\gamma'$.

6.2 NETWORK MODEL AND EQUATIONS

Our point of departure is a variant of the widely used susceptible–exposed–infectious–resistant (or recovered) (SEIR) model of epidemiology, extended through inclusion of hospitalized (H) and deceased (D) compartments to an SEIHRD model [10]. Compartmental epidemiological models have traditionally been applied on the level of aggregated individuals (e. g., the population of a city or country) [14]; here we follow more recent work and apply the SEIHRD on an individual level on a time-dependent contact network [52, 75]. Thus, instead of having 5 ODEs stemming from the mean-field approximation (the number of states, in this case, six, minus one because fractions of the population in each state must sum to one), we work with $5N$ equations, where N is the number of individuals.

For ease of exposition, we provide here a general form of the reduced master equations with three possible states X, Y and Z [96] (the exact form is provided in Section 6.2.2):

$$\begin{aligned}\langle \dot{X}_i \rangle &= R_i^{yx}\langle Y_i\rangle + R_i^{zx}\langle Z_i\rangle - (R_i^{xy} + R_i^{xz})\langle X_i\rangle,\\ \langle \dot{Y}_i \rangle &= R_i^{xy}\langle X_i\rangle + R_i^{zy}\langle Z_i\rangle - (R_i^{yx} + R_i^{yz})\langle Y_i\rangle,\end{aligned} \tag{6.1}$$

where $i = 1, \ldots, N$, and $\langle\cdot\rangle$ denotes probability of being in that state. Here R_i denotes the matrix of transition rates between states X, Y and Z for the i^{th} individual. These matrices can depend on other individual's states, more specifically in the case of SEIHRD model, one individual can become infected

only through coming in contact with some other infectious individual. This is where the information from the contact network comes in: in short, the duration of a contact between two individuals influences transition rates.

In the contact network, each individual is represented by a node on the network; time-dependent edges between the nodes are established by close contacts between individuals, as recorded by proximity data from mobile devices. Virus transmission can occur during close contacts from infectious or hospitalized nodes to susceptible nodes, which thereupon become exposed (Figure 6.2). The probability of transmission increases with contact duration, and the transmission rate can vary from node to node and with time, for example, to reflect time-varying transmission rates resulting from virus mutations or a reduced transmission rate when masks are worn. From being exposed, nodes progress to becoming infectious, and later they may progress to requiring hospitalization, recover, or die.

In real world, at any time t, each node i is in one of the six health and vital states $S_i(t)$, $E_i(t)$ etc. of the SEIHRD model. The risk network learns about the probabilities $\langle S_i(t)\rangle$, $\langle E_i(t)\rangle$, etc. of finding an individual node i at time t in each of the different states. We adopt a sequential Bayesian learning approach that propagates an ensemble of individual probabilities $\langle S_i(t)\rangle$, $\langle E_i(t)\rangle$, etc. across the network and periodically updates them and the SEIHRD model parameters with new data [5, 60, 80, 95]. Data falling within a DA window of length Δ (typically, $\Delta \approx 1$ day) are incorporated into the model by adjusting the ensemble to minimize the misfit to the data in the window. An interval Δ later, the updating procedure is repeated. Such DA cycles and the underlying algorithms are used daily in weather forecasting to estimate up to 10^9 variables characterizing the state of the atmosphere; they easily scale to network epidemiology models with millions of nodes or more. Essentially all types of data and their error characteristics can be assimilated with this approach, even data that are less sensitive to infectiousness, such as readings of heart rates [84] or body temperatures [16, 83] (Fig. 6.1).

6.2.1 *SEIHRD model on a contact network*

We first need to derive a Markov chain-type model that can only be in a finite number of states at any given moment. Equations that govern such models are called *master equations*. However, in practice numerically integrating the resulting system of ODEs is infeasible for any large enough number of individuals N, as the number of equations is then $6^N - 1$. What is known as the *reduced master equations* are then derived, with the number of equations equal to $5N$. This section introduces the Markov chain model, whereas Section 6.2.2 presents the reduced master equations.

We consider a population of N individuals, indexed within this chapter by i. At any time t, an individual i is in exactly one of six possible health and vital states:

- $S_i(t)$: susceptible, i. e. can get infected;
- $E_i(t)$: exposed, i. e. infected but not yet *infectious*;
- $I_i(t)$: infectious, i. e. shedding the virus (with or without clinical symptoms) but not hospitalized;
- $H_i(t)$: hospitalized with active disease and also shedding the virus;
- $R_i(t)$: resistant, i. e. immune to the disease through either vaccination or a prior infection (we assume lifelong resistance for now but this can be relaxed to make immunity temporary);
- $D_i(t)$: deceased.

We take $S_i(t)$, $E_i(t)$, $I_i(t)$, $H_i(t)$, $R_i(t)$, and $D_i(t)$ to be Bernoulli random variables that depend on time t and take only the values 0 and 1. That is, $S_i(t) = 1$ when individual i is susceptible at time t, and otherwise $S_i(t) = 0$ (and analogously for the other variables). Because the six SEIHRD states enumerate all health and vital states of individuals in this model, we have

$$S_i(t) + E_i(t) + I_i(t) + H_i(t) + R_i(t) + D_i(t) = 1\,. \tag{6.2}$$

Therefore, there are only five independent states.

In the network epidemiology model, a close contact between individuals i and j establishes a temporary network edge with weight $w_{ji}(t) = 1$ for the duration τ of the contact; outside the contact period, $w_{ji}(t) = 0$. Transmission along the temporary edges from one node to another and transitions between health and vital states within each node are modeled as independent Poisson processes [34, 52, 67, 75]. Each process is characterized by a rate that may vary from node to node and may depend on external variables such as age, sex, and medical risk factors (see Figure 6.2 for a schematic).

We make the following assumptions about the transmission rate and the parameters characterizing transition rates between SEIHRD states, including prior distributions used in the network model for DA:

- *Transmission rate*: During the contact period between an infectious or hospitalized individual ($I_j(t) = 1$ or $H_j(t) = 1$) and a susceptible individual ($S_i(t) = 1$), virus can be transmitted across the edge between nodes j and i. When transmission occurs, the susceptible node i becomes exposed and switches state to $E_i(t) = 1$. During the contact period in which an edge is active ($w_{ji}(t) = 1$), we assume the transmission rate to a susceptible node

with $S_i(t) = 1$ from an infectious node with $I_j(t) = 1$ is $\kappa^I_{ji} = a_{ji}(t)\beta$, and that from a hospitalized node with $H_j(t) = 1$ is $\kappa^H_{ji} = a'_{ji}(t)\beta$. The parameter β is a transmission rate across active edges, which we set to a global constant in the stochastic surrogate-world simulations and learn on a nodal basis in the model used for DA; $a_{ji}(t)$ and $a'_{ji}(t)$ are time-dependent transmission modifiers that can be adjusted to incorporate additional information that may be available, for example, user-supplied information that individual i is using PPE at time t. In our proof-of-concept simulations, we use $a_{ji}(t) = 0.1$ within hospitals and $a_{ji} = 1$ otherwise, to reflect the rarity of SARS-CoV-2 transmission in hospitals in which PPE is worn [87]. (In reality, however, depending on the types of PPE and adherence to hygiene protocols, the degree of transmissibility reduction may vary substantially among hospitals [87].) A typical value for the transmission rate of respiratory viruses is around $\beta = 0.5\ \text{hour}^{-1} = 12\ \text{day}^{-1}$ [89].

Because we model transmission as a Poisson process, the probability that transmission occurs during contact increases with the duration of the contact period τ, e. g., for an infectious node as [73]

$$T_{ji}(\tau) = 1 - e^{-\kappa^I_{ji}\tau}\,.$$

(This holds provided the contact period τ is short relative to the duration of infectiousness, so that the infectiousness status of a node does not change during contact.)

In the model used for DA, we do not assume perfect knowledge of the transmission rate; instead, we learn a partial transmission rate β_i for each node i, and compute transmission rates from node j to node i as the averages $\kappa^I_{ji} = 0.5a_{ji}(t)(\beta_i + \beta_j)$ and $\kappa^H_{ji} = 0.5a'_{ji}(t)(\beta_i + \beta_j)$. We assume independent normal priors for β_i for each node, with a mean of $12\ \text{day}^{-1}$ and a standard deviation of $3\ \text{day}^{-1}$. We truncate these distributions to $[1\ \text{day}^{-1}, 20\ \text{day}^{-1}]$, though in practice these bounds are rarely reached.

- *Latent period*: Exposed nodes with $E_i(t) = 1$ transition to being infectious with $I_i(t) = 1$ at the rate σ_i, which is the inverse of the latent period: the time it takes for an exposed individual to become infectious. For COVID-19, the latent period lies between about 2 days and about 12 days [53, 58, 60]. We take the latent period σ_i^{-1} to be fixed for each node i but heterogeneous across nodes. In the model used for DA, we represent it as $\sigma_i^{-1} = 1\ \text{day} + l_i$, where l_i has a gamma prior distribution with shape parameter $k = 1.35$ and scale parameter $\theta = 2$ day; hence, the minimum latent period is 1 day, and its prior mean value is 3.7 days ($1\ \text{day} + k\theta$).

- *Duration of infectiousness in community*: Infectious nodes with $I_i(t) = 1$ transition to resistant, hospitalized, or deceased at the rate γ_i, which is the

inverse of the duration of infectiousness in the community (i. e., outside hospitals). For COVID-19, the median duration of infectiousness is around 3.5 days [60], but its distribution has a long tail, for example, from individuals with serious or critical disease progression [46]. Like σ_i, we take γ_i to be fixed for each node i but heterogeneous across nodes. In the model used for DA, we model the duration of infectiousness as $\gamma_i^{-1} = 1 \text{ day} + g_i$, where g_i has gamma prior distribution with shape parameter $k = 1.1$ and scale parameter $\theta = 2$ days; hence, the minimum duration of infectiousness is 1 day, and its prior mean value is 3.2 days [46, 60].

- *Duration of hospitalization*: Hospitalized nodes with $H_i(t) = 1$ transition to resistant or deceased at the rate γ_i', which is the inverse of the duration of hospitalization. As before, we take γ_i' to be fixed for each node i but heterogeneous across nodes. In the model used for DA, we model the duration of hospitalization as $\gamma_i'^{-1} = 1 \text{ day} + g_i'$, where g_i' has a gamma prior distribution with shape parameter $k = 1.0$ and scale parameter $\theta = 4$ days; hence, the minimum duration of hospitalization is 1 day, and its prior mean value is 5 days. We assume hospitalized nodes are infectious. (If there is evidence that a hospitalized patient no longer sheds the virus, this can be taken into account by setting the transmission rate modifier $a_{ji}(t)$ from the corresponding node to zero; however, we are not considering this situation in our proof-of-concept.)

- *Hospitalization rate*: We assume a fraction h_i of infectious nodes with $I_i(t) = 1$ requires hospitalization after becoming infectious. More precisely, we assume that infectious nodes transition to becoming hospitalized at the rate $h_i\gamma_i$. This implies that, over a period Δt that is short relative to the duration of infectiousness γ_i^{-1}, the probability of transitioning from being infectious to hospitalized, relative to the total probability of leaving the infectious state, is

$$\frac{1 - e^{-h_i\gamma_i\Delta t}}{1 - e^{-\gamma_i\Delta t}} \approx h_i \quad \text{for} \quad \gamma_i\Delta t \ll 1\,.$$

  We take h_i to be fixed for each node i but heterogeneous across nodes; it generally depends on age and other risk factors [10, 116]. We model the age dependence in the stochastic surrogate-world simulations according to clinical data as described below (Table 6.3), and we assume the same parameters in the model used for DA.

- *Mortality rate*: We assume a fraction d_i of infectious nodes with $I_i(t) = 1$ and a fraction d_i' of hospitalized nodes with $H_i(t) = 1$ die. More precisely, we assume infectious nodes die at the rate $d_i\gamma_i$, and hospitalized nodes die at the rate $d_i'\gamma_i'$. Both d_i and d_i' are fixed for each node but are heterogeneous across nodes, depending on age and other risk factors [10, 116]. Both in

the stochastic surrogate-world simulation and in the model used for DA, we assume the same age-dependent mortality rates (Table 6.3).

- *Resistance*: For now, we assume resistance to be lifelong, so that an individual who becomes resistant remains so indefinitely and does not return to being susceptible. This assumption can be relaxed by allowing transitions back to the susceptible state if resistance is not permanent.

The health and vital states and transition rates define a Markov chain for the individual-level SEIHRD states. The SEIHRD Markov chain on a contact network can be simulated directly with kinetic Monte Carlo methods [41], as in previous studies [34, 35, 62, 89]. We use kinetic Monte Carlo simulations both to benchmark a model for the SEIHRD probabilities and to provide a surrogate for the real world in our proof-of-concept simulations.

6.2.2 *Reduced master equations*

We are principally interested in the individual SEIHRD probabilities, which are the expected values $\langle S_i(t)\rangle$, $\langle E_i(t)\rangle$, etc. associated with the Bernoulli random variables for the states. That is, $\langle S_i(t)\rangle$ is the probability that individual i is susceptible at time t.

These probabilities could be obtained as averages over an ensemble of kinetic Monte Carlo simulations; however, it is more computationally efficient to solve reduced master equations for the probabilities directly. The equations are [75, 96]:

$$\langle \dot{S}_i\rangle = -\left[\zeta_i + k_i^x \langle w_i\rangle P(t)\eta_i\right]\langle S_i\rangle\,, \tag{6.3a}$$

$$\langle \dot{E}_i\rangle = \left[\zeta_i + k_i^x \langle w_i\rangle P(t)\eta_i\right]\langle S_i\rangle - \sigma_i\langle E_i\rangle\,, \tag{6.3b}$$

$$\langle \dot{I}_i\rangle = \sigma_i\langle E_i\rangle - \gamma_i\langle I_i\rangle\,, \tag{6.3c}$$

$$\langle \dot{H}_i\rangle = h_i\gamma_i\langle I_i\rangle - \gamma_i'\langle H_i\rangle\,, \tag{6.3d}$$

$$\langle \dot{R}_i\rangle = (1-h_i-d_i)\gamma_i\langle I_i\rangle + (1-d_i')\gamma_i'\langle H_i\rangle\,, \tag{6.3e}$$

$$\langle \dot{D}_i\rangle = d_i\gamma_i\langle I_i\rangle + d_i'\gamma_i'\langle H_i\rangle\,, \tag{6.3f}$$

where

$$\zeta_i(t) = \frac{\sum_{j=1}^{\tilde{N}} w_{ji}(t)\left(\kappa_{ji}^I\langle S_i(t)I_j(t)\rangle + \kappa_{ji}^H\langle S_i(t)H_j(t)\rangle\right)}{\langle S_i\rangle} \tag{6.3g}$$

is the total infectious pressure on node i from within the network formed by the $\tilde{N}$ users. The infectious pressure represents the possibility of transmission to node i from all network nodes that are at least temporarily connected with node i. Additionally, we have included an exogenous infection rate η_i. This allows for infection from outside the network of $\tilde{N}$ users when the master equation network

represents only a subset of a larger network with N nodes, and so transmission can occur from unaccounted nodes. The exogenous infection rate η_i is scaled by the number of external neighbors k_i^x of node i that are not part of the user network, by the probability $\langle w_i \rangle$ of an edge of node i being active, and by the time-dependent prevalence of infectiousness $P(t)$, estimated from the network of $\tilde{N}$ users as described below in eq. (6.16). The probability of exogenous infection then increases with the prevalence of infectiousness $P(t)$ within the user base, which is taken as a proxy of prevalence outside the user base. In an idealization that may not be achievable in practice, we take the number of external neighbors k_i^x as given from the network structure. In practice, the number of external neighbors can be estimated through use of the same proximity technologies (e. g., Bluetooth) on which exposure notification apps rely, which allow the sensing of other nearby mobile devices, even if they do not participate in the proximity sensing and exposure notification protocol. While this is unlikely to yield perfect knowledge about the number of external neighbors, it may be combined with statistical approximations [99]. The net effect of these assumptions and approximations is that a user surrounded by other users will have no exogenous infection rate, while users with many external neighbors will have a larger exogenous infection rate. We have confirmed in simulations that exact knowledge of the number of neighbors can be replaced by statistical knowledge; for example, replacing the node-dependent k_i^x by the user-network average for all nodes (exterior connectivity in Table 6.1) yields similar results (Figures 6.17 and 6.18).

We use with a `Runge-Kutta-Fehlberg 4(5)` scheme to integrate these ordinary differential equations, with an adaptive timestep of maximum length 3 hours. The weights $w_{ji}(t)$ vary on shorter timescales. This is taken into account in the numerical integration by averaging $w_{ji}(t)$ over the length of a time step, rather than evaluating $w_{ji}(t)$ at discrete intervals.

6.2.3 *Closure of reduced master equations*

The master equations (6.3) for the probabilities are not closed because they depend on the joint probabilities $\langle S_i(t) I_j(t) \rangle$ and $\langle S_i(t) H_j(t) \rangle$ in the infectious pressure (Eq. 6.3g). Various approaches to closing this term have been proposed [52, 75, 96]. Our approach is to estimate it from the ensemble used in the DA cycle, as follows.

The joint-event probability $\langle S_i(t) I_j(t) \rangle$ and the marginal probabilities $\langle S_i(t) \rangle$ and $\langle I_j(t) \rangle$ in the master equations (6.3) are related through the ratio

$$\mathscr{C}\big[S_i(t), I_j(t)\big] = \frac{\langle S_i(t) I_j(t) \rangle}{\langle S_i(t) \rangle \langle I_j(t) \rangle}, \tag{6.4}$$

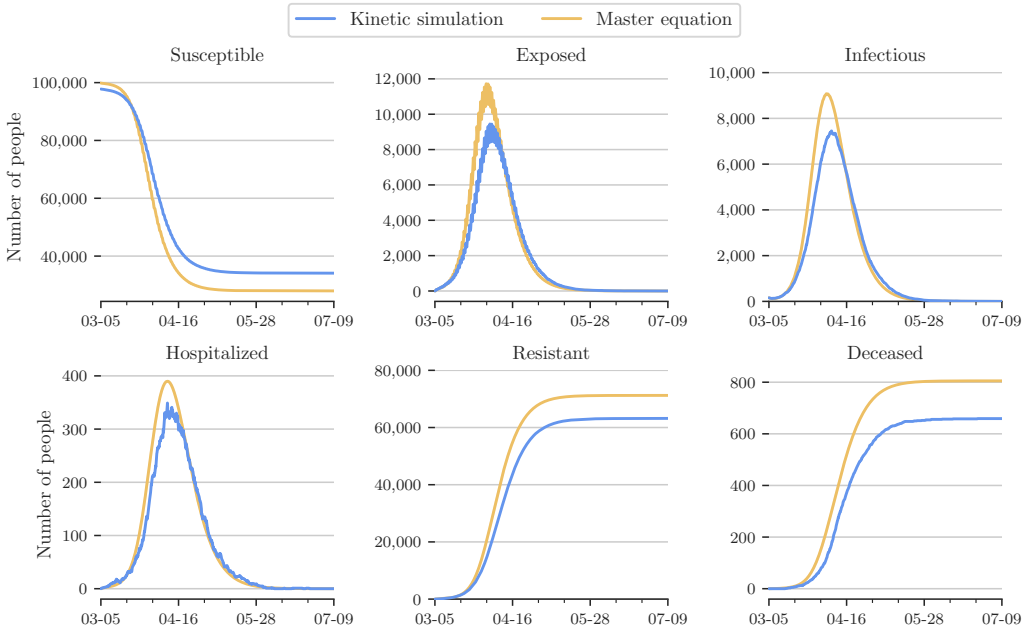


Figure 6.3: Overall epidemic dynamics from SEIHRD model using mean-field approximation.

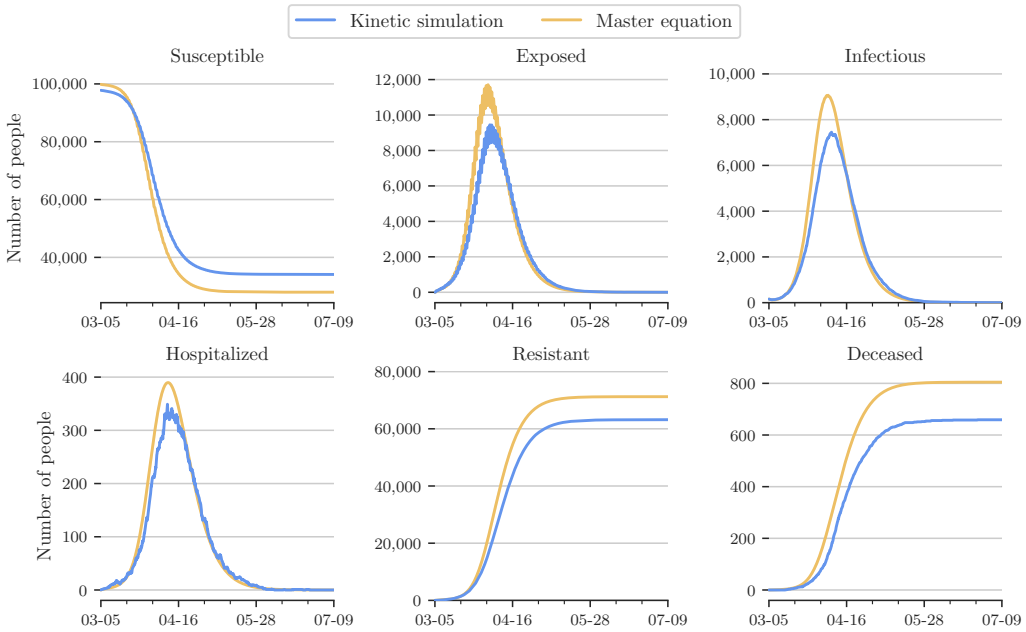


Figure 6.4: Overall epidemic dynamics from SEIHRD model using mean-field approximation with ensemble correction.

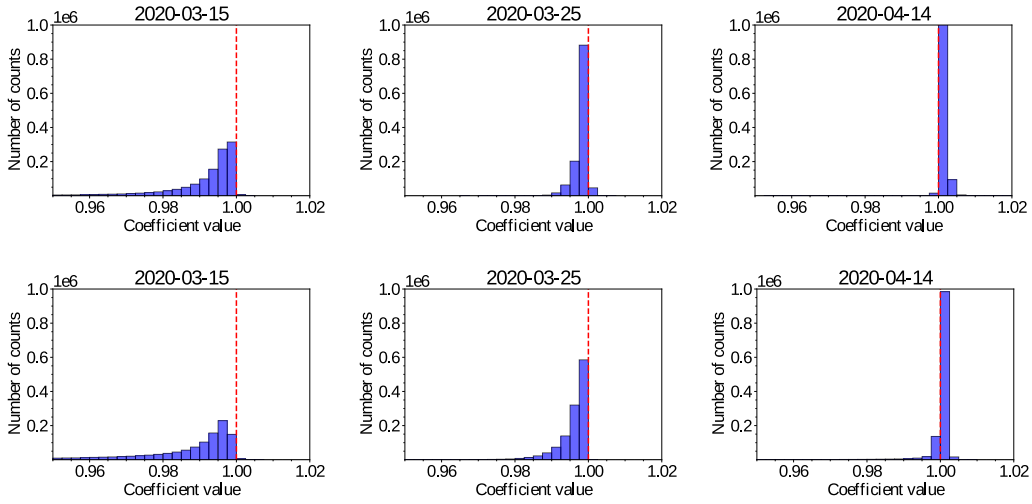


Figure 6.5: Histograms of correction coefficients (top row) $\mathscr{C}_M\big[S_i(t), I_j(t)\big]$ and (bottom row) $\mathscr{C}_M\big[S_i(t), H_j(t)\big]$ at different times during the simulated epidemic.

where $\mathscr{C}\big[S_i(t), I_j(t)\big]$ is the rescaled joint probability of $S_i(t)$ and $I_j(t)$. We estimate it by its ensemble analogue:

$$\mathscr{C}_M\big[S_i(t), I_j(t)\big] = \frac{\overline{\langle S_i(t)\rangle\langle I_j(t)\rangle}}{\overline{\langle S_i(t)\rangle}\;\overline{\langle I_j(t)\rangle}}, \tag{6.5}$$

where $\overline{(\cdot)} = M^{-1}\sum_m(\cdot)$ denotes the mean over the ensemble (with $m = 1, \ldots, M$ labeling ensemble members). Thus, we approximate the joint probability in eq. (6.3g) as follows:

$$\langle S_i(t) I_j(t)\rangle = \langle S_i(t)\rangle\langle I_j(t)\rangle\,\mathscr{C}_M\big[S_i(t), I_j(t)\big]. \tag{6.6}$$

With this empirical approximation, we obtain a closed-form expression for the second moment $\langle S_i^m(t) I_j^m(t)\rangle$ for each ensemble member m, which we use in the reduced master equations. The second moment $\langle S_i^m(t) H_j^m(t)\rangle$ follows analogously. If $\mathscr{C}_M\big[S_i(t), I_j(t)\big] = 1$ and $\mathscr{C}_M\big[S_i(t), H_j(t)\big] = 1$, this reduces to the mean-field approximation that is commonly made in epidemiological models [75, 96] and that often is accurate on real-world networks [42].

We verified this closure against direct kinetic Monte Carlo simulations of the SEIHRD model on the synthetic network described below, in the free-running NYC simulation without lockdown (Fig. 6.10). The closure has similar performance as the mean-field approximation and adequately, albeit not perfectly, captures the stochastic network dynamics (Figures 6.3 and 6.4). The closure correction coefficients (6.5) concentrate close to the value of 1 (Figure 6.5), which explains the similar performance to the mean-field approximation.

6.3 DATA ASSIMILATION

6.3.1 *The algorithm*

For DA, we use a version of the *ensemble adjustment Kalman filter* (EAKF) [5], which has previously been used with epidemiological models [60, 80, 81, 95]. EAKF treats an ensemble of M model parameters and states $\langle S_i^m(t)\rangle$, $\langle E_i^m(t)\rangle$, etc. $(m = 1, \ldots, M)$ from a previous DA cycle as a prior and then linearly updates the ensemble of model parameters and states to obtain an approximate Bayesian posterior given the available data. Unlike other algorithms for computing Bayesian posteriors on networks [4], it makes no assumptions about the network structure, and it scales well to high-dimensional problems [5].

To learn about parameters and the states of nodes on the network, we use a scheme based on iterating forward passes of the master equations over a time window Δ, with EAKF updates between each pass; a similar scheme has been used in epidemiology models before [60, 80, 81, 95]. In this way, we effectively use EAKF as a smoother, harnessing all available data in a DA window $(t_f - \Delta, t_f)$. There are two parts to the DA procedure:

1 UPDATE STAGE. An EAKF update step is performed to assimilate all data available for the window $(t_f - \Delta, t_f)$, using the previous ensemble model run as prior. The mismatch between the simulated ensemble trajectories and the data is used to update the combined ensemble of parameters and states at the initial time $t_f - \Delta$.

2 FORECAST STAGE. The updated ensemble of states $\langle S_i^m(t_f - \Delta)\rangle$, $\langle E_i^m(t_f - \Delta)\rangle$, etc., with the updated model parameters, is integrated forward up to time t_f, to serve as prior for the next update cycle.

EAKF relies on linear updates and assumes Gaussian error statistics. However, the forward equations (6.3) are nonlinear. As a result, the EAKF update does not always conserve total probability, in the sense that SEIHRD probabilities for each node will not always sum to 1. We therefore augment the state with an additional total probability conservation variable, with observation equal to the target probability sum 1. The Gaussian assumption is also at odds with probabilities in $[0, 1]$. We have experimented with approaches of transforming variables to an unbounded space, leading to total probability conservation becoming highly nonlinear. We found it to be more robust to work in the original space where total probability conservation is a linear constraint. This approach does, however, violate Gaussianity assumptions about the ensemble when we reinforce the probability bounds by clipping the states $\langle S_i^m(t_f - \Delta)\rangle$, $\langle E_i^m(t_f - \Delta)\rangle$, etc. to $[0, 1]$.

We assume data errors to be uncorrelated, so that their error covariance matrix is diagonal (see below for how we specify error variances in the proof-of-concept

simulations). Prior information on parameters and states is specified in EAKF through the initial condition of the ensemble. We draw the parameters of the ensemble from the above-specified prior distributions, and we initialize the state by seeding each ensemble member with a fraction of (possibly different) infectious nodes, the rest being susceptible. The initial fraction of infectious nodes is drawn from a beta distribution with shape parameters $\alpha = 0.0016$ and $\beta = 1$ (not to be confused with the transmission rate β). The mean fraction (here, 0.16%) of initially infected nodes agrees with the fraction of initially infected nodes in the stochastic surrogate-world simulations.

To account for the multi-fidelity nature of the assimilated data, we perform EAKF in multiple passes. This allows for better conditioned data covariance matrices and for different hyperparameter choices for the different types of data. We perform the following passes to assimilate data from the lowest to the highest fidelity:

- In a first EAKF pass, we update parameters and states at $t_f - \Delta$ using the poorest fidelity data (e.g., temperature sensor data), followed by a forecast over $(t_f - \Delta, t_f)$;
- In a second EAKF pass, we update parameters and states at $t_f - \Delta$ using moderate-accuracy diagnostic test data, followed by another forecast over $(t_f - \Delta, t_f)$;
- In a final EAKF pass, we update parameters and states at $t_f - \Delta$ using data about hospitalization and death status with small error variances, followed by a final forecast over $(t_f - \Delta, t_f)$.

There are three well-established challenges that ensemble-based filters must tackle when assimilating a number of parameters/states that is large relative to the ensemble size [49]: overestimation of long-range covariances, underestimation of variances, and ensemble collapse.

1. To prevent spurious long-range covariances, we localize the effect of observations on states within a single node [6, 49]. That is, direct updates of a nodal state are only due to observations at that node during the DA window. This also provides large computational savings because EAKF updates may be performed sequentially node-by-node, in any order.

2. To prevent underestimation of variances by the finite-size ensemble, which can lead to discounting of data points [5], and to ensure well-posedness of the matrix inversions, we use regularization of the ensemble covariance matrix Σ. If $\Lambda_{\min}$ and $\Lambda_{\max}$ denote the minimum and maximum eigenvalues of Σ, we replace Σ in the EAKF algorithm with with $\Sigma + \max(\delta(\Lambda_{\max} - \Lambda_{\min}), \delta_{\min})I$. We choose $\delta = 5/M$ to assimilate diagnostic test data, and $\delta = 1/M$ to assimilate hospitalization/death status; $\delta_{\min}$ is taken to be the mean observational noise standard deviation of the update.

3. To prevent ensemble collapse, we add a hybrid inflation to an assimilated state with a map $x \mapsto a(x-\bar{x})+\bar{x}+N(0, b\bar{x})$, where $\bar{x}$ is the ensemble mean state and $N(0, b\bar{x})$ is Gaussian noise with mean zero and standard deviation $b\bar{x}$ [49]. We take $a = 3.0$ and $b = 0.1$.

Because of the binary nature of the hospitalization and death data, we do not update these states directly; doing so can lead to shocks in the system dynamics. We only update the SEIR states $\langle S_i(t_f-\Delta)\rangle$, $\langle E_i(t_f-\Delta)\rangle$, $\langle I_i(t_f-\Delta)\rangle$, $\langle R_i(t_f-\Delta)\rangle$ at the beginning of a DA window $t_f - \Delta \le t \le t_f$ when assimilating hospitalization and death data that fall within the DA window.

6.3.2 *Testing strategy*

We use a simple testing strategy that randomly tests a given budget of nodes once per day. Our framework provides a testbed for different strategies. We found random testing consistently outperformed three other simple strategies: (i) concentrating the test budget on near-neighbors of positively tested nodes; (ii) continuous testing of a fixed subset of the population; and (iii) testing nodes with high predicted infectiousness values. We attribute this to the low prevalence of disease. However, it is possible that more effective testing strategies can be discovered that exploit estimated nodal states, the network structure, and the intervention strategy. In a real-world scenario, systematic biases in testing (e. g., testing biased toward certain workplaces or educational institutions) may also affect quantitative details of our results.

6.3.3 *Parameter Learning*

In addition to assimilating probabilities of SEIHRD states, we can in principle learn about parameters in the reduced master equation model (6.3), for example:

- individual partial and time-dependent transmission rates β_i;
- individual inverse latent periods σ_i;
- individual inverse durations of infectiousness γ_i and hospitalization γ_i';
- individual hospitalization rates h_i and mortality rates d_i and d_i';
- exogenous infection rates η_i.

We have not fully explored the efficacy of learning about the different parameters from data. For now, we include only the partial transmission rates β_i, the inverse latent periods σ_i, and the durations of infectiousness γ_i and hospitalization γ_i' in the DA, all with the priors stated above. Figure 6.6 shows the prior distributions of the parameters at the beginning of the epidemic, as well as the posterior

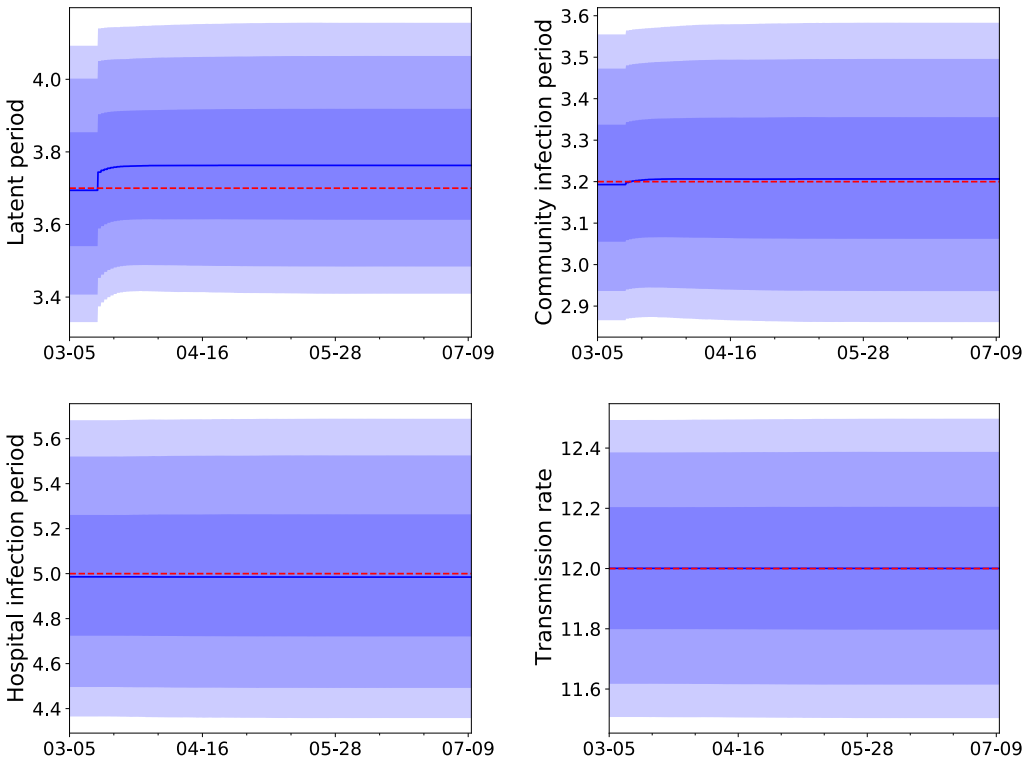


Figure 6.6: Distribution of ensemble averaged model parameters across nodes as a function of time during the epidemic.

The shaded regions contain 50%, 80% and 90% of the distribution. The dashed line represents the true parameters in the stochastic simulation. During the first 8 days, no DA is performed, and the parameter distributions are the prior distributions.

distributions as the epidemic evolves and the network model learns about the parameters. The results show that the DA does not refine the prior estimates of the parameters. When priors were not initially centered on the true values, they remained biased during the simulation. Further investigations focusing, for example, on learning statistical averages of parameters rather than individual node-per-node parameters would be beneficial.

The hospitalization rates h_i and mortality rates d_i and d_i' are fixed at the same values as in the stochastic surrogate-world simulation (Table 6.3). We assume the exogenous infection rates η_i to be equal to the partial transmission rates β_i, and we estimate the probability of an edge of node i being active as

$$\langle w_i \rangle = \frac{\bar{A}_i}{\mu + \bar{A}_i}, \tag{6.7}$$

where $\bar{A}_i$ is the diurnally averaged edge activation rate,

$$\bar{A}_i = \frac{1}{1\text{ day}} \int_0^{1\text{ day}} A_i(t)\ dt, \tag{6.8}$$

with

$$A_i(t) = \frac{1}{\hat{k}} \max \left\{ \lambda_{i,\min},\ \lambda_{i,\max} \left[1 - \cos^4 \left(\frac{\pi t}{1\text{ day}} \right) \right]^4 \right\}. \tag{6.9}$$

With our parameters, this is $\bar{A}_i = \lambda_{i,\min}/\hat{k} = 0.4\text{ day}^{-1}$ for isolated nodes and $\bar{A}_i = 3.77\text{ day}^{-1}$ otherwise. For a stationary birth-death process, this estimate for $\langle w_i \rangle$ is the stationary probability of an edge being active; it approximates the diurnally averaged probability in the case of the birth-death process with diurnally varying edge activation rates. Through this probability $\langle w_i \rangle$, the exogenous infectious pressure depends on the isolation status of a node.

6.3.4 *Classification in Risk Network*

Nodes i in the community group (c) are classified as possibly infectious ($\mathcal{I}_i = 1$) or not ($\mathcal{I}_i = 0$) according to

$$\mathcal{I}_i = \begin{cases} 1 & \text{if } \overline{\langle I_i^m \rangle} > c_I, \\ 0 & \text{otherwise.} \end{cases} \tag{6.10}$$

Here, c_I is a classification threshold, which can be determined from *receiver operating characteristic* (ROC) curves as some optimum tradeoff between wanting to achieve high true positive rates while keeping false positive rates modest. The ROC curves we use are adapted to the setting in which we are primarily

interested in the fraction of users that is classified as possibly infectious (and thus may be asked to self-isolate). We plot the *true positive rate* (TPR), i. e. fraction of the nodes with $\mathcal{I}_i = 1$ for which $I_i = 1$ in the stochastic simulation, against the *true positive rate* (TPR), fraction of the nodes with $\mathcal{I}_i = 1$ in the user base of size $\tilde{N}$. ROC curves are traced out by lowering the classification threshold c_I, thereby increasing both TPR and predicted positive fraction (PPF).

6.3.5 *Classification in TTI*

For the TTI scenarios, we assume the dynamic contact network among users is known, as in the network DA scenarios, and we assume instantaneous tracing. When a node i in the community group (c) is tested positive, it is classified as infectious; all nodes that have had at least one 15-minute contact with node i within the preceding 10 days are classified as exposed. All infectious and exposed community nodes are immediately isolated. This TTI scheme mimics the methods of typical exposure notification apps; although it is idealized and overestimates TTI performance achievable in practical settings [28, 36], it provides a fair baseline for comparison with the risk network model.

6.3.6 *Contact Interventions*

We implement two types of intervention scenarios in our test cases. In the first, a lockdown scenario (Figure 6.10), we set $\lambda_{i,\max}$ for all nodes in the community group (c) to 33 day^{-1}. This amounts to a reduction of the mean contact rate (6.13) in group (c) by 58%. In the second, a time-limited isolation intervention, we reduce the contact rates of targeted high-risk nodes by setting $\lambda_{i,\max} = \lambda_{i,\min} =$ 4 day^{-1}; thus, these high-risk nodes are assumed to self-isolate, with only 4 contacts per day on average, corresponding to a reduction of their average contact rate by 91%.

The duration of contact reduction takes three possible values. In the lockdown scenario (Figure 6.10), all nodes have contact reduction from the inception of the lockdown until the end of the simulation. In the TTI scenario, self-isolating nodes have contact reduction for 14 days, in accordance with current CDC guidelines [24], after which contact rates are reset to the original values. For the risk network scenario, self-isolating nodes have contact reduction until they are classified as negative ($\overline{\langle I_i^m \rangle} \le c_I$) for a period of 5 consecutive days, after which contact rates are reset to their original values. This corresponds to reinstatement of original contact rates for 50%, 90% and >99% of isolated nodes within 7, 14 and 21 days, respectively.

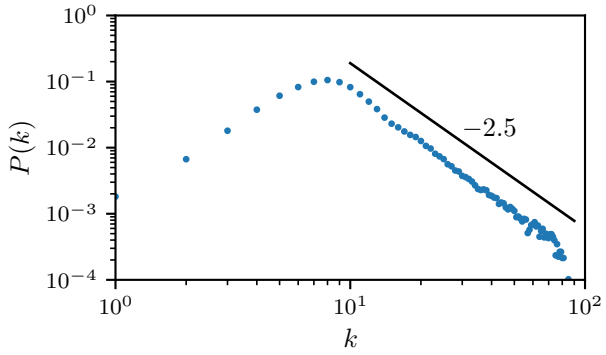


Figure 6.7: Distribution of degrees k in synthetic contact network with 97,942 nodes.

6.4 SYNTHETIC NETWORK FOR PROOF-OF-CONCEPT

To illustrate the methods with simulated data in a computational proof-of-concept, we construct a large synthetic contact network with $N = 97,942$ nodes and about 1 million connections among them. The network has typical characteristics of a human contact network. It has a time-dependent contact rate minimum at night and a maximum midday, and it has a connectivity (degree) distribution similar to human networks: there are many individuals with few connections and a few highly connected individuals who are more likely to become superspreaders [31] (Figs. 6.7, 6.8).

The network also contains a block representing hospitals, where hospitalized patients are connected to healthcare workers, who in turn are connected to the community in the rest of the network. Transmission rates in hospitals are reduced by a factor of 10 to reflect the use of PPE, which has proven effective in making SARS-CoV-2 transmission rates in hospitals rare. The purpose of explicitly including hospitals in the network architecture is twofold: first, to illustrate how reliable data (such as hospital admittance records) can be incorporated in the risk network model; second, to enable comparison of hospitalization rates in the simulated and real epidemic while mimicking the reduced transmission rate in hospitals. Realistic human contact networks contain other structures, such as households, workplaces, and schools [1]. Such features are not explicitly taken into account in our synthetic network architecture; rather, contact clusters arise randomly in the synthetic network. In the real world, such clusters would arise naturally in the contact network assembled from proximity data, without the need to account for them explicitly.

As surrogates for real-world health data, we use stochastic simulations of the epidemiology model for the state variables $S_i(t)$, $E_i(t)$, etc. on the network. We reproduce various scenarios of the early COVID-19 epidemic in New York City (NYC), beginning during March 2020. Because age is an important risk factor

for COVID-19 severity, we assign ages to nodes based on the age distribution of NYC and use them to model age-dependent disease progression according to current knowledge about COVID-19 (Tables 6.2 and 6.3). While we assign ages to nodes randomly, the realism of the model could be improved with age-stratified contact patterns [68, 69]. With the resulting surrogate worlds of contact patterns and disease progression, we explore how individual risk assessment and epidemic management and control can be achieved in what-if scenarios.

6.4.1 *Synthetic network for proof-of-concept*

We generate a synthetic time-dependent contact network in two steps:

1. We generate a static degree-corrected *stochastic block model* (SBM) [51, 82], consisting of N nodes in three groups. The three groups represent (a) hospitalized patients, (b) healthcare workers with contacts both within hospitals and in the community, and (c) the community of all remaining individuals (e. g., people in an urban environment). Hospital beds in group (a) are filled when infected nodes become hospitalized; we assume an infinite supply of hospital beds. Healthcare workers in group (b) make up 5% of all nodes, and the remaining 95% of nodes constitute group (c).

   We describe connections within groups (a) and (b) with an Erdős–Rényi model and use mean degrees of 5 in group (a) and 10 in group (b), based on a social-contact analyses in a hospital setting [30]. Hospitalized patients in group (a) can interact only with each other and with the healthcare workers in group (b). To model the interactions between groups (a) and (b), we set the corresponding mean degrees per node to 5 for edges connecting the groups. We parameterize the contacts among nodes in the community group (c) with a power-law degree correction. As pointed out in [29], when groups are ignored, degree distributions associated with social interactions are well-described by a negative binomial distribution, which, for example, has also been used to describe degree distributions associated with sexual-contact networks [44]. In the presence of groups, however, degree distributions of social-interaction networks have been found to exhibit a power-law tail with an exponent of about 2.5 [29]. In accordance with the results presented in [18, 29], we therefore describe parts of the synthetic contact network by a stochastic block model with power-law degree correction with exponent 2.5, mean degree $\hat{k} = 10$, and maximum degree 100; Figure 6.7 shows the degree distribution. The community (c) as a group only interacts with healthcare workers (b), and we set the corresponding mean degree to 5.

2. To model time-dependence of the network, we make the edges of the static SBM network created in the first step time-dependent by switching

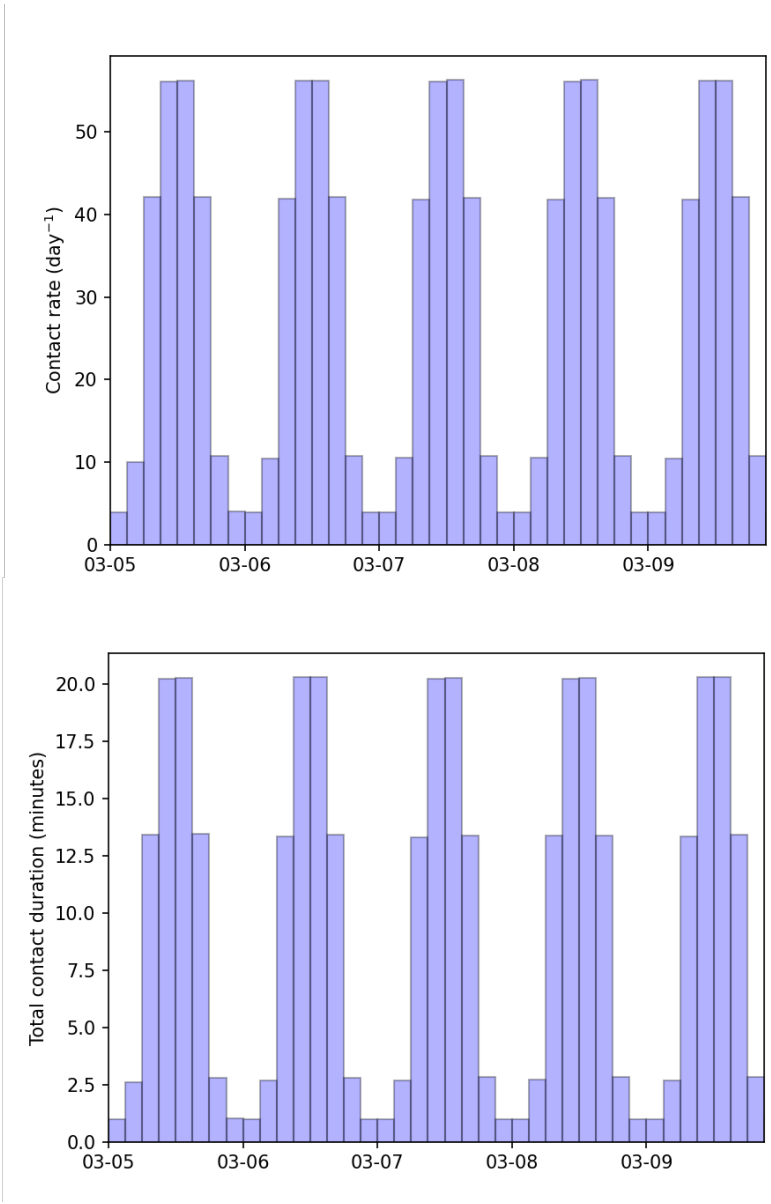


Figure 6.8: Dynamic contact network behavior in the first five simulated days, batched into 3-hour windows (starting at midnight).

Displayed are the ensemble-averaged and node-averaged contact rate and total contact duration.

them on and off. That is, neighbors of all nodes remain fixed in time, but their connections are activated or deactivated with time. We account for day/night cycles in the edge weights $w_{ji}(t)$, but we ignore, e. g., weekly cycles. We generate a diurnal cycle that replicates some properties of observed time-dependent human contact networks [37]: The fraction of active edges is small at night and in the early morning hours, reaches a maximum around noon, and approaches small values again in the evening.

To model the time-dependence of $w_{ji}(t)$, we use a birth-death process commonly used in queuing theory. The birth-death process is a Markov chain in which arrivals (edge activations) are inhomogeneous Poisson processes with a diurnally varying mean rate $A_{ji}(t)$; contact durations are exponentially distributed with a mean contact duration τ (i. e., a mean rate parameter $\mu = \tau^{-1}$). We choose a mean duration of $\tau = 2$ min (hence $\mu = 720\ \text{day}^{-1}$), based on high-resolution human contact data [89]. We model the mean edge activation rate as

$$A_{ji}(t) = \frac{1}{\hat{k}} \max\Big\{ \min(\lambda_{j,\min}, \lambda_{i,\min}), \min(\lambda_{j,\max}, \lambda_{i,\max}) \left[1 - \cos^4\left(\frac{\pi t}{1\ \text{day}}\right)\right]^4 \Big\}. \tag{6.11}$$

Here, $t = 0$ starts at midnight, and $\hat{k}$ is the mean degree of the network in the community group (c), so that $\hat{k}A_{ji}$, when averaged over edges, is an average contact rate per node. The diurnally averaged edge activation rate then is

$$\bar{A}_{ji} = \frac{1}{1\ \text{day}} \int_0^{1\ \text{day}} A_{ji}(t)\ dt. \tag{6.12}$$

For the minimum and maximum contact rates per node, $\lambda_{i,\min}$ and $\lambda_{i,\max}$, we choose the default values $\lambda_{\min} = 4\ \text{day}^{-1}$ and $\lambda_{\max} = 84\ \text{day}^{-1}$. If the default contact rates apply for all nodes, this gives for the community group (c) a mean contact rate per node of

$$\hat{k}\bar{A}_{ji} \approx 37.7\ \text{day}^{-1}; \tag{6.13}$$

this is about a factor 3–4 larger than typical human contact rates as assessed by self-reports [70], consistent with the fact that we also take fleeting contacts into account that would likely not be self-reported. The minimum and maximum contact rates $\lambda_{i,\min}$ and $\lambda_{i,\max}$ for a node i are the principal

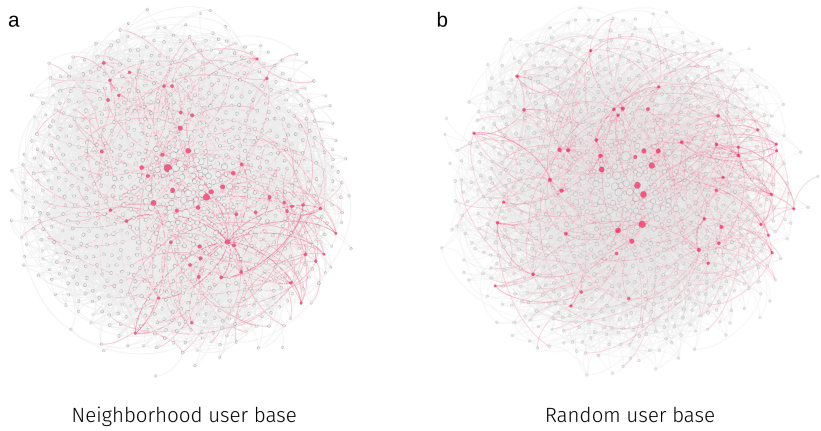


Figure 6.9: Illustration of different user base topologies.
(a) Neighbor-based user base, constructed by iteratively adding neighborhoods. (b) Random subnetwork of users. Red nodes and edges are part of a user base, grey nodes and edges of the overall population. The shown networks have 982 nodes and 5,916 edges. Both user bases contain 5% of all nodes.

parameters we vary to explore the effect of contact interventions. Reducing $\lambda_{i,\min}$ and $\lambda_{i,\max}$ for a node reduces the fraction of time edges connecting to node i are active. The contact rate and total contact duration over the network for five simulated days are displayed in Figure 6.8.

The time dependencies of all edges $w_{ji}(t)$ are treated as independent. We update the time-dependence of each edge at midnight every simulated day, running independent birth-death processes with parameters $A_{ji}(t)$ and μ for the next day.

If a node becomes hospitalized, it is deactivated at its previous location in the network and transferred to the hospital group (a). Hospitalized nodes are assumed to be infectious. (This assumption may later be relaxed to model the situation that a patient is no longer infectious but may still be hospitalized with ongoing disease.)

Different choices of network architecture are, of course, possible and justifiable. The network merely serves to generate simulated data for our proof-of-concept, and the algorithms we demonstrate adapt to any network architecture, which in practice would be provided by proximity data. We do not expect our results to depend sensitively on our choice of network architecture.

6.4.2 *Selection of subnetwork for user base*

To select a subnetwork for a user base with $\tilde{N} < N$ users, we construct subgraphs with two different topologies. First, a neighbor-based subgraph is constructed

Type	Population	Interior	Exterior connectivity
75% neighbor	73,456	21,499	1.9
50% neighbor	48,971	6,301	5.2
25% neighbor	22,381	2,107	10.0
75% random	73,353	7,061	3.1
50% random	48,371	550	6.3
25% random	24,482	33	9.3

Table 6.1: Details of the different user bases. The percentage represents approximately $\tilde{N}/N$, for the user population $\tilde{N}$. The interior defines how many users are completely surrounded by other users. The exterior connectivity gives the average number of exterior nodes connected to a node inside the user base.

from a randomly selected seed user, by adding all neighbors of this user to the subgraph, then in a greedy fashion adding all neighbors of each member of this new subgraph, and so on, until a desired user population is reached. Second, a random subgraph is constructed by randomly choosing nodes from the full network. Figure 6.9 illustrates the different user base topologies, and Table 6.1 summarizes their characteristics.

From Table 6.1, we see the topology of the user base affects the average number of external neighbors. To mitigate this effect, we take into account that users can be infected by neighbors external to the user base, through the additional infectious pressure terms in the equations for $\langle \dot{S}_i \rangle$ and $\langle \dot{E}_i \rangle$ in Eq. (6.3). Such neighbors are still detectable by proximity technologies (e. g., Bluetooth), but because they are not users of the network DA protocol, we do not know their current state.

6.4.3 *Surrogate world simulation*

To generate surrogate worlds with which to test the DA algorithm and interventions, we simulate epidemics on the synthetic network stochastically with kinetic Monte Carlo methods [41]. For these stochastic simulations (but not for the model used for DA), we choose mean transmission and transition rates between SEIHRD states that are homogeneous across nodes, except for hospitalization and mortality rates that depend on age. The mean rates we use are based on current knowledge about COVID-19 (Table 6.2). We simulate 20 epidemics that only differ in their random seed. They are initialized on March 5, 2020, with 0.16% of nodes randomly assigned to be infectious.

The dependencies of the hospitalization rate $h_i = h(a)$ and mortality rates $d_i = d(a)$ and $d'_i = d'(a)$ on age a are estimates based on recent data (Table 6.3).

Parameter	Value
β, κ^I	$12\ \text{day}^{-1}$
κ^H	0.1β
σ	$(3.7\ \text{day})^{-1}$
γ	$(3.2\ \text{day})^{-1}$
γ'	$(5\ \text{day})^{-1}$
λ_{max}	$84\ \text{day}^{-1}$
λ_{min}	$4\ \text{day}^{-1}$

Table 6.2: Mean transmission and transition rates and maximum/minimum contact rates for the surrogate-world simulations with the stochastic SEIHRD model (Fig. 6.2). The mean rates are taken to be the same for all nodes; hence, the nodal indices are suppressed. The latent period σ^{-1} and duration of infectiousness in the community γ^{-1} are approximated from those in refs. [60] and [46]; the duration of hospitalization γ'^{-1} is from ref. [86], and the transmission rate β is fit to be consistent with data for respiratory viruses [89] and to roughly reproduce NYC data.

To model age-dependent disease progression, we randomly assign ages to network nodes in the community group (c) according to the age distribution for NYC, as given by U.S. Census data [107]. Additional factors we neglect in our synthetic examples, such as age-dependent contact patterns, are likely small perturbing factors for the risk assessment results we show. We assign ages to nodes in the healthcare worker group (b) according to the age distribution among working-age adults (21–65 years old). Initially, there are no hospitalized nodes (i.e., group (a) is empty).

The network has 97,942 nodes (with the difference to 100,000 arising from stochastic effects in the generative algorithm). We choose the global mean transmission rate β so that our simulations are qualitatively aligned with the evolution of the COVID-19 epidemic in NYC [74]. We find that a global value of $\beta = 12\ \text{day}^{-1}$ can qualitatively reproduce the observed rate of infections and can quantitatively reproduce the rate of hospitalizations and deaths during the initial phases of the epidemic in NYC.

6.4.4 *Synthetic data*

We sample synthetic data from the stochastic surrogate-world simulation on the network with N nodes and assimilate data for a possible subset of $\tilde{N} \leq N$ users in the reduced master equation model. To model data errors, we randomly corrupt the synthetic data sampled from the surrogate world network with the

Age group a (yrs)	$f(a)$	$h(a)$	$d(a)$	$d'(a)$
0–17	20.7%	0.2%	0.0001%	1.9%
18–44	40.0%	1.0%	0.001%	7.3%
45–64	24.5%	4.0%	0.1%	19.3%
65–74	8.3%	7.6%	0.7%	32.7%
≥ 75	6.5%	16.0%	1.5%	51.2%

Table 6.3: Age-dependent mean hospitalization and mortality rates in the surrogate-world simulation. The share $f(a)$ of the population in each age group a is taken from U.S. Census data [107]. The age-dependent death rate in hospitals d' is obtained from cumulative hospitalization and death rates in NYC by June 1, 2020 [74], under the assumption that 90% of deaths occurred in hospitals. Age-dependent hospitalization rates $h(a)$ and mortality rates $d(a)$ in the community (outside hospitals) are difficult to obtain directly from NYC data because of an age-dependent undercount of infections [118]. We choose hospitalization rates $h(a)$ that approximate data from France [90], adjusting the rates so that the overall hospitalization rate is $\sum_a f(a)h(a) \approx 3.1\%$, which is NYC's overall hospitalization rate if one assumes a cumulative COVID-19 incidence rate of 23% [88], together with NYC's actual hospitalization count (52,333 on June 1, 2020) and population (8.34 million) [74]. Similarly, the mortality rate in the community $d(a)$ is chosen such that the overall infection fatality rate is $\sum_a f(a)\big[d(a) + h(a)d'(a)\big] \approx 1.1\%$, which is NYC's overall infection fatality rate if one considers the same cumulative incidence of 23% and the confirmed and probable cumulative death count from COVID-19 (21,607 by June 1, 2020).

false positive and false negative rates implied by the sensitivity (false negative rate = 1 – sensitivity) and specificity (false positive rate = 1 – specificity). The types of data and their error rates are outlined below.

6.4.4.1 *Testing positive on high-fidelity tests*

A positive virus test for node i is taken to imply

$$\langle I_i(t) \rangle = \text{PPV} \tag{6.14}$$

at the time the test sample is taken. The *positive predictive value* (PPV) is calculated as

$$\text{PPV} = \frac{\text{sensitivity} \times P(t)}{\text{sensitivity} \times P(t) + (1 - \text{specificity}) \times (1 - P(t))}, \tag{6.15}$$

where we take the sensitivity of the test to be 80% and the specificity to be 99%, which we use as an approximation of the currently imprecisely known actual sensitivities and specificities [94, 112]. As an estimate of the infectiousness prevalence $P(t)$ in the population, we use the average of the infectiousness probabilities both over the network of size $\tilde{N}$ and over the ensemble of size M,

$$P(t) = \max\left(\frac{1}{\tilde{N}M} \sum_{m=1}^{M} \sum_{i=1}^{\tilde{N}} \langle I_i^m(t) \rangle, \frac{1}{\tilde{N}} \right). \tag{6.16}$$

The cutoff of $\tilde{N}^{-1}$ is included to guard against erroneously assuming prevalence to be zero because of subsampling on the reduced network. For the DA, we assume an error rate of 1 – PPV for a positive test result.

6.4.4.2 *Testing negative on high-fidelity tests*

A negative virus test for node i is similarly taken to imply

$$\langle I_i(t) \rangle = \text{FOR} \tag{6.17}$$

at the time the test sample is taken. The false omission rate (FOR) is calculated as

$$\text{FOR} = \frac{(1 - \text{sensitivity}) \times P(t)}{(1 - \text{sensitivity}) \times P(t) + \text{specificity} \times (1 - P(t))},$$

with the same sensitivity, specificity, and prevalence as for a positive virus test. For the DA, we assume an error rate equal to FOR for a negative test result.

6.4.4.3 *Low-fidelity tests*

To assimilate low-fidelity data such as those from temperature sensors, we assume $\langle I_i(t) \rangle$ = PPV as for a positive virus test when they indicate infectiousness (e. g., when a temperature reading is elevated). However, we use a sensitivity of 20% and specificity of 98% to reflect the lack of sensitivity of temperature sensors in detecting COVID-19 infection [16]. For the DA, we assume an error rate equal to 1 – PPV, analogous to a positive virus test.

6.4.4.4 *Low-fidelity tests*

Data about hospitalization with COVID-19 imply that $H_i(t) = 1$ for the duration of hospitalization. We assume the hospitalization status of all users to be known with certainty, that is, we assimilate the hospitalization status $H_i(t) = 0$ or $H_i(t) = 1$ for all users; however, we only update the SEIR probabilities at the beginning of a DA window $t_f - \Delta \leq t \leq t_f$ with hospitalization data.

6.4.4.5 *Deaths*

Death implies $D_i(t) = 1$. We assume the vital status of all users to be known with certainty, that is, we assimilate $D_i(t) = 0$ or $D_i(t) = 1$ for all users; as for hospitalization, however, we only update the SEIR probabilities at the beginning of a DA window $t_f - \Delta \leq t \leq t_f$ with death data.

6.4.4.6 *Anti-body tests*

For completeness, we state that a positive serological test for SARS-CoV-2 for node i would be taken to imply

$$\langle R_i(t) \rangle = \text{PPV},$$

with the positive predictive value calculated from (6.15). Typical values for sensitivity would be 90% and for specificity 95% [108], and the prevalence of resistance can be estimated from the resistance probabilities on the reduced master equation network. We would again assume an error rate equal to 1 – PPV. However, we did not assimilate simulated serological tests in our proof-of-concept because achievable serological test rates were low at the time of the study.

6.5 LARGE-SIZED NETWORK NUMERICAL RESULTS

6.5.1 *Lockdown and world avoided*

As an illustrative example, we simulate the evolution of an epidemic that, when scaled up from our network size to the NYC population of 8.3 million, resembles

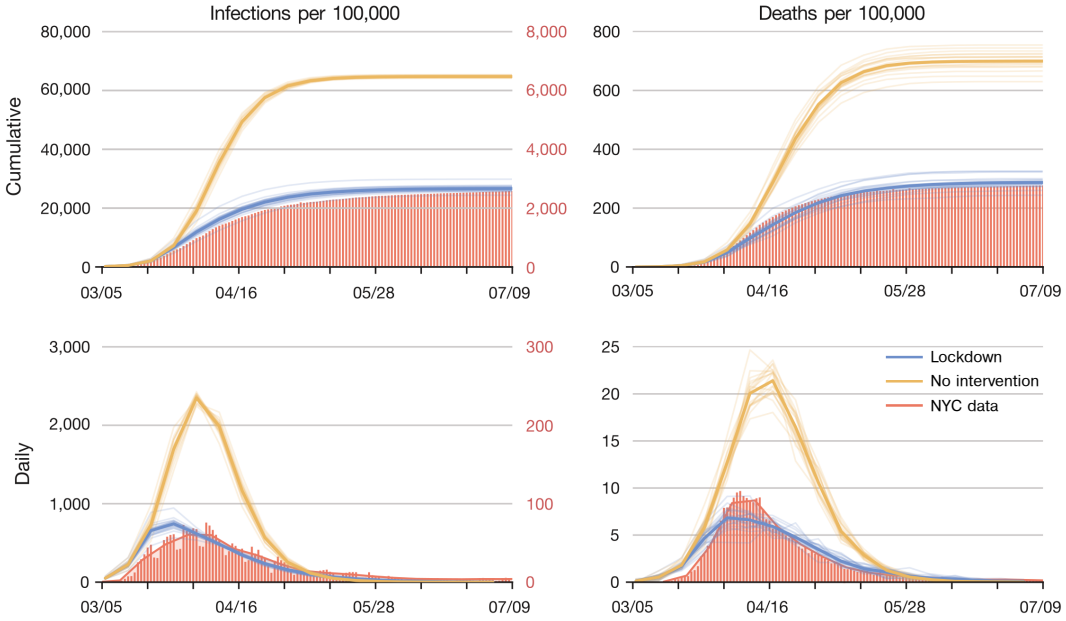


Figure 6.10: Evolution of an outbreak in surrogate-world simulations with a lockdown (blue) and without (orange).

The left column shows infection rates and the right column death rates. Upper row for cumulative counts and lower row for daily counts, smoothed with a 7-day moving average filter. Red bars represent confirmed and probable COVID-19 deaths and confirmed infection rates for New York City [74], with the red axis labels on the right for confirmed infection counts. Solid lines indicate the corresponding counts in the simulations, with the black axis labels on the left for infections on the network. (The axes for infections in the simulations are stretched by a factor 10 relative to the axes for confirmed NYC infections, reflecting the undercount of infections by confirmed cases [118].) The light lines show 20 simulations that only differ by random seeds, with the thicker lines indicating the ensemble mean; thus, they give an indication of sampling variability. The average contact rate for all nodes is reduced by 58% from March 25, 2020 onward to mimic the lockdown effect (blue solid line).

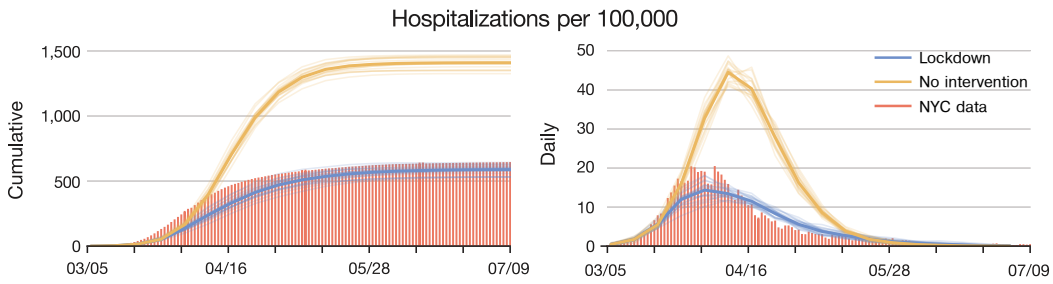


Figure 6.11: Hospitalization rates in surrogate-world simulation with a lockdown (blue) and without (orange).

The left panel shows cumulative hospitalizations and the right panel daily hospitalizations per 100,000 population for the same simulations as those in Figure 6.10. Red bars represent COVID-19-related hospitalization rates for New York City [74]. As in Figure 6.10, the simulation data are smoothed with a 7-day moving average filter.

the early evolution of the COVID-19 epidemic in NYC in 2020 (Fig. 6.10). If the contact rate on the network remains unchanged in the simulations, the number of infections and deaths rises from early March into April, with daily deaths reaching a peak of around 21 per 100,000 population in the second half of April.

However, this world was avoided by a lockdown, which became mandatory in NYC from March 22 onward. In its wake, the number of daily new cases and deaths began to decline from mid-April onward (Figure 6.10). We can reproduce similar behavior in the stochastic simulations by reducing the average contact rate of all nodes by 58% starting March 25 (Figure 6.10). The infection rates in the stochastic simulations exceed the number of confirmed cases in NYC, presumably because the latter undercount actual infections [118]. However, the death rates in the stochastic simulations are close to the NYC death rate (Figure 6.10). The hospitalization rates in the simulation also track the actual hospitalization rates closely (Fig. 6.11).

Thus, the simulated epidemics on the synthetic network reproduce statistics similar to the actual early epidemic in NYC, with realistic parameter choices for transmission rates and disease progression. Notwithstanding the simplifications of the network structure, this points to the qualitative adequacy of the synthetic network epidemiology model as a testbed for network DA, which makes no a priori assumptions about the structure of the network.

6.5.2 *Accuracy of individual risk assessment*

To demonstrate the accuracy of individual risk assessments, we assume the network DA platform has $\tilde{N} \leq N$ users who exchange proximity data with each other, with 25% to 100% of the population in the user base (i.e., $0.25 \leq \tilde{N}/N \leq 1$).

In an idealization, the contact patterns of individuals within the user base are assumed to be known completely; the contact patterns of individuals outside the user base are assumed unknown. We also assume the number of external contacts of individuals in the user base to be known, for example, from proximity-sensing devices that can also detect devices of non-users. (However, we have verified that this assumption can be replaced by only assuming knowledge of the average number of external contacts, without material effects on the results; see Figures 6.17 and 6.18.) For subsets of the $\tilde{N}$ users, we assimilate results of simulated rapid diagnostic tests from the corresponding nodes in the surrogate-world simulation. A fraction f = 1%, 5%, or 25% of the user base is assumed to be tested daily, with results available on the day of test administration; that is, every user is tested on average every 100, 20, or 4 days. Testing the population of a major metropolitan center such as NYC every 100 or 20 days is achievable in principle, as was the case with the testing capacity during the height of the COVID pandemic. For more limited user bases ($\tilde{N}/N \leq 1$), test rates of 25% per day within the user base are locally achievable and in fact are routine, for example, on some college campuses.

We first illustrate the risk network model in the worst-case scenario of the free-running synthetic epidemic, in which contact patterns do not change. DA begins on March 5. We show results for April 9, near the epidemic peak, when about 7% of the population are infectious, and for April 30, when new infections are waning (Fig. 6.10). (In this free-running epidemic, the maximum prevalence of infectiousness is considerably higher than in the lockdown simulations, in which prevalence peaks at 1.5%–2% — more in line with what actually occurred during the lockdown in NYC.)

We classify users i as possibly infectious ("positive") when the estimated probability of infectiousness $\langle I_i(t) \rangle$ exceeds a threshold c_I. The TPR — the rate at which users who are infectious in the stochastic surrogate-world simulation are classified as positive — naturally increases as the classification threshold c_I decreases; at the same time, the PPF — the rate at which users overall are classified as positive, whether correctly or incorrectly — also increases because the *false positive rate* (FPR) increases. The ROC curves trace out these competing changes in TPR and PPF (or FPR) as the classification threshold c_I is varied (Figure 6.12). Choosing a classification threshold c_I means finding a trade-off between wanting a high TPR while keeping FPR and hence PPF low.

In the ideal albeit unrealistic scenario when the user base encompasses the whole population ($\tilde{N}/N$ = 100%), TPRs for April 9 are 12%, 19%, and 47% for a PPF of 8% and test rates from f = 1%, 5%, and 25%. Later in the epidemic, for April 30, TPRs are 13%, 27%, and 59% (Figure 6.12 (a, b)). That is, the classification results improve as the network model learns about the evolution of the epidemic. The classification results are insensitive to the user base coverage $\tilde{N}/N$: the accuracy of the classification does not change for user bases consisting of neighborhoods in the

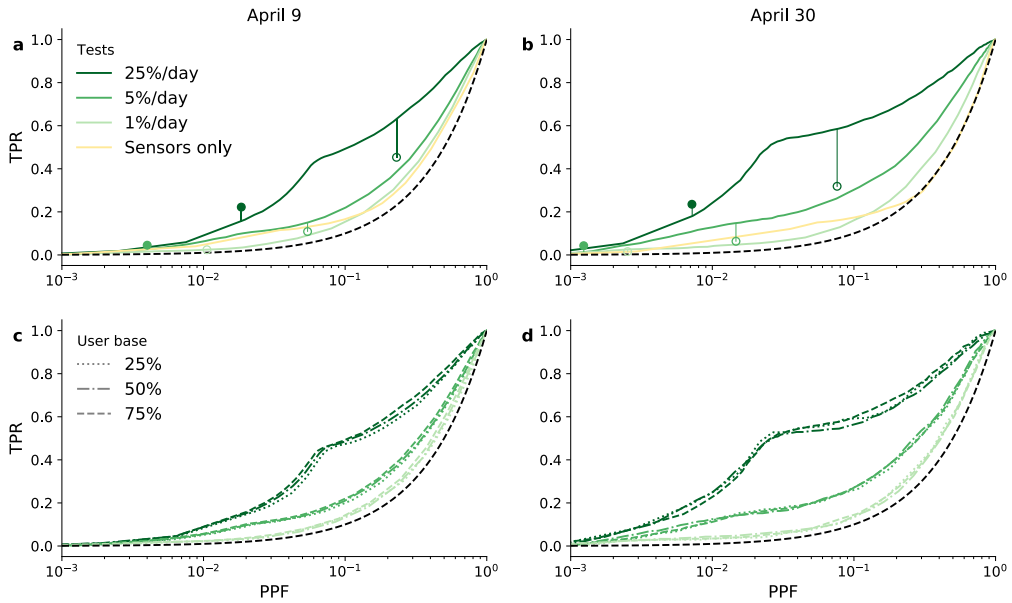


Figure 6.12: ROC curves for classification as possibly infectious.

ROC curves trace out the TPR vs. PPF as the classification threshold is varied. TPR and PPF are given relative to the user base size $\tilde{N}$. Green shades of the ROC curves from lighter to darker correspond to increasing diagnostic test rates f. Left column for April 9; right column for April 30. (a, b) For the ideal user base of $\tilde{N}/N = 100\%$. For comparison, the filled circles are for a test-only scenario when only users with positive diagnostic tests are classified as positive. (The 1%/day case falls outside the plotting region; values for panel (a) are $(7\times10^{-4}, 0.008)$ and for (b) are $(2\times10^{-4}, 0.01)$.) The open circles are for a contact-tracing scenario in which additionally prior close contacts of users with positive diagnostic tests are classified as positive. Also shown is a sensors-only scenario in which 75% of the user base is assumed to provide daily body temperature readings. (c, d) For user bases consisting of neighborhoods in the network covering 25%, 50%, and 75% of the total population (Figure 6.9), with the same test rates f in shades of green as in (a, b). The black dashed line represents a random classifier that provides a lower bound on performance.

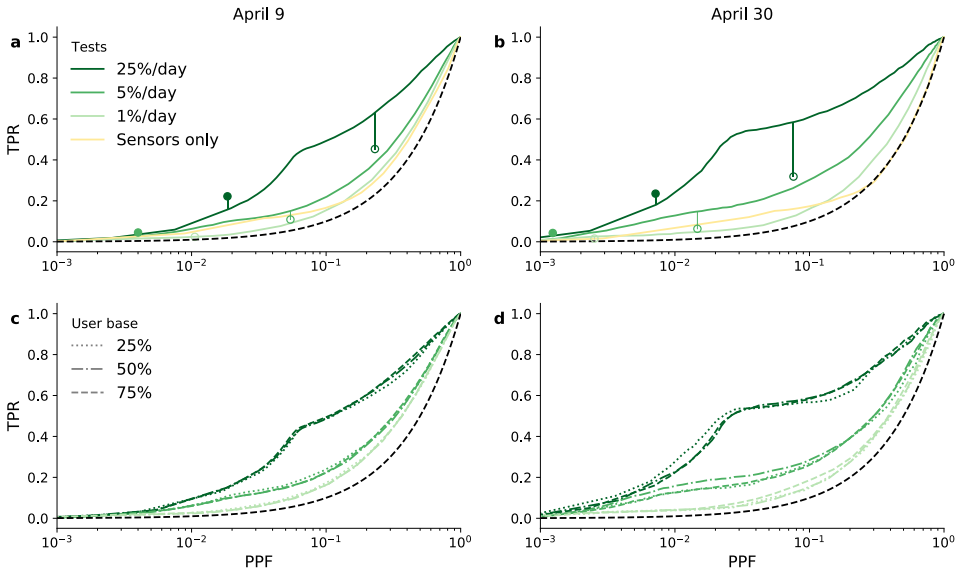


Figure 6.13: Receiver operating characteristic (ROC) curves for classification as possibly infectious.

As in Figure 6.12, but for subnetworks with randomly selected nodes rather than for subnetworks with a neighborhood topology. For the filled circles, the 1%/day case falls outside the plotting region; values for panel (a) are $(7 \times 10^{-4}, 0.009)$ and for panel (b) are $(2 \times 10^{-4}, 0.01)$.

network covering between 25% and 100% of the total population, even though the scenarios with more limited user bases only use contact information for the users, not for non-users (Figure 6.12 (c, d)). The results are also insensitive to the user base topology (Figure 6.9): classification performance is not substantially affected whether the user base consists of neighborhoods in the total population network (Figure 6.12) or of randomly selected nodes (Figure 6.13).

To put these results in context, compare them with the following two traditional approaches:

- If only users with positive diagnostic tests are classified as positive, TPRs reach 0.8%, 4%, and 22% for test rates f = 1%, 5%, and 25%, respectively, with PPFs 0.07%, 0.4%, and 1.8% on April 9 and corresponding TPRs of 1%, 4%, and 23% with PPFs 0.02%, 0.1%, and 0.7% on April 30 (Figure 6.12 (a, b), solid circles). This test-only TPR is close to but slightly smaller than f because the test sensitivity is less than 100%. Classification by network DA can achieve much higher TPRs than testing alone, especially at low test rates, at the expense of increased but still modest PPFs.

- Contact tracing and exposure notification apps classify as positive users with positive diagnostic tests, plus their potentially exposed nearest neighbors on the network. If, following standard contact tracing protocols, individuals are classified as positive if, over the 10 days preceding the diagnosis, they had at least one contact of more than 15 minutes length with a user who had a positive diagnostic test, the so-obtained contact-tracing TPRs for April 9 are 2.4%, 11%, and 45% for test rates from f = 1%, 5% and 25%, with PPFs of 1%, 5% and 23% (Figure 6.12 (a, b), open circles). For April 30 the corresponding TPRs are 2%, 6%, and 32% with PPFs 0.2%, 1.5%, and 7.6%. Network DA exploits the same data as contact tracing and exposure notification apps but achieves substantially higher TPRs at the same PPF. For example, at the same PPF as contact tracing, network DA achieves about a 40% higher TPR than contact tracing for April 9, and about a 100% (factor 2) higher TPR for April 30 (Figure 6.12 (a, b), vertical lines above open circles). That is, risk network in this synthetic example exploits the exact same data as contact tracing or exposure notification apps, but it does so much more effectively.

Risk network can also be used to assess quantitatively to what extent lower-fidelity data can improve classification. As an example, we conducted a set of experiments in which 75% of the users were assumed to report body temperatures daily—for example, with wearable sensors [83]—with infectiousness indicated by elevated temperature readings with 20% sensitivity [16]. Such temperature readings improve the classification when no or few (f = 1%) diagnostic tests are available; however, they do not provide a substantial benefit when f = 5% of the user base or more can be tested daily (Figure 6.12 (a, b)). Nonetheless, if widely

adopted, temperature sensors can provide a modest benefit when diagnostic testing capacity is low [83].

The results show that risk network allows identification of a large fraction of infectious individuals, provided widespread testing is available. The improved identification of infectious individuals over traditional methods is insensitive to the fraction of the population covered by the user base, to the user base topology, and to stochastic variability of the epidemic. The risk network maodel extends classification beyond the nearest network neighbors on which contact tracing and exposure notification apps focus. This gives it an advantage especially when testing capacity is limited.

The capability of risk network to identify infectious individuals can be used to tailor individualized contact interventions for epidemic management and control. For epidemic management and control to be effective, however, it is important not only that the classification accuracy is high but also that the user base coverage is sufficiently large so that a large fraction of infectious individuals can be identified in the population, rather than just within the user base.

6.5.3 *Risk-tailored contact interventions*

The individual risk assessments can be used to prompt those who are classified as possibly infectious for contact interventions. As an illustrative example of such individual contact interventions, we assume that users of the app self-isolate by reducing their contact rate with others by 91%, to an average of 4 contacts per day, during the time when they are classified as positive and 5 days thereafter; all others in the population, whether app users or not, do not change their behavior. As a baseline for comparison, we present TTI scenarios with the same contact rate reduction but continuing over 14 days after diagnosis or identification as possibly exposed through contact with an infectious individual. For this baseline TTI scenario, an individual is classified as exposed if over the preceding 10 days, they had at least one contact lasting more than 15 minutes with an individual who had a positive diagnostic test; that is, the contact trace stage of this baseline TTI emulates techniques used in exposure notification apps, relying on the same data as those available for the risk network model in our synthetic examples. For a direct and fair comparison with risk network, TTI compliance is assumed to be confined to the user base. We use uniform testing regimes with test rates $f = 1\%$, 5%, and 25% within the user base. As classification threshold, we choose a fixed threshold $c_I = 1\%$, resulting in TPR $\gtrsim 40\%$ and PPF $\lesssim 9\%$ when contact interventions commence. Choosing the classification threshold c_I adaptively, in response to current prevalence of infectiousness in the population, may further improve the results.

In the idealized but unrealistic case with full user base coverage ($\tilde{N}/N = 100\%$), the epidemic is more strongly suppressed with the risk network interventions than

in the lockdown scenario, with 50–70% fewer cumulative deaths (Figure 6.14). However, whereas in the lockdown scenario the entire population has reduced contacts, with the risk network model only a small fraction of the population self-isolates. The self-isolation fraction has an initial peak of 15–17% for about a week and then falls quickly to 5–10%, with damped relaxation oscillations over several weeks in the case with lower test rates ($f = 5\%$); 50% of those who isolate do so for 7 days or less, and 90% for 14 days or less. That is, in this idealized case, risk-tailored self-isolation achieves effective epidemic control with isolation of only a small fraction of the population at a time. Risk network does not squash daily infections to zero, because the classification threshold c_I was chosen as a compromise between wanting a reliable classification with a high TPR while avoiding isolation of a too large fraction of the population with a too high PPF (Figure 6.12). For comparison, TTI with 100% compliance does not achieve epidemic control at a test rate $f = 1\%$; at a test rate $f = 5\%$, cumulative deaths are 3 times higher than with network DA because TTI misses more infections than risk network. At the test rate $f = 25\%$, the cumulative death rate with TTI is comparable to or lower than with risk network, but at the expense of a 2–5 times higher isolated fraction of the population. Whereas the performance of TTI is strongly test-rate dependent, that of network DA is less sensitive to test rate, and it is always more efficient than TTI.

In the somewhat more realizable case with $\tilde{N}/N = 75\%$ user base coverage, we simulate a demanding scenario in which testing and contact interventions are confined to the user base; no contact information among non-users is harnessed, and non-users maintain their contact patterns without isolation. In this case, risk-tailored self-isolation still achieves epidemic control at all test rates of $f = 1\%$, 5%, and 25% within the user base (Figure 6.15), and attains a cumulative death rate similar to the 100% user base. The fraction of the population in isolation again peaks at just over 15% initially and then drops to 5–10%. As before, TTI with 75% compliance and with the highest test rates ($f = 25\%$) also achieves epidemic control, but with a higher isolated fraction of the population. At the test rate $f = 5\%$, TTI results in an about four times higher cumulative death rate than isolation tailored by network DA, which additionally isolates fewer individuals. TTI fails to achieve epidemic control at a test rate $f = 1\%$.

With a further reduced user base coverage of $\tilde{N}/N = 50\%$, classification remains accurate, and isolation tailored by network DA can still achieve epidemic control and can remain more effective than a lockdown in preventing infections and deaths (Figure 6.19). The initial fraction of the population in isolation increases to around 30%, and then drops again to between 5–10%. However, this means that initially, the majority of the user base (50% of the population) is in isolation, which creates perverse incentives: it effectively puts the user base, but not others, in a lockdown. TTI with 50% compliance fails to control the

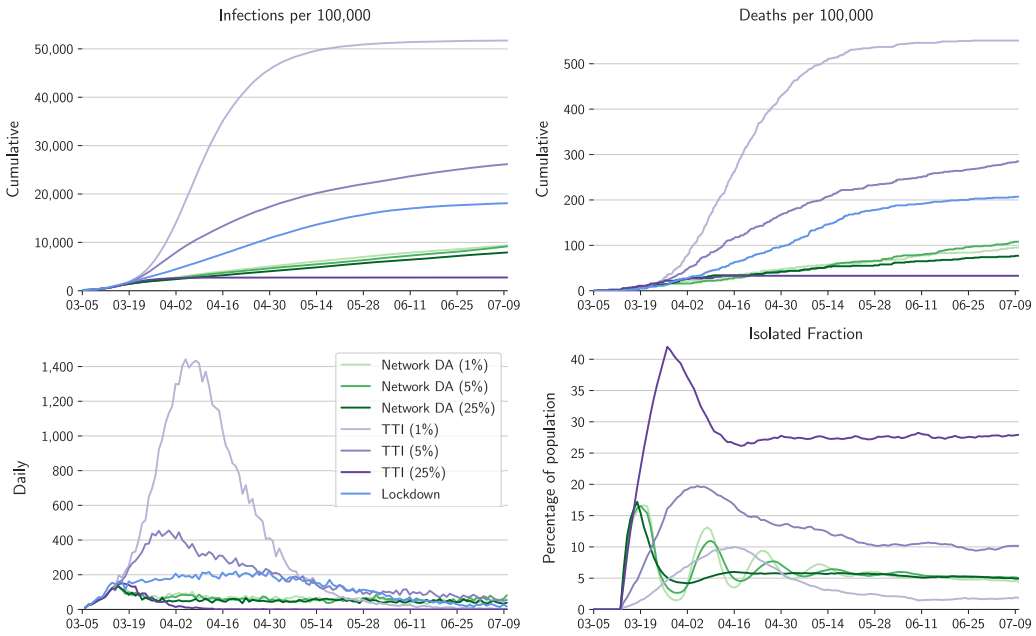


Figure 6.14: Comparison of different contact intervention scenarios for full user base with $\tilde{N}/N = 100\%$.

Shown are the lockdown scenario (blue) from Fig. 6.10, the results of network DA and isolation of positive individuals for test rates $f = 1\%$, 5%, and 25% (greens), and the results of TTI with the same test rates as for network DA (purples).

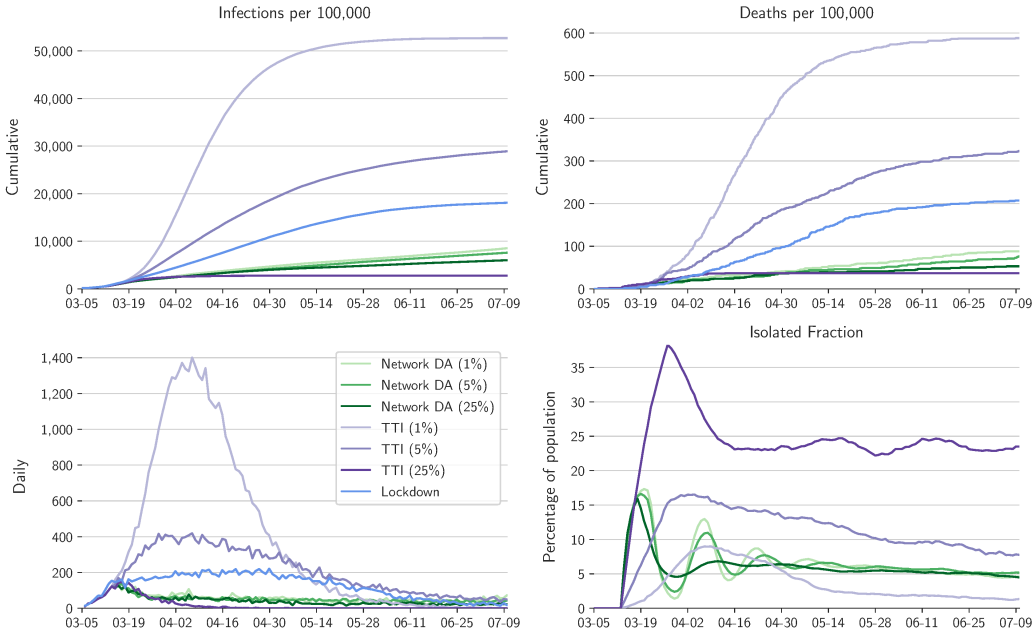


Figure 6.15: Comparison of different contact intervention scenarios for a user base with $\tilde{N}/N = 75\%$.

Plotting conventions as in Fig. 6.14. TTI here is confined to the same user base as network DA, implying 75% compliance.

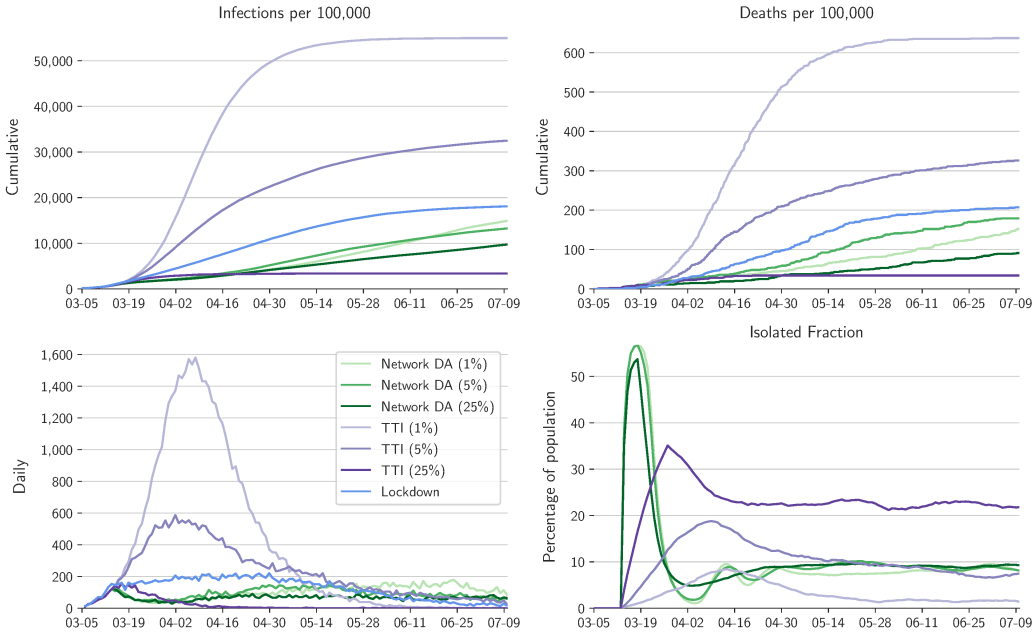


Figure 6.16: Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 75\%$.

As in Figure 6.15, but with a subnetwork with randomly selected nodes and with a classification threshold $c_I = 0.25\%$.

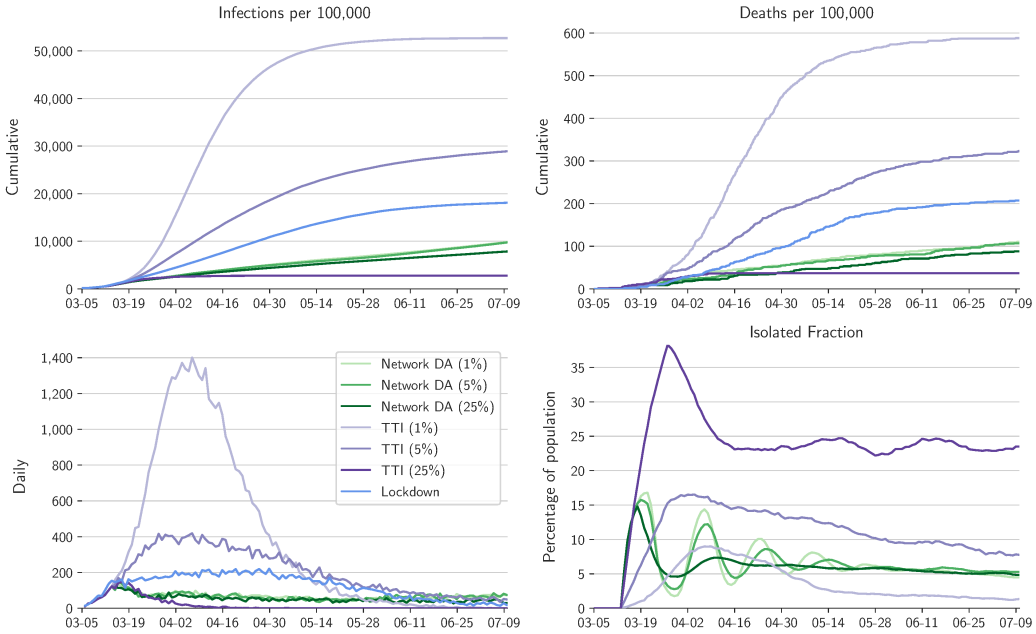


Figure 6.17: As in Figure 6.15, but with constant exterior connectivity.

Comparison of different contact intervention scenarios for neighborhood user base with $\tilde{N}/N = 75\%$ and with a classification threshold $c_I = 1\%$, but replacing the user-dependent number of external neighbours k_i^x by the constant exterior connectivity from Table 6.1.

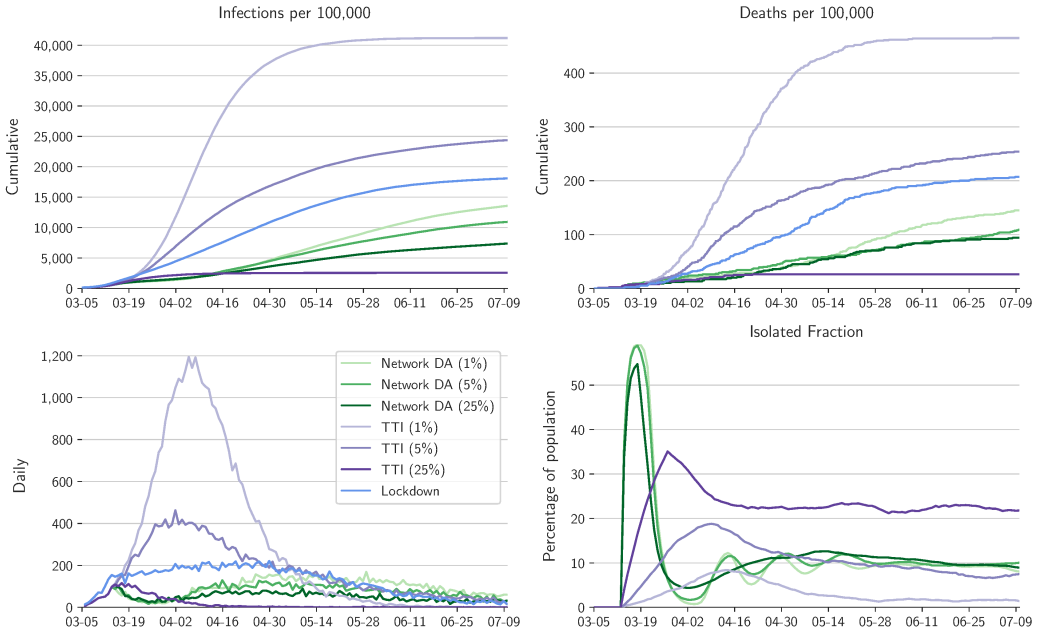


Figure 6.18: As in Figure 6.16, but with constant exterior connectivity.

Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 75\%$ and with a classification threshold $c_I = 0.25\%$, but replacing the user-dependent number of external neighbours k_i^x by the constant exterior connectivity from Table 6.1.

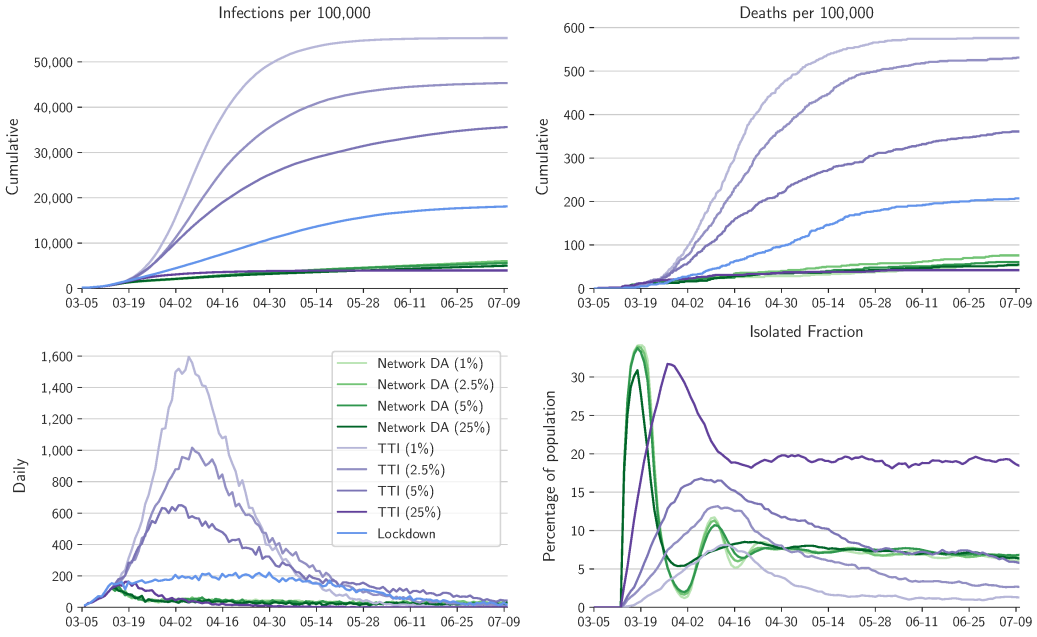


Figure 6.19: Comparison of different contact intervention scenarios for neighborhood user base with $\tilde{N}/N = 50\%$ and with a classification threshold $c_I = 0.5\%$.

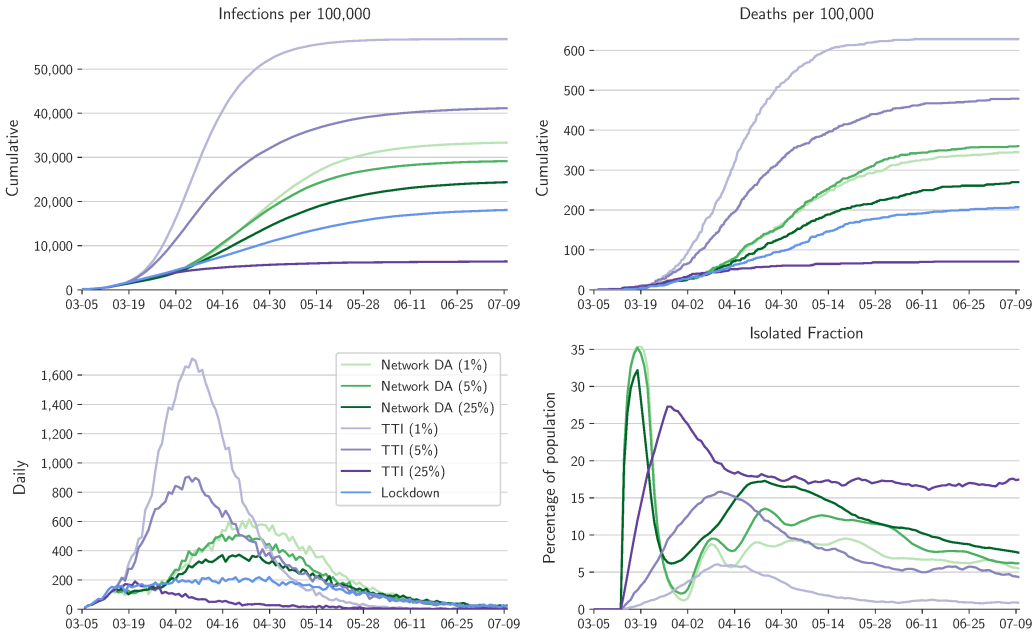


Figure 6.20: Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 50\%$ and with a lower classification threshold $c_I = 0.25\%$.

As in Figure 6.19, but with a subnetwork with randomly selected nodes.

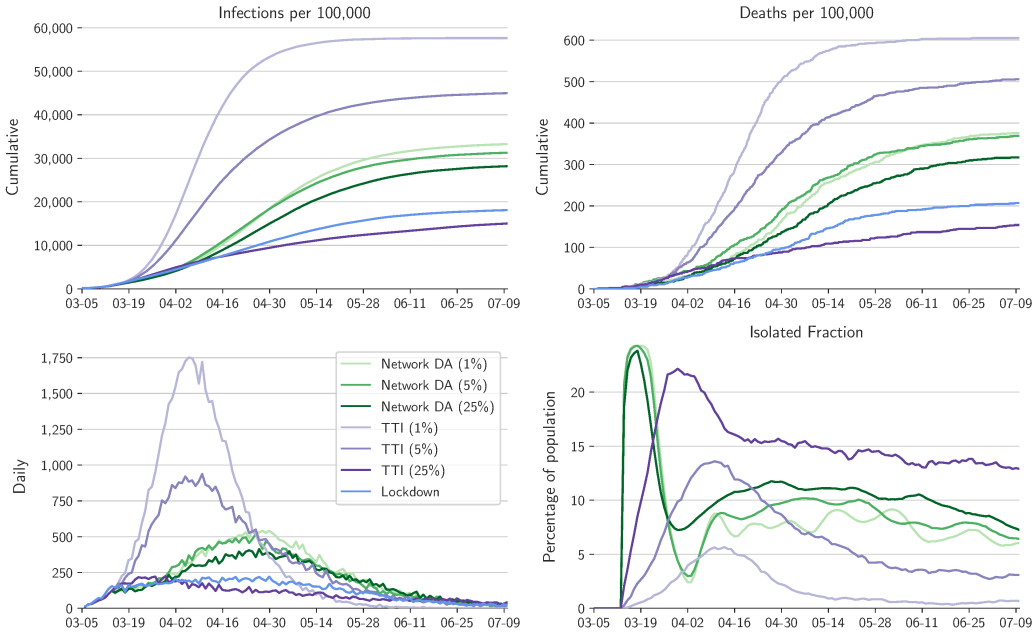


Figure 6.21: Comparison of different contact intervention scenarios for neighborhood user base with $\tilde{N}/N = 25\%$ and with a classification threshold $c_I = 0.25\%$.

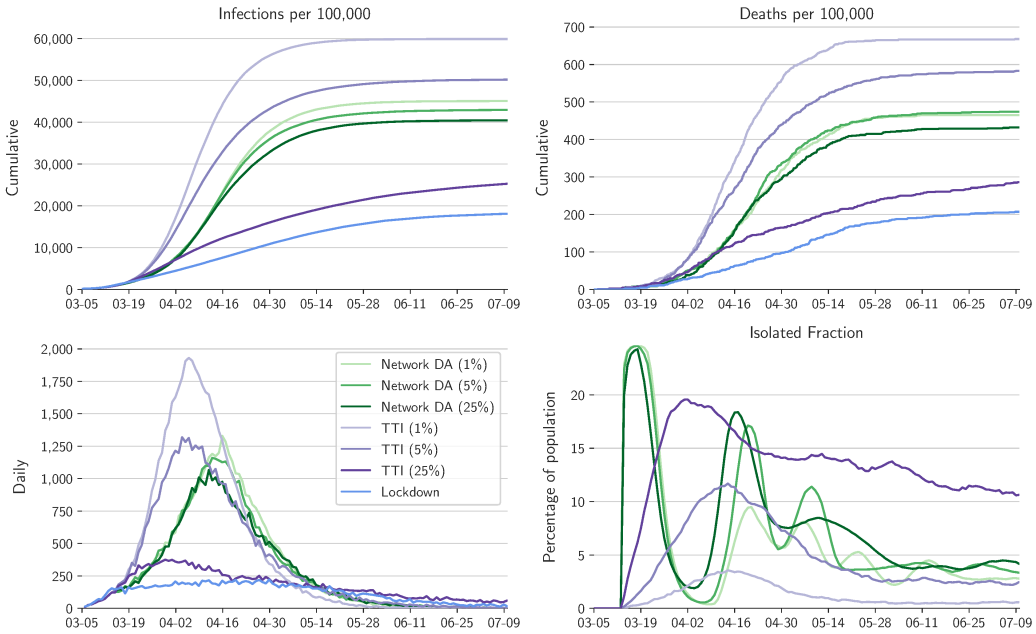


Figure 6.22: Comparison of different contact intervention scenarios for random user base with $\tilde{N}/N = 25\%$ and with a lower classification threshold $c_I = 0.01\%$.

As in Figure 6.21, but with a subnetwork with randomly selected nodes.

epidemic for test rates below $f = 5\%$ but still achieves some control at $f = 25\%$, albeit with a higher isolated population fraction than with network DA.

For the yet smaller user base coverage of $\tilde{N}/N = 25\%$, classification remains accurate (Figure 6.12); however, here the dominance of non-users within the population, who do not isolate, rules out epidemic control (Figure 6.21). As with any epidemic management measure, control cannot be achieved with low compliance rates.

These results for reduced user bases are for sub-networks consisting of neighborhoods in the overall population network. Results for user bases consisting of nodes selected at random from the overall population are qualitatively similar for $\tilde{N}/N = 75\%$, albeit with an adjusted classification threshold and a higher fraction of the population in isolation (Figure 6.16). For $\tilde{N}/N = 50\%$ with a random user base, risk network, while still being able to identify a large fraction of infectious individuals in the user base (Figure 6.13), ceases to be effective for epidemic control (Figure 6.20); similar behavior is observed in the $\tilde{N}/N = 25\%$ case. That is, while network topology was rather unimportant for the accuracy of classification, it does play a role for the effectiveness of epidemic management and control strategies. It is possible the performance of the risk network in managing the epidemic may be improved with data-adaptive classification thresholds.

In scenarios in which the user base and/or test rates are too small to achieve epidemic control, there is still a pronounced reduction in the cumulative death

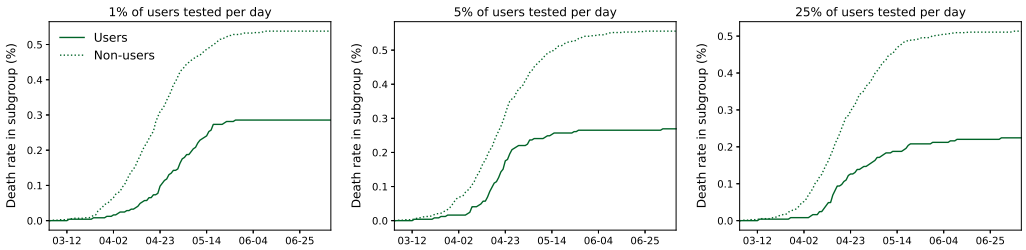


Figure 6.23: Cumulative death rate of users vs. non-users for the $\tilde{N}/N = 25\%$ user base consisting of nodes selected at random from the overall population network.

Individual contact interventions are applied within the user base from March 15 onward.

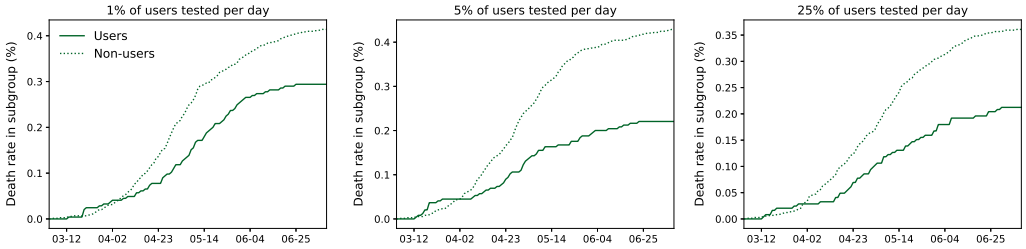


Figure 6.24: Cumulative death rate of users vs. non-users for the $\tilde{N}/N = 25\%$ user base consisting of neighborhoods in the overall population network.

Individual contact interventions are applied within the user base from March 15 onward.

rate of users relative to the general non-user population (Figure 6.24). For test rates $f = 1\%$, 5%, and 25% per day within the $\tilde{N}/N = 25\%$ user base consisting of neighborhoods in the overall population network, the cumulative death rate is respectively 29%, 48%, and 42% lower than the death rate among non-users. Additionally, although the contact interventions are confined to the user base, the death rate in the non-user population is still reduced by about 50% compared with the no-intervention scenario (Figure 6.10). For a $\tilde{N}/N = 25\%$ user base consisting of nodes selected at random, the results are qualitatively similar: Death rates among users relative to non-users are reduced by 47%, 52%, and 56% for respective test rates $f = 1\%$, 5%, and 25% (Figure 6.23).

That is, risk-tailored isolation on the basis of risk network generally outperforms TTI as an epidemic management and control approach when both are presented with the same contact and test data. Even when it does not achieve epidemic control because of low compliance rates, it still offers advantages to users in terms of reduced death rates.

6.5.4 *Discussion*

We have demonstrated a platform concept for individual health risk assessment, which exploits the same proximity data from mobile devices that exposure notification apps rely upon but is substantially better at identifying infectious individuals. It achieves these gains by assimilating crowdsourced data from diverse sources into an epidemiological model defined on a contact network. The risk network model provides informative and actionable risk assessments for individuals, even when only a modest fraction of the population uses the app necessary to obtain proximity data. The accuracy of the risk assessments is largely independent of the fraction of the population using the platform and of the user base topology; it improves with increasing diagnostic test rates, as should be expected.

When the user base is sufficiently large (covering around 75% of the population), the platform can be used to tailor interventions that are more efficient for epidemic management and control than lockdowns or TTI. For example, with a user base covering 75% of the population and users tested every 20 days, simulations for NYC showed that risk-tailored self-isolation achieves epidemic control with 63% fewer deaths than during NYC's lockdown, with typically only 5–10% of the population in isolation at any given time. This risk-tailored isolation approach is more effective at preventing infections and deaths than a TTI approach that uses the same contact and diagnostic test data. Our experiments were solely based on self-isolation among app users, without considering other public health interventions. As a result, 75% coverage may be a conservative estimate. In reality, multiple non-pharmaceutical interventions will likely be employed simultaneously at the population level, which may reduce the user coverage required to achieve epidemic control.

We have produced a modular codebase that allows for exploration and benchmarking of tools to manage and control epidemics in a synthetic setting. To validate and further optimize our choices of diagnostic test and intervention strategies, further analyses are required. For example, our results may be improved by the inclusion of additional information from the contact network or more data-adaptive use of the risk assessments provided by the risk network model. Additionally, it is possible to learn about the model parameters that appear in the network epidemiology model; we have only skimmed the surface with respect to what is possible in this regard, so far with limited success. Further investigation to delineate which model parameters are identifiable from data would be beneficial.

The platform has a relatively low barrier to widespread implementation. It can be realized by expanding the computational backend of existing exposure notification apps. High-precision proximity data are now available through Bluetooth protocols [8], and lower-precision location data from mobile devices have been exploited commercially for some time. Statistical techniques may be required to optimize the reconstruction of contact networks from such proximity data in practical implementations with imperfect knowledge of contact patterns [99]. To be effective, the platform requires that users provide proximity data and other crowdsourced data, such as test results and reports of clinical symptoms. The more detailed data users make available, the more accurate and detailed risk profiles can be produced in return. Uptake rates of exposure notification apps have already reached up to 75% in some urban areas, as in our simulated scenarios (e. g., more than 90% of Singapore's population over 6 years of age [43] is using an exposure notification app). Uptake rates on a national scale so far have been more modest (e. g., a third of the UK population [117]), in part, for example, because of rural-urban digital divides but also, probably, because of the limited information provided by current exposure notification apps. However, smartphone usage rates worldwide are around 50% and continue to grow rapidly [101]; thus, widespread use of a risk-network-type model in future epidemics will become technically possible. And while routine surveillance test rates in much of the world are still low, more widespread surveillance testing on the scale of major cities or regions at this point is feasible; for example, NYC currently is already testing up to roughly 2.5% of its population daily [105]. Our conclusions provide further evidence of the benefits of widespread testing, especially when that is combined with the risk network to spread the test information over dynamic contact networks assembled from proximity data.

Challenges to widespread and successful adoption of a network DA platform center around equity, compliance, and privacy questions. Smartphone use is not equitably distributed within the population, and there are disincentives (e. g., unavailability of sick leave) to comply with individual contact interventions. Conversely, classification of users as "low risk" may encourage risky and

counterproductive behavior. It is also unknown, and we did not address, how correlations between smartphone use, compliance, and factors influencing infection risk would affect our results. An additional impediment to widespread adoption of network DA are concerns about protecting users' privacy. The network DA platform requires data to be transferred temporarily to a central computing facility for data assimilation [12]. This makes the platform more difficult to harden against malicious exploitation than exposure notification apps, which only require central data exchange when there is direct evidence of an infection [48]. Nonetheless, the data need not be stored beyond a data assimilation window that is at most a few days long. Additionally, the platform requires only anonymized proximity data but not absolute location data, and it does not rely on humans in the loop, reducing risks of malicious exploitation. There may be ways to harden the platform itself and the data exchange with users against privacy breaches [111].

The network DA platform provides obvious benefits in managing and controlling epidemics, for example, in reducing the need for lockdowns while preventing infections and deaths, and in providing users tools to manage their personal risks. It provides a scalable alternative to manual TTI programs, and a backend that delivers more accurate and actionable information than current digital TTI and exposure notification programs developed by many governments [43, 72]. The effectiveness of such programs has been modelled [36, 47, 55], but their impact in practice is only beginning to be elucidated [117]. Given that many TTI programs are voluntary, and documentation of contacts in manual programs is subjective, it will be important to compare both the control and cost effectiveness of manual and digital trace programs with the more objective and automated network DA approach presented here.

In addition to its health impacts, the COVID-19 pandemic has exacted an enormous economic toll on countries throughout the world [25, 54]. There is a continuing need to identify approaches that precisely and effectively control epidemics while minimizing economic disruption. With sufficient uptake and testing, the platform described here provides a means for achieving these dual aims.

BIBLIOGRAPHY

[1] Alberto Aleta et al. "Modelling the impact of testing, contact tracing and household quarantine on second waves of COVID-19." In: *Nature Human Behaviour* 4.9 (Sept. 2020), pp. 964–971. ISSN: 2397-3374. DOI: 10.1038/s41562-020-0931-9.

[2] Romeo Alexander and Dimitrios Giannakis. "Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques." In: *Physica D: Nonlinear Phenomena* 409 (Aug. 2020). ISSN: 0167-2789. DOI: 10.1016/j.physd.2020.132520.

[3] D. Allen et al. *Roadmap to Pandemic Resilience: Massive Scale Testing, Tracing, and Supported Isolation (TTSI) as the Path to Pandemic Resilience for a Free Society*. Tech. rep. Edmond J. Safra Center for Ethics at Harvard University, Apr. 2020.

[4] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Alejandro Lage-Castellanos, and Riccardo Zecchina. "Bayesian Inference of Epidemics on Networks via Belief Propagation." In: *Physical Review Letters* 112.11 (Mar. 2014), p. 118701. DOI: 10.1103/PhysRevLett.112.118701.

[5] Jeffrey L. Anderson. "An Ensemble Adjustment Kalman Filter for Data Assimilation." In: *Monthly Weather Review* 129.12 (2001), pp. 2884–2903. DOI: 10.1175/1520-0493(2001)129<2884:AEAKFF>2.0.CO;2.

[6] Jeffrey L. Anderson. "Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation." In: *Monthly Weather Review* 140.7 (July 2012), pp. 2359–2371. DOI: 10.1175/MWR-D-11-00013.1.

[7] Muthuvel Chelliah Anthony G. Bamston and Stanley B. Goldenberg. "Documentation of a Highly ENSO-Related SST Region in the Equatorial Pacific: Research note." In: *Atmosphere-Ocean* 35.3 (1997), pp. 367–383. DOI: 10.1080/07055900.1997.9649597.

[8] Apple/Google. *Privacy-Preserving Contact Tracing*. 2020. URL: https://www.apple.com/covid19/contacttracing.

[9] Hassan Arbabi and Igor Mezic. "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator." In: *SIAM Journal on Applied Dynamical Systems* 16.4 (2017), pp. 2096–2126.

[10] Alex Arenas, Wesley Cota, Jesús Gómez-Gardeñes, Sergio Gómez, Clara Granell, Joan T. Matamalas, David Soriano-Paños, and Benjamin Steinegger. "Modeling the Spatiotemporal Epidemic Spreading of COVID-19 and the Impact of Mobility and Social Distancing Interventions." In: *Physical Review X* 10.4 (Dec. 2020), p. 041055. DOI: 10.1103/PhysRevX.10.041055.

[11] Peter Bauer, Alan Thorpe, and Gilbert Brunet. "The quiet revolution of numerical weather prediction." In: *Nature* 525.7567 (Sept. 2015), pp. 47–55. ISSN: 1476-4687. DOI: 10.1038/nature14956.

[12] Gabrielle Berman, Karen Carter, Manuel Garcia Herranz, and Vedran Sekara. *Digital contact tracing and surveillance during COVID-19: General and child-specific ethical issues*. 2020-01. Innocenti, Florence: UNICEF Office of Research, 2020.

[13] Tyrus Berry and John Harlim. "Variable Bandwidth Diffusion Kernels." In: *Appl. Comput. Harmon. Anal.* 40.1 (2016), pp. 68–96. DOI: 10.1016/j.acha.2015.01.001.

[14] Andrea L. Bertozzi, Elisa Franco, George Mohler, Martin B. Short, and Daniel Sledge. "The challenges of modeling and forecasting the spread of COVID-19." In: *Proceedings of the National Academy of Sciences* 117.29 (2020), pp. 16732–16738. DOI: 10.1073/pnas.2006520117.

[15] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. "Accurate medium-range global weather forecasting with 3D neural networks." In: *Nature* 619.7970 (2023), pp. 533–538. DOI: 10.1038/s41586-023-06185-3.

[16] Michel Bielecki, Giovanni Andrea Gerardo Crameri, Patricia Schlagenhauf, Thomas Werner Buehrer, and Jeremy Werner Deuel. "Body temperature screening to identify SARS-CoV-2 infected young adult travellers is ineffective." In: *Travel Medicine and Infectious Disease* 37 (Aug. 2020), p. 101832. DOI: 10.1016/j.tmaid.2020.101832.

[17] Jan M. Brauner et al. "Inferring the effectiveness of government interventions against COVID-19." In: *Science* 371.6531 (2021), eabd9338. DOI: 10.1126/science.abd9338.

[18] Chloë Brown, Anastasios Noulas, Cecilia Mascolo, and Vincent Blondel. "A place-focused model for social networks in cities." In: *2013 International Conference on Social Computing*. 2013, pp. 75–80. DOI: 10.1109/SocialCom.2013.18.

[19] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. "Modern Koopman Theory for Dynamical Systems." In: *SIAM Review* 64.2 (2022), pp. 229–340. DOI: 10.1137/21M1401243.

[20] Christopher J. Burant. "A Methodological Note: An Introduction to Autoregressive Models." In: *The International Journal of Aging and Human Development* 95.4 (2022), pp. 516–522. DOI: 10.1177/00914150211066554.

[21] Dmitry Burov, Dimitrios Giannakis, Krithika Manohar, and Andrew Stuart. "Kernel Analog Forecasting: Multiscale Test Problems." In: *Multiscale Modeling & Simulation* 19.2 (2021), pp. 1011–1040. DOI: 10.1137/20M1338289.

D. B. wrote numerical code for several dynamical systems and Gaussian-process closures, performed simulations and provided theoretical explanations, and helped write Sections 4 and 5 (Averaging and Conclusions).

[22] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. "A Limited Memory Algorithm for Bound Constrained Optimization." In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[23] G. Cencetti, G. Santin, A. Longa, E. Pigani, A. Barrat, C. Cattuto, S. Lehmann, M. Salathé, and B. Lepri. "Digital proximity tracing on empirical contact networks for pandemic control." In: *Nature Communications* 12.1 (Mar. 2021), p. 1655. ISSN: 2041-1723. DOI: 10.1038/s41467-021-21809-w.

[24] Centers for Disease Control and Prevention. *Contact Tracing for COVID-19*. 2021. URL: https://cdc.gov/coronavirus/2019-ncov/php/contact-tracing/COVIDTracerTools.html.

[25] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. "Mobility network models of COVID-19 explain inequities and inform reopening." In: *Nature* 589.7840 (Jan. 2021), pp. 82–87. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2923-3.

[26] Ronald R. Coifman and Matthew J. Hirn. "Bi-stochastic kernels via asymmetric affinity functions." In: *Applied and Computational Harmonic Analysis* 35.1 (2013), pp. 177–180. DOI: 10.1016/j.acha.2013.01.001.

[27] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps." In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[28] Sebastian Contreras, Jonas Dehning, Matthias Loidolt, Johannes Zierenberg, F. Paul Spitzner, Jorge H. Urrea-Quintero, Sebastian B. Mohr, Michael Wilczek, Michael Wibral, and Viola Priesemann. "The challenges of containing SARS-CoV-2 via test-trace-and-isolate." In: *Nature Communications* 12.1 (Jan. 2021), p. 378. ISSN: 2041-1723. DOI: 10.1038/s41467-020-20699-8.

[29] Leon Danon, Thomas A. House, Jonathan M. Read, and Matt J. Keeling. "Social encounter networks: collective properties and disease transmission." In: *Journal of The Royal Society Interface* 9.76 (2012), pp. 2826–2833. DOI: 10.1098/rsif.2012.0357.

[30] Audrey Duval, Thomas Obadia, Lucie Martinet, Pierre-Yves Boëlle, Eric Fleury, Didier Guillemot, Lulla Opatowski, Laura Temime, and I-Bird study group. "Measuring dynamic social contacts in a rehabilitation hospital: effect of wards, patient and staff characteristics." In: *Scientific Reports* 8.1 (Jan. 2018), p. 1686. ISSN: 2045-2322. DOI: 10.1038/s41598-018-20008-w.

[31] Akira Endo, Centre for the Mathematical Modelling of Infectious Diseases COVID-19 Working Group, Quentin J. Leclerc, Gwenan M. Knight, Graham F. Medley, Katherine E. Atkins, Sebastian Funk, and Adam J. Kucharski. "Implication of backward contact tracing in the presence of overdispersed transmission in COVID-19 outbreaks." In: *Wellcome Open Research* 5 (Mar. 2021), p. 239. DOI: 10.12688/wellcomeopenres.16344.3.

[32] Expii, Inc. *NOVID*. 2020. URL: https://www.novid.org/.

[33] Ibrahim Fatkullin and Eric Vanden-Eijnden. "A computational strategy for multiscale systems with applications to Lorenz 96 model." In: *Journal of Computational Physics* 200.2 (2004), pp. 605–638.

[34] Neil M. Ferguson, Derek A. T. Cummings, Simon Cauchemez, Christophe Fraser, Steven Riley, Aronrag Meeyai, Sopon Iamsirithaworn, and Donald S. Burke. "Strategies for containing an emerging influenza pandemic in Southeast Asia." In: *Nature* 437.7056 (Sept. 2005), pp. 209–214. ISSN: 1476-4687. DOI: 10.1038/nature04017.

[35] Neil M. Ferguson, Derek A. T. Cummings, Christophe Fraser, James C. Cajka, Philip C. Cooley, and Donald S. Burke. "Strategies for mitigating an influenza pandemic." In: *Nature* 442.7101 (July 2006), pp. 448–452. ISSN: 1476-4687. DOI: 10.1038/nature04795.

[36] Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing." In: *Science* 368.6491 (2020), eabb6936. DOI: 10.1126/science.abb6936.

[37] Julie Fournet and Alain Barrat. "Contact Patterns among High School Students." In: *PLOS ONE* 9.9 (Sept. 2014), e107878. DOI: 10.1371/journal.pone.0107878.

[38] Peter R. Gent et al. "The Community Climate System Model Version 4." In: *Journal of Climate* 24.19 (2011), pp. 4973–4991. DOI: 10.1175/2011JCLI4083.1.

[39] Dimitrios Giannakis. "Data-driven spectral decomposition and forecasting of ergodic dynamical systems." In: *Applied and Computational Harmonic Analysis* 47.2 (2019), pp. 338–396. ISSN: 1063-5203. DOI: 10.1016/j.acha.2017.09.001.

[40] Dimitrios Giannakis and Joanna Slawinska. "Indo-Pacific Variability on Seasonal to Multidecadal Time Scales. Part II: Multiscale Atmosphere–Ocean Linkages." In: *Journal of Climate* 31.2 (2018), pp. 693–725. DOI: 10.1175/JCLI-D-17-0031.1.

[41] Daniel T. Gillespie. "Exact stochastic simulation of coupled chemical reactions." In: *The Journal of Physical Chemistry* 81.25 (1977), pp. 2340–2361. DOI: 10.1021/j100540a008.

[42] James P. Gleeson, Sergey Melnik, Jonathan A. Ward, Mason A. Porter, and Peter J. Mucha. "Accuracy of mean-field theory for dynamics on real-world networks." In: *Physical Review E* 85.2 (Feb. 2012), p. 026106. DOI: 10.1103/PhysRevE.85.026106.

[43] Government of Singapore. *TraceTogether*. 2022. URL: www.tracetogether.gov.sg.

[44] Mark S. Handcock and James Holland Jones. "Likelihood-based inference for stochastic models of sexual network formation." In: *Theoretical Population Biology* 65.4 (2004), pp. 413–422. ISSN: 0040-5809. DOI: 10.1016/j.tpb.2003.09.006.

[45] Nils Haug, Lukas Geyrhofer, Alessandro Londei, Elma Dervic, Amélie Desvars-Larrive, Vittorio Loreto, Beate Pinior, Stefan Thurner, and Peter Klimek. "Ranking the effectiveness of worldwide COVID-19 government interventions." In: *Nature human behaviour* 4.12 (2020), pp. 1303–1312.

[46] Xi He et al. "Temporal dynamics in viral shedding and transmissibility of COVID-19." In: *Nature Medicine* 26.5 (May 2020), pp. 672–675. ISSN: 1546-170X. DOI: 10.1038/s41591-020-0869-5.

[47] Joel Hellewell et al. "Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts." In: *The Lancet Global Health* 8.4 (Apr. 2020), E488–E496. ISSN: 2214-109X. DOI: 10.1016/S2214-109X(20)30074-7.

[48] Robert Hinch et al. *Effective Configurations of a Digital Contact Tracing App: A report to NHSX*. Apr. 2020. URL: https://cdn.theconversation.com/static_files/files/1009/Report_-_Effective_App_Configurations.pdf.

[49] P. L. Houtekamer and Fuqing Zhang. "Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation." In: *Monthly Weather Review* 144.12 (Dec. 2016), pp. 4489–4532. DOI: 10.1175/MWR-D-15-0440.1.

[50] Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

[51] Brian Karrer and Mark E. J. Newman. "Stochastic blockmodels and community structure in networks." In: *Physical Review E* 83.1 (Jan. 2011), p. 016107. DOI: 10.1103/PhysRevE.83.016107.

[52] István Z. Kiss, Joel C. Miller, and Péter L. Simon. *Mathematics of Epidemics on Networks: From Exact to Approximate Models*. Interdisciplinary Applied Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-50804-7.

[53] Stephen M. Kissler, Christine Tedijanto, Edward Goldstein, Yonatan H. Grad, and Marc Lipsitch. "Projecting the transmission dynamics of SARS-CoV-2 through the postpandemic period." In: *Science* 368.6493 (2020), pp. 860–868. DOI: 10.1126/science.abb5793.

[54] Michael König and Adalbert Winkler. "COVID-19: Lockdowns, Fatality Rates and GDP Growth: Evidence for the First Three Quarters of 2020." In: *Intereconomics* 56.1 (Jan. 2021), pp. 32–39. DOI: 10.1007/s10272-021-0948-y.

[55] Adam J. Kucharski et al. "Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in different settings: a mathematical modelling study." In: *The Lancet Infectious Diseases* 20.10 (Oct. 2020), pp. 1151–1160. ISSN: 1473-3099. DOI: 10.1016/S1473-3099(20)30457-6.

[56] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. "FourCastNet: Accelerating Global High-Resolution Weather Forecasting Using Adaptive Fourier Neural Operators." In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '23. Davos, Switzerland: Association for Computing Machinery, 2023. ISBN: 9798400701900. DOI: 10.1145/3592979.3593412.

[57] Remi Lam et al. "Learning skillful medium-range global weather forecasting." In: *Science* 382.6677 (2023), pp. 1416–1421. DOI: 10.1126/science.adi2336.

[58] Stephen A. Lauer, Kyra H. Grantz, Qifang Bi, Forrest K. Jones, Qulu Zheng, Hannah R. Meredith, Andrew S. Azman, Nicholas G. Reich, and Justin Lessler. "The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and

Application." In: *Annals of Internal Medicine* 172.9 (Mar. 2020), pp. 577–582. DOI: 10.7326/M20-0504.

[59] Dyani Lewis. "Contact-tracing apps help reduce COVID infections, data suggest." In: *Nature* 591.7848 (Mar. 2021), pp. 18–19.

[60] Ruiyun Li, Sen Pei, Bin Chen, Yimeng Song, Tao Zhang, Wan Yang, and Jeffrey Shaman. "Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2)." In: *Science* 368.6490 (2020), pp. 489–493. DOI: 10.1126/science.abb3221.

[61] Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Learning Chaotic Dynamics in Dissipative Systems." In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 16768–16781.

[62] Quan-Hui Liu, Marco Ajelli, Alberto Aleta, Stefano Merler, Yamir Moreno, and Alessandro Vespignani. "Measurability of the epidemic reproduction number in data-driven contact networks." In: *Proceedings of the National Academy of Sciences* 115.50 (2018), pp. 12680–12685. DOI: 10.1073/pnas.1811115115.

[63] Po-Shen Loh. *Flipping the Perspective in Contact Tracing*. 2020. arXiv: 2010.03806.

[64] Edward N. Lorenz. "Deterministic Nonperiodic Flow." In: *Journal of the Atmospheric Sciences* 20.2 (1963), pp. 130–141.

[65] Edward N. Lorenz. "Atmospheric Predictability as Revealed by Naturally Occurring Analogues." In: *Journal of Atmospheric Sciences* 26.4 (1969), pp. 636–646. DOI: 10.1175/1520-0469(1969)26<636:APARBN>2.0.CO;2.

[66] Edward N. Lorenz. "Predictability: A problem partly solved." In: *Proceedings of Seminar on Predictability*. Vol. 1. 1996, pp. 40–58.

[67] Lauren Ancel Meyers, Babak Pourbohloul, Mark E.J. Newman, Danuta M. Skowronski, and Robert C. Brunham. "Network theory and SARS: predicting outbreak diversity." In: *Journal of Theoretical Biology* 232.1 (2005), pp. 71–81. ISSN: 0022-5193. DOI: 10.1016/j.jtbi.2004.07.026.

[68] Dina Mistry et al. "Inferring high-resolution human mixing patterns for disease modeling." In: *Nature Communications* 12.1 (Jan. 2021), p. 323. ISSN: 2041-1723. DOI: 10.1038/s41467-020-20544-y.

[69] Mélodie Monod et al. "Age groups that sustain resurging COVID-19 epidemics in the United States." In: *Science* 371.6536 (2021), eabe8372. DOI: 10.1126/science.abe8372.

[70] Joël Mossong et al. "Social contacts and mixing patterns relevant to the spread of infectious diseases." In: *PLOS Medicine* 5.3 (Mar. 2008), e74. DOI: 10.1371/journal.pmed.0050074.

[71] Arnold Neumaier and Tapio Schneider. "Estimation of parameters and eigenmodes of multivariate autoregressive models." In: *ACM Transactions on Mathematical Software* 27.1 (2001), pp. 27–57. ISSN: 0098-3500. DOI: 10.1145/382043.382304.

[72] New York City. *Test and Trace Corps*. 2021. URL: https://www1.nyc.gov/site/coronavirus/get-tested/test-trace-corps.page.

[73] Mark E. J. Newman. "Spread of epidemic disease on networks." In: *Physical Review E* 66.1 (July 2002), p. 016128. DOI: 10.1103/PhysRevE.66.016128.

[74] NYC Department of Health and Mental Hygiene. *NYC Coronavirus Disease 2019 (COVID-19) Data*. 2020. URL: https://github.com/nychealth/coronavirus-data.

[75] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. "Epidemic processes in complex networks." In: *Reviews of Modern Physics* 87.3 (Aug. 2015), pp. 925–979. DOI: 10.1103/RevModPhys.87.925.

[76] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge: Cambridge University Press, Apr. 2016. DOI: 10.1017/CBO9781316219232.

[77] Grigorios A. Pavliotis and Andrew M. Stuart. *Multiscale Methods: Averaging and Homogenization*. Texts in Applied Mathematics. Springer New York, 2008, pp. XVIII, 310. ISBN: 978-0-387-73829-1. DOI: 10.1007/978-0-387-73829-1.

[78] Corey M. Peak, Rebecca Kahn, Yonatan H. Grad, Lauren M. Childs, Ruoran Li, Marc Lipsitch, and Caroline O. Buckee. "Individual quarantine versus active monitoring of contacts for the mitigation of COVID-19: a modelling study." In: *The Lancet Infectious Diseases* 20.9 (2020), pp. 1025–1033. ISSN: 1473-3099. DOI: 10.1016/S1473-3099(20)30361-3.

[79] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[80] Sen Pei, Sasikiran Kandula, Wan Yang, and Jeffrey Shaman. "Forecasting the spatial transmission of influenza in the United States." In: *Proceedings of the National Academy of Sciences* 115.11 (2018), pp. 2752–2757. DOI: 10.1073/pnas.1708856115.

[81] Sen Pei, Sasikiran Kandulaa, and Jeffrey Shaman. "Differential Effects of Intervention Timing on COVID-19 Spread in the United States." In: *Science Advances* 6.49 (2020), eabd6370. DOI: 10.1126/sciadv.abd6370.

[82] Tiago P. Peixoto. "The graph-tool Python library." May 2017. DOI: 10.6084/m9.figshare.1164194.v14.

[83] Giorgio Quer, Jennifer M. Radin, Matteo Gadaleta, Katie Baca-Motes, Lauren Ariniello, Edward Ramos, Vik Kheterpal, Eric J. Topol, and Steven R. Steinhubl. "Wearable sensor data and self-reported symptoms for COVID-19 detection." In: *Nature Medicine* 27.1 (Jan. 2021), pp. 73–77. ISSN: 1546-170X. DOI: 10.1038/s41591-020-1123-x.

[84] Jennifer M. Radin, Nathan E. Wineinger, Eric J. Topol, and Steven R. Steinhubl. "Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study." In: *The Lancet Digital Health* 2.2 (Feb. 2020), e85–e93. ISSN: 2589-7500. DOI: 10.1016/S2589-7500(19)30222-5.

[85] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001.

[86] Safiya Richardson, Jamie S. Hirsch, Mangala Narasimhan, James M. Crawford, Thomas McGinn, Karina W. Davidson, and the Northwell COVID-19 Research Consortium. "Presenting Characteristics, Comorbidities, and Outcomes Among 5700 Patients Hospitalized With COVID-19 in the New York City Area." In: *JAMA* 323.20 (May 2020), pp. 2052–2059. ISSN: 0098-7484. DOI: 10.1001/jama.2020.6775.

[87] Aaron Richterman, Eric A. Meyerowitz, and Muge Cevik. "Hospital-Acquired SARS-CoV-2 Infection: Lessons for Public Health." In: *JAMA* 324.21 (Dec. 2020), pp. 2155–2156. ISSN: 0098-7484. DOI: 10.1001/jama.2020.21399.

[88] Eli S. Rosenberg et al. "Cumulative incidence and diagnosis of SARS-CoV-2 infection in New York." In: *Annals of Epidemiology* 48 (2020), pp. 23–29. ISSN: 1047-2797. DOI: 10.1016/j.annepidem.2020.06.004.

[89] Marcel Salathé, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W. Feldman, and James H. Jones. "A high-resolution human contact network for infectious disease transmission." In: *Proceedings of the National Academy of Sciences* 107.51 (Dec. 2010), pp. 22020–22025. DOI: 10.1073/pnas.1009094108.

[90] Henrik Salje et al. "Estimating the burden of SARS-CoV-2 in France." In: *Science* 369.6500 (July 2020), pp. 208–211. DOI: 10.1126/science.abc3517.

[91] Daniel Sanz-Alonso, Andrew Stuart, and Armeen Taeb. *Inverse Problems and Data Assimilation*. London Mathematical Society Student Texts. Cambridge: Cambridge University Press, 2023. DOI: 10.1017/9781009414319.

[92] Tapio Schneider and Stephen M. Griffies. "A Conceptual Framework for Predictability Studies." In: *Journal of Climate* 12.10 (1999), pp. 3133–3155. DOI: 10.1175/1520-0442(1999)012<3133:ACFFPS>2.0.CO;2.

[93] Tapio Schneider et al. "Epidemic management and control through risk-dependent individual contact interventions." In: *PLOS Computational Biology* 18.6 (June 2022), pp. 1–32. DOI: 10.1371/journal.pcbi.1010171.

D. B. wrote extensive amounts of project's code, such as integration of ordinary differential equations, various data assimilation (DA) algorithms and graph operations; conducted numerical experiments, participated in developement of specific to the problem DA methods, investigation of the results and their interpretation.

[94] Nandini Sethuraman, Sundararaj Stanleyraj Jeremiah, and Akihide Ryo. "Interpreting Diagnostic Tests for SARS-CoV-2." In: *JAMA* 323.22 (June 2020), pp. 2249–2251. ISSN: 0098-7484. DOI: 10.1001/jama.2020.8259.

[95] Jeffrey Shaman and Alicia Karspeck. "Forecasting seasonal outbreaks of influenza." In: *Proceedings of the National Academy of Sciences* 109.50 (2012), pp. 20425–20430. DOI: 10.1073/pnas.1208772109.

[96] Kieran J. Sharkey. "Deterministic epidemiological models at the individual level." In: *Journal of Mathematical Biology* 57.3 (Sept. 2008), pp. 311–331. ISSN: 1432-1416. DOI: 10.1007/s00285-008-0161-7.

[97] Noam Shental et al. "Efficient high-throughput SARS-CoV-2 testing to detect asymptomatic carriers." In: *Science Advances* 6.37 (2020), eabc5961. DOI: 10.1126/sciadv.abc5961.

[98] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-52452-8. DOI: 10.1007/978-3-319-52452-8.

[99] Bojan Simoski, Michel C. A. Klein, Eric Fernandes de Mello Araújo, Aart T. van Halteren, Thabo J. van Woudenberg, Kirsten E. Bevelander, Moniek Buijzen, and Henri Bal. "Understanding the complexities of Bluetooth for representing real-life social networks." In: *Personal and Ubiquitous Computing* (Aug. 2020), pp. 1–20. ISSN: 1617-4917. DOI: 10.1007/s00779-020-01435-x.

[100] Joanna Slawinska and Dimitrios Giannakis. "Indo-Pacific Variability on Seasonal to Multidecadal Time Scales. Part I: Intrinsic SST Modes in Models and Observations." In: *Journal of Climate* 30.14 (2017), pp. 5265–5294. DOI: 10.1175/JCLI-D-16-0176.1.

[101] Statista. *Penetration rate of smartphones in selected countries 2021*. 2022. URL: https://statista.com/statistics/539395/smartphone-penetration-worldwide-by-country/.

[102] Ingo Steinwart and Andreas Christmann. "Kernels and Reproducing Kernel Hilbert Spaces." In: *Support Vector Machines*. Information Science and Statistics. New York, NY: Springer New York, 2008, pp. 110–163. DOI: 10.1007/978-0-387-77242-4_4.

[103] Andrew Stuart and Anthony R Humphries. *Dynamical systems and numerical analysis*. Vol. 2. Cambridge University Press, 1998.

[104] Floris Takens. "Detecting strange attractors in turbulence." In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381. ISBN: 978-3-540-38945-3.

[105] THE CITY. *Coronavirus in New York City*. 2022. URL: https://projects.thecity.nyc/2020_03_covid-19-tracker/.

[106] Warwick Tucker. "The Lorenz attractor exists." In: *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 328.12 (1999), pp. 1197–1202.

[107] U.S. Census. *Age demographics, New York, NY*. 2018. URL: https://datausa.io/profile/geo/new-york-ny/#demographics.

[108] U.S. Food and Drug Administration. *EUA Authorized Serology Test Performance*. 2020. URL: https://www.fda.gov/medical-devices/emergency-situations-medical-devices/eua-authorized-serology-test-performance.

[109] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[110] Divakar Viswanath. "The fractal property of the Lorenz attractor." In: *Physica D: Nonlinear Phenomena* 190.1 (2004), pp. 115–128. DOI: 10.1016/j.physd.2003.10.006.

[111] Mark B. van der Waal, Carolina dos S. Ribeiro, Moses Ma, George B. Haringhuizen, Eric Claassen, and Linda H. M. van de Burgwal. "Blockchain-facilitated sharing to advance outbreak R&D." In: *Science* 368.6492 (May 2020), pp. 719–721. DOI: 10.1126/science.aba1355.

[112] Wenling Wang, Yanli Xu, Ruqin Gao, Roujian Lu, Kai Han, Guizhen Wu, and Wenjie Tan. "Detection of SARS-CoV-2 in different types of clinical specimens." In: *JAMA* 323.18 (May 2020), pp. 1843–1844. ISSN: 0098-7484. DOI: 10.1001/jama.2020.3786.

[113] Xinyang Wang, Joanna Slawinska, and Dimitrios Giannakis. "Extended-range statistical ENSO prediction through operator-theoretic techniques for nonlinear dynamics." In: *Scientific Reports* 10.1 (Feb. 2020), p. 2636. DOI: 10.1038/s41598-020-59128-7.

[114] C. Watson, A. Cicero, J. Blumenstock, and M. Fraser. *A National Plan to Enable Comprehensive COVID-19 Case Finding and Contact Tracing in the US*. Tech. rep. Johns Hopkins Bloomberg School of Public Health, Center for Health Security, Apr. 2020.

[115] Daniel S. Wilks. "Effects of stochastic parametrizations in the Lorenz '96 system." In: *Quarterly Journal of the Royal Meteorological Society* 131.606 (2005), pp. 389–407. DOI: 10.1256/qj.04.03.

[116] Zunyou Wu and Jennifer M. McGoogan. "Characteristics of and Important Lessons From the Coronavirus Disease 2019 (COVID-19) Outbreak in China: Summary of a Report of 72314 Cases From the Chinese Center for Disease Control and Prevention." In: *JAMA* 323.13 (Apr. 2020), pp. 1239–1242. DOI: 10.1001/jama.2020.2648.

[117] Chris Wymant et al. "The epidemiological impact of the NHS COVID-19 App." In: *Nature* 594.7863 (June 2021), pp. 408–412. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03606-z.

[118] Wan Yang et al. "Estimating the infection-fatality risk of SARS-CoV-2 in New York City during the spring 2020 pandemic wave: a model-based analysis." In: *The Lancet Infectious Diseases* 21.2 (Feb. 2021), pp. 203–212. ISSN: 1473-3099. DOI: 10.1016/S1473-3099(20)30769-6.

[119] Lihi Zelnik–Manor and Pietro Perona. "Self-Tuning Spectral Clustering." In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004, pp. 1601–1608.

[120] Zhizhen Zhao and Dimitrios Giannakis. "Analog forecasting with dynamics-adapted kernels." In: *Nonlinearity* 29.9 (Aug. 2016), pp. 2888–2939. DOI: 10.1088/0951-7715/29/9/2888.

[121] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization." In: *ACM Trans. Math. Softw.* 23.4 (Dec. 1997), pp. 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236.