A conditional Gaussian process model for molecular property prediction and chemical discovery

Gosnell, Arron

Award date:
2022

Awarding institution:
University of Bath

A CONDITIONAL GAUSSIAN PROCESS MODEL FOR MOLECULAR PROPERTY PREDICTION AND CHEMICAL DISCOVERY

ARRON JAMES GOSNELL

A thesis submitted for the degree of:
Doctor of Philosophy

University of Bath
Department of Mathematical Sciences
May 2022

Arron James Gosnell: *A conditional Gaussian process model for molecular property prediction and chemical discovery* , Doctor of Philosophy, © May 2022

DECLARATION

The material presented here for examination for the award of a higher degree by research has not been incorporated into a submission for another degree.

I am the author of this thesis, and the work described therein was carried out by myself personally.

Arron James Gosnell

ABSTRACT

With the proliferation of screening tools for chemical testing, it is now possible to create vast databases of chemicals easily. However, rigorous statistical methodology used to analyse these databases are in their infancy, and further development to facilitate chemical discovery is imperative. In this thesis, conditional Gaussian process models are developped within a regression and classification setting to predict herbicidal efficacy from glasshouse experiments . The Tanimoto metric is employed within the covariance of the Gaussian processes to account for distances and capture correlated effects within the chemical space. Using molecular fingerprints, a representation of a compound within the chemical space, it is shown that by accounting for correlation amongst herbicidal compounds, predictive performance can be improved over the uncorrelated model, where the effects between compounds are assumed to be independent. Moreover, several optimisation techniques on discrete spaces are presented for the facilitation of chemical discovery. These methods assist in searching interesting regions of the chemical space and support the identification of key molecular features attributing to high efficacy. Furthermore, a simulation study is conducted to confirm the suitability of the both the Tanimoto metric and the method of scoring rules to evaluate model performance on the novel application. We conclude that the spatially correlated model has the ability to improve predictions, and also has the potential to be applied to other drug discovery settings and beyond.

ACKNOWLEDGMENTS

A personal thank you to my first supervisor, Dr Evangelos Evangelou, for imparting so much knowledge on the topic of statistics and for always being available to support me when I had any questions. I would like to thank my other supervisors, Dr Jonathan Bartlett from the University of Bath and Dr Konstantinos Papachristos and Dr Agisilaos Chantzis from Syngenta for their advice, support, and contributions to the project.

I would like to personally thank Andrew Chapman, who offered excellent advice and suggestions on my project and for his words of encouragement and support. I would like to express my gratitude to SAMBa for providing a positive working environment to feel a part of. I would like to thank my colleagues within my office who brought about good conversations and laughter. I would like to send special thanks UKRI and Syngenta for enabling me to pursue this research through their funding.

I would like to thank my partner for her enduring support throughout my PhD and for keeping me motivated at arduous times. And thank you to my friends and my family for their ongoing support and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI artificial intelligence

AIC Akaike information criterion

ANOVA analysis of variance

AUC area under the curve

BIC Bayesian information criterion

BFGS Broyden-Fletcher-Goldfarb-Shanno

CDF cumulative density function

EPS early profile screening

GA genetic algorithm
GAM generalised additive model
GLM generalised linear model
GP Gaussian process
MAE mean absolute error
ML machine learning
MS mean square
MSE mean squared error
NN neural network
PDF probability density function
PMF probability mass function
PPS primary profile screening
QSAR quantitative structure-activity relationships
RBF radial basis function
RF random forest
ROC receiver operating characteristic curve
SA simulated annealing
SVM support vector machine

1 INTRODUCTION AND MOTIVATION

1.1 INTRODUCTION TO THE PROBLEM IN A GENERAL CONTEXT

1.1.1 *Project aim and hypothesis*

This thesis aims to develop suitable methodology in assessing compounds from their descriptive characteristics and predicting their chemical properties within controlled experiments. By representing molecular structures as chemical fingerprints, we aim to account for the "closeness" of compounds within the chemical space. GPs will be adapted to live on the chemical space, see Section 1.1.4, and will be employed and tested for predicting compound performance in both a regression and classification setting, the latter presenting greater novelty to the thesis. Furthermore, a ranking system for the compounds will be implemented in terms of a given property. In-turn, several optimisation algorithms will be applied, including the genetic algorithm GA and simulated annealing (SA), for the navigation of the chemical space, i. e., the space containing the ensemble of all chemical compounds, and to identify potentially promising compounds in terms of a given property. By implementing these novel approaches, the effect of untested compounds as well as capturing the uncertainty of the proposed effects.

An underlying assumption of the model is that two compounds "close" within the chemical space will demonstrate similar biological properties. Using chemical fingerprints, a representation of molecules as a binary string, we aim to identify compounds similar to those known to be effective molecules. The novelty of this project is that closeness in the chemical space is accounted for within the model, meaning information on observed compounds will inform the assessment of unseen compounds. In-turn, novel compounds of high efficacy may be proposed, with the consideration that proposed compounds may not obey necessary physical laws. In identifying similarity between compounds, we employ the Tanimoto (Jaccard) coefficient, a proper metric on dichotomous spaces.

With the model developed to predict both the effect of the compound and the class damage resulting from experiments, we then identify the features within the compounds which present the greatest contribution towards potency. This will be achieved by applying several optimisation techniques, including SA and GA. These newly proposed compounds can then be assessed to determine their effect.

In practice, the results imply that compounds similar in nature to the proposed compounds me be derived.

The data used to demonstrate our methods are provided by Syngenta, an agro-technology company. At Syngenta, thousands of potential herbicides undergo a sequence of screening tests in glasshouses, where compounds are tested under a variety of experimental conditions for any given project. The dataset used to demonstrate our methods consists of 35,740 unique glasshouse experiments conducted at the initial stages of testing. Within the glasshouse dataset, there are 745 distinct compounds. Each compound is described by a unique chemical fingerprint which comprises 2048 features. The chemical fingerprint may be viewed as a representation of the chemical space, see Section 2.1. The glasshouse experiments are characterised by several factors, including the compound used, rate of application (dosage), accustomed climate of the plant (warm or cold), stage of growth (pre- or post-emergence), and the plant group (dicot or monocot). The outcome of each experiment is the level of damage on the plant, recorded within 14 days of application and measured by comparing the pigmentation of the plant against colour codes. The damages are recorded as percentages in multiples of 10. A recorded damage of 0% indicates no herbicidal effect, whilst a damage of 100% indicates complete necrosis of the plant.

1.1.2 *Contributions*

We show that, indeed, GPs can be defined on spaces other than Euclidean with the Tanimoto metric, and present suitable correlation functions within this pursuit. We define a GP model in both a regression and an ordinal classification context [23], with the focus being on the latter. The GP model developed in this thesis incorporates fixed covariates which capture the additional information on the testing process. The GP model can be described as a cumulative link model with correlated random effects [1, Section 5.1]. As the likelihood is not available in closed-form, the Laplace approximation is employed to evaluate and fit the model. Thus, another contribution of this thesis is the application of the Laplace approximation for estimation and prediction of ordinal data. Due to the correlation structure of the GP model, we can predict the effect of untested compounds and their outcomes from glasshouse experiments. To date, this has not been achieved using a generalised linear mixed model with independent random effects. Confidence intervals for the predicted effects are assumed through the closed form nature of the predictive distribution, a particularly attractive feature of the GP. Another contribution of this project is the implementation of drug discovery methods using novel approaches. By defining the predictions form the GP as the target variable and the features within the fingerprints as independent

predictors, we may identify key features within the chemical fingerprints attributing to high efficacy. This is performed using several optimisation methods, including GA and SA.

1.1.3 *Review of existing methods*

Traditionally, drug discovery is a lengthy and expensive process prone to high failure rates. Machine learning (ML) has become increasingly utilised to aide in drug discovery and development due to its computational speed and ease of scaling. While traditional physical models rely on quantum chemistry or molecular dynamics simulations, ML uses algorithms to recognise patterns and relationships between empirical observations of compounds [87]. The trained ML model can be used to make predictions in various stages of the drug discovery process, such as predicting target structure, biological activities, and interactions.

In drug discovery, a key application of ML is assisting with the understanding of relationships between molecular structures and biological activity [2]. For example, given a promising herbicidal compound from a series of screening trials, we may wish to know how its chemical structure can be optimized to improve several molecular properties, including its biological responses or mode of action. Until recently, investigating these types of relationships could only be performed through costly and labour-intensive analysis [4].

Today, modern ML techniques can be used to model quantitative structure-activity relationships (QSAR) and develop artificial intelligence (AI) programs that accurately predict how chemical modifications influence biological behaviour [20]. Various molecular properties, such as metabolism, toxicity, and interactions, have been effectively modelled by QSAR techniques [20]. Even though early methods were successful at the time, ultimately these approaches were limited by the scarcity of relevant experimental data. Therefore, sophisticated ML techniques capable of capturing non-linear relationships, as well as data of increasing depth and complexity, are required.

The motivation for this project comes from our collaboration with Syngenta, an agrotechnology company that operate worldwide. At Syngenta, thousands of chemical compounds are screened yearly in glasshouse experiments under regulated conditions. The core aim of these trials is to identify compounds, and their associated chemical properties, which have the potential to become effective herbicidal products. Within each experiment, a compound is applied to crops of various physical characteristics. The effect of the compound is assessed by a biologist typically within two weeks of initial application. A score is then attributed to the experiment, indicating the level of damage inflicted on the plant. Successful herbicides will progress through a series of trials with the final compounds being sent to field trials.

1.1.4 *Chemical space*

The chemical space describes the ensemble of all organic molecules [74]. The chemical space is not continuous, i. e., there are only discrete changes that can be made to a compound, e. g., including or excluding a substructure within the molecule. Using the concept of "closeness" between two molecules is a central principle in chemoinformatics for the navigation of the chemical space, and allows for the discovery of compounds with similar biochemical properties [7, 53]. However, the vast number of regions to explore makes virtual screening particularly challenging [8]. Referring to the set of all possible chemicals as a space likens this to a geographical map illustrating the distribution of molecular properties. To visualise such a map, each molecular descriptor can be assigned a dimension, thereby assigning each molecule a location within a multidimensional space. Using the concept of a space containing positional information allows for the navigation of bioactive molecules, thus performing virtual screening to select compounds for in-vitro testing [74].

The concept of the chemical space has wide-spread practical applications. Within the drug discovery realm, the chemical space has provided a solid conceptual framework to support diversity analysis, structure classification, library design, compound selection and assessment of structure-property, and structure-activity relationships [63]. Drug discovery is important to many applications outside the pharmaceutical industry. In particular, drug discovery methods can be applied to herbicide selection, where it is currently impossible to evaluate all combinations of atomic features to identify the ideal herbicide. Instead, intelligent methods are needed as a guide for exploration of interesting regions within the chemical space

Many chemoinformatics methods rely on QSAR analysis [67, 86], which quantitatively correlates the chemical structure with the biological activity [66]. Chemical graph theory is vital in understanding the influence chemical structures have on their biological activity [9]. A chemical graph is a mathematical representation of a chemical structure and containins sufficient information to model and provide insight into a wide range of biological phenomena [58]. The chemical graph contains chemical descriptors, which are numerical features extracted from chemical structures. Representing the chemical structure in this way allows the application of several machine learning techniques, including molecular data mining, compound diversity analysis and compound activity prediction [27]. Chemical descriptors can be represented in several dimensions, ranging from 1D to 4D [58]. The representation of 3D molecular fingerprints have been applied in recent chemoinformatics literature [68, 95]. However, it has been shown that 2D and 3D descriptors have similar performance in certain

applications, with 2D being the most common representation due to their generation being easy, fast, and convenient [36].

1.2 EXISTING APPROACHES

1.2.1 *Existing uses of chemical structures*

Drug discovery is of vital importance to many fields including agricultural sciences, chemistry, medicine, and the food industry [90]. Many of the methods applied in analysing biological activity for drug discovery fall under the umbrella of chemoinformatics. Chemoinformatics is the branch of statistics concerned with the prediction of chemical and biological properties of chemical compounds based on each compound's chemical structure [64]. There are many areas in chemoinformatics related to the discovery of novel drugs, including computer-aided drug synthesis, chemical space exploration, scaffold analysis, and library design, [3, 51].

1.2.1.1 *Existing studies incorporating chemical structures*

Statistical modelling of herbicide performance has long been utilised when developing effective herbicides. Various methods have been employed to analyse herbicidal activity, most commonly through the use of regression analysis [54, 69, 76]. Other statistical methods of modelling herbicidal data have been employed. Colby [24] presents a method for calculating expected responses of herbicide combinations through dose-response curves. Flint, Cornelius, and Barrett [34] use analysis of variance (ANOVA) to check for difference in treatment means of herbicides, whilst Seefeldt, Jensen, and Fuerst [78] fitted the log-logistic function to express dose-responses.

A notable drawback of these methods is that they fail to incorporate the chemical structure of the herbicidal compounds, thereby neglecting important information in the model. We therefore require methods which allow for chemical structures. Machine learning methods address this shortfall and can be used in the application of chemical discovery.

1.2.2 *ML applications*

A vast number of machine learning methods can incorporate molecular fingerprints for drug discovery, including random forests, gradient boosted decision trees, single- and multitask deep neural networks [36].

Recent literature in chemoinformatics includes the application of several machine learning techniques for the discovery of novel compounds, including support vector machine (SVM) [59, 70], K-nearest

neighbours [71], random forest (RF) [15, 96] and deep learning [18, 75]. These methods posses several benefits including their ability to scale and their accuracy in classification. However, these methods fail to account for the correlation between effects of the compounds, thereby assuming the effects of the compounds to be independent of one another.

The effects within our application correspond to the damage inflicted upon a plant by the application of each compound. Incorporating a spatial effect within the model allows correlation to be accounted for. Typically, correlation is incorporated as a function of Euclidean distance, thereby restricting the application to problems of a geographical nature. By incorporating distances on dichotomous spaces within the model, we can broaden the application to other data types, in particular categorical and count data.

1.3 SIMILARITY OF COMPOUNDS

There exist over 70 methods for quantifying closeness in dichotomous features-spaces [21], however many are discounted as proper metrics as they do not satisfy the metric criteria. The Tanimoto coefficient [33] is a measure of similarity between two dichotomous vectors and does satisfy the metric criteria, see Section 3.1. It has been shown that when neglecting a *priori* knowledge on the compounds in testing, the Tanimoto similarity scores highest in terms of modelling results, alongside similarity metrics such as the Dice index and cosine similarity [4, 19].

The value of the Tanimoto coefficient ranges between 0 and 1, with a value of 1 indicating that the two compounds are considered identical and a value of 0 indicating no common features exist between them. Compounds are not necessarily identical when the Tanimoto coefficient is 1, as one only considers a subset of the entire feature space. However, the number of features is rich enough in practice that compounds are distinguishable. Subtracting the Tanimoto coefficient from 1 converts the expression into a distance, known as the Sorgel or Jaccard distance [57], and is sometimes referred to as the Tanimoto dissimilarity. This means two compounds are treated as identical if their Tanimoto distance is 0. This is an important procedure when calculating the GP covariance structure for modelling similarity of compounds, since covariances are functions of distances. An important result is that the Tanimoto distance does satisfy the metric criteria, in particular the triangular inequality [57], thereby making it a suitable metric. Within the chemoinformatics literature, it is the most widely used method for quantifying similarity of molecular substructures [4].

Since the matrix of Tanimoto similarities is itself positive definite, it has been applied in several machine learning applications, including artificial neural networks [47], and SVMs [87]. Further, Tanimoto similarities have been utilised in GPs as a correlation matrix for the effects

of compounds [65]. One criticism these papers is the absence of the scale parameter which controls for correlation, thereby treating correlations as fixed. We overcome this drawback by using the Tanimoto distance as a distance measure within well known GP correlations within the spatial statistic literature, such as the exponential kernel. This widens the application of the GP by living on non-Euclidean spaces, resulting in a vast number of data-driven problems being open to the application of GP.

1.3.1 *Tanimoto literature*

The Tanimoto coefficient is widely applied within the ML literature. However, it has seldom been applied within spatial GP models. Moss and Griffiths [65] proposed a GP framework for molecular property prediction within a regression setting using Tanimoto coefficient. Bajusz, Rácz, and Héberger [4] quantifies the similarity of molecules using the Tanimoto distance, as well as other similarity metrics. Swamidass et al. [87] and Gärtner, Le, and Smola [37] used a Tanimoto based kernel in the application of support vector machines in assessing molecular similarity. Czarnecki [26] applied weighted Tanimoto kernels as hidden layer nodes within a neural network for drug classification. Our application uses the Tanimoto similarity within a GP, thereby capturing the distance between compounds as a means of informing the effects of unobserved compounds.

1.3.2 *Gaussian process models*

We present a novel approach to incorporating chemical distance into a statistical model with the use of GP. GP are commonly defined on Euclidean spaces and act on the notion of distance. Two items close in Euclidean space should have a similar effect on the outcome we are modelling. For instance, predicting earthquakes, oil locations and many other spatially correlated phenomena. In this project, the metric we use for chemical structures is non-Euclidean, and therefore the GP may be defined on the chemical space. The values of the GP represent the effect of each compound on the outcome we wish to model. In our application, we are modelling the level of damage resulting from a glasshouse experiment. Distance in the chemical space is measured by the Tanimoto or Jaccard metric [55, Ch. 5].

1.4 THESIS LAYOUT

This thesis is organised as follows. Chapter 2 discusses the type of data subject to analysis and describes the herbicidal selection process. We discuss measuring similarity of compounds using the Tanimoto coefficient. We then conduct preliminary analysis to provide insight

to the data. Chapter 2 also discusses the statistical methods to be applied to the data, including GPs (Section 2.2) and RF with its variants (Section 2.5.3). We discuss how these methods will be used to predict the characteristics of herbicidal compounds using the Tanimoto coefficient as a measure of correlation. Section 2.2.1 examines the covariance functions applied to modelling herbicidal data with GPs and how we inform our choice of covariance by the nature of the data. We discuss obtaining parameter estimates of the covariance function through maximising the likelihood. Section 2.2.6 presents the bottlenecks in modelling large amounts of data with GPs and how we may overcome certain impracticalities. Chapter 3 provides details on the Tanimoto metric and its properties within a GP framework. A variogram is discussed and implemented to demonstrate the spatial dependence of the Tanimoto distance. The conditional GP model for ordinal outcomes is detailed in addition parameter estimation methods. Further, optimisation methods, including GA and SA are discussed. Furthermore, a simulation study is conducted to demonstrate the suitability of proper scoring rules when evaluating the GP predictive performance. Chapter 4 details the steps in modelling Syngenta's glass house data using GPs along with graphical illustrations of our model's accuracy. Chapter 5 gives a summary of the project findings and potential directions for further research. All calculations in this project were performed using R [88].

2

BACKGROUND

2.1 REPRESENTATION OF THE CHEMICAL SPACE

Molecular fingerprints are a widely used concept for assessing molecular substructures within the chemical space, and are frequently used in the the application of drug discovery [87]. They comprise a sequence of dichotomous features indicating the presence of some atomic substructure, e. g., a vertex or a cycle [52]. Figure 2.1 illustrates two molecules and a sample of their associated fingerprints. Representing the graphical structure of molecules as a vector of features allows one to assess "closeness" within the chemical space.

Each compound within our dataset, introduced in Section 4.1, is uniquely identified by a chemical fingerprint. These chemical fingerprints are 2048 features in length, where each feature takes a dichotomous value and indicates the presence or absence of some atomic feature, see Figure 2.1. These features can represent, for example, a ring or a nitrogen molecule or a ring and, in general, form part of the compounds molecular graph. Fingerprint features may also take integer values representing counts of the features occurrence. Counts can, however, be converted to dichotomous features by repeating the feature within the fingerprint by the number of counts. Within the data received by Syngenta, only 95 of these features were present in some compounds.

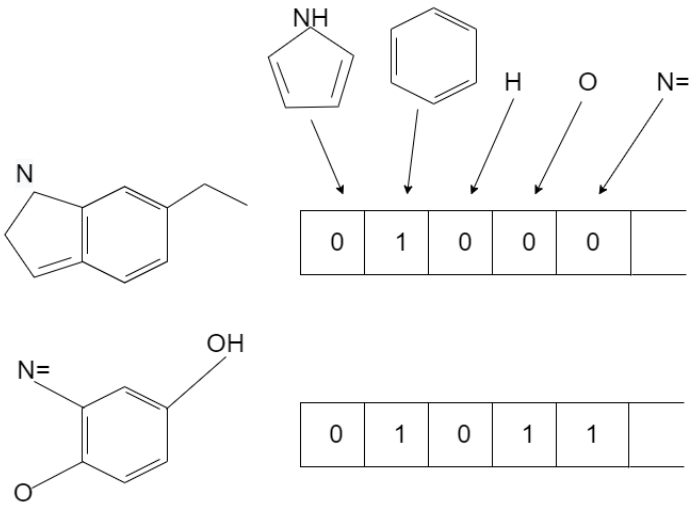


Figure 2.1: Two molecules and their corresponding fingerprints based on their substructures. The second feature within the two fingerprints have the value of 1, indicating the common presence of a cycle.

2.2 DEFINITION AND THEORY GP

GPs are a popular method of modelling data due to their flexibility, simplicity, and substantial theoretical support [22]. With GPs, rather than prescribe a parametric formula for the function $f(x)$, as with linear regression, we let the data 'speak for itself'. A GP is a random function on an $n \times d$ dimensional space $X^{n\times d}, f : X^{n\times d} \to \mathbb{R}$ with Gaussian finite dimensional distributions. Here, n represents the number of data points and d represents the number of variables in the design matrix. Existence is guaranteed by Kolmogorov's consistency theorem [83, Ch. 2]. Formally, a GP is a distribution over functions such that any finite set of function values follows a joint multivariate Gaussian distribution [72]. Essentially, a GP is an extension of a multivariate normal distribution to infinite dimensions. The two main characterisations of a GP are its mean function, μ, and variance-covariance function, k,

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}_1) &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}_1) - \mu(\mathbf{x}_1))] \\ &= \mathrm{Cov}(f(\mathbf{x}), f(\mathbf{x}_1)), \end{aligned}$$

where $\mathbf{x}$ and $\mathbf{x}_1$ are two input vectors. Covariance functions can take several forms depending on the nature of the data, see Section 2.2.1. We denote the class of a GP, $f(\cdot)$ with mean function $\mu(\cdot)$ and covariance $k(\cdot,\cdot)$ by

$$f(\cdot) \sim \mathrm{GP}(\mu(\cdot), k(\cdot,\cdot)). \tag{2.1}$$

By convention, it is common to assume a zero-mean prior i. e., $\mathbb{E}[f(x)] = 0$, resulting in the structure of a GP being entirely determined by its covariance. In our application, we incorporate covariates into the model. We later incorporate the mean within the GP through a series of basis functions. The value of the kernel at pairwise inputs $x_1, \ldots, x_n$ gives a variance-covariance matrix. The general form of the variance-covariance matrix of a GP is

$$K(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \ldots & k(x_n, x_n) \end{bmatrix}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$. New observations also form a kernel with existing data points, $K_* = \mathrm{Cov}(f(\mathbf{x}), f(x_*)) = [k(x_*, x_1) \ldots k(x_*, x_n)]$, with the covariance of the new data point being denoted by $K_{**} = \mathrm{Cov}(f(x_*), f(x_*)) = k(x_*, x_*)$. GPs are effective at interpolation since there is no dependence on 'far away points'. However, extrapolation of new data increases uncertainty in our predictions.

2.2.1 *Kernel*

The covariance function is an important aspect of predictive modelling as it encodes assumptions about how the underlying process appears [79]. New inputs with similar performance to those already seen will be highly correlated, which is an important feature in the prediction process. If, for example, $x \approx x_*$ then the kernel $k(x, x_*)$ approaches its maximum value, resulting in $f(x) \approx f(x_*)$. This feature essentially defines the smoothness of the function in question. If, on the other hand, we assess a new input x_* which is unlike any existing data, i. e., a low correlation being present, we expect the value of the covariance to be near zero, i. e., $k(x, x_*) \approx 0$.

There are a whole family of kernels to choose from, with our choice being motivated by the nature of the data. Valid kernels must be both symmetric and positive definite, meaning all eigenvalues of the GP variance-covariance matrix are positive. More formally, a kernel is positive semi-definite if the following holds

$$\int\int k(x, x_*)s(x)s(x_*)d\nu(x)d\nu(x_*) \geqslant 0, \tag{2.2}$$

for all squared integrable functions $s \in X$ with respect to the measure. The integral in Equation (2.2) corresponds to the Stieltjes integral and ν is a measure on the Borel sigma-algebra generated by X.

Kernels can be either stationary or non-stationary. Stationary kernels are a function of $x - x'$, which are invariant to translations in the input space. If the kernel is a function of $|x - x'|$ alone, where $|x - x'|$ is the Euclidean distance between x and x', then we say it is isotropic. Another important aspect of a stationary covariance function is continuity in mean square (MS). In defining MS continuity, we let $x_1, x_2, \ldots$ be a sequence of points and $x_* \in X$, such that $|x_k - x_*| \to 0$ as $k \to \infty$. Suppose f is a random field on X, then $f(x)$ is MS continuous at x if $\mathbb{E}[|f(x_k) - f(x_*)|^2] \to 0$ as $k \to \infty$. It turns out that an isotropic GP is MS continuous if and only if $k(h)$ is continuous at 0. To prove this, let

$$\begin{aligned}\mathbb{E}[(f(x_k) - f(x_*))^2] &= \mathbb{E}[(f(x_k)^2 + f(x_*)^2 - 2f(x_k)f(x_*)] \\ &= 2k(0) - 2k(x_k - x_*).\end{aligned}$$

The right-hand side goes to 0 if and only if $\lim_{k\to\infty} k(x_k - x_*) = k(0)$. For a GP, $f(x)$, define

$$f_k(x_*) = \frac{f(x_k) - f(x_*)}{|x_k - x_*|}. \tag{2.3}$$

Then $f(x)$ is MS differentiable if $\lim_{k\to\infty} f_k(x_*)$ exists and is finite. An isotropic GP is MS differentiable if and only if $k''(0) < \infty$. Therefore the choice of kernel characterises continuity and differentiability of a GP.

The following subsections give examples of commonly used kernels in the literature along with illustrations of their behaviour when varying the kernel's hyperparameters. We follow the basis that $X \in \mathbb{R}$ and that $|x - x'|$ corresponds to the Euclidean distance.

2.2.1.1 *Exponential*

One of the simplest kernels is the exponential kernel, parametrised by the length scale, $\phi > 0$, and a variance term σ^2. The exponential kernel takes the form

$$k(x, x_*) = \sigma^2 \exp\left[-\frac{|x - x_*|}{\phi}\right].$$

The length-scale parameter, ϕ, reflects how 'wiggly' the function fitted to the data is. Informally, we may view ϕ as the distance needed to move in the input space before a function value will change significantly [72]. The second parameter, σ^2, determines the functions average squared distance from the mean and gives an idea of the amount of variability in the population.

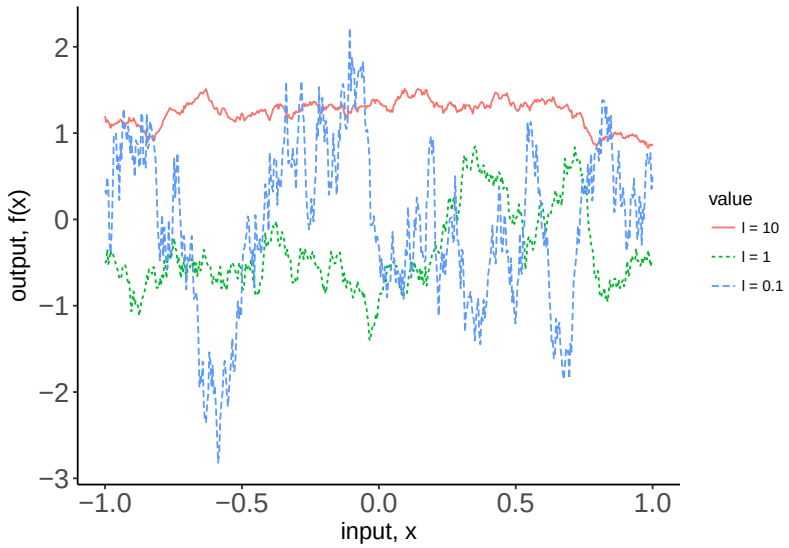


Figure 2.2: Exponential covariance with length-scale parameters $\phi = 1$, 0.1, and 0.01 respectively

2.2.1.2 *Radial basis function*

One of the most frequently used kernels in SVM is the RBF, also known as the squared exponential kernel, denoted

$$k(x, x_*) = \sigma^2 \exp\left[\frac{-|x - x_*|^2}{\phi^2}\right].$$

The RBF is stationary invariant and is parametrised by ϕ and σ^2. The RBF is infinitely differentiable, meaning a GP with an RBF kernel will have mean square derivatives of all orders.

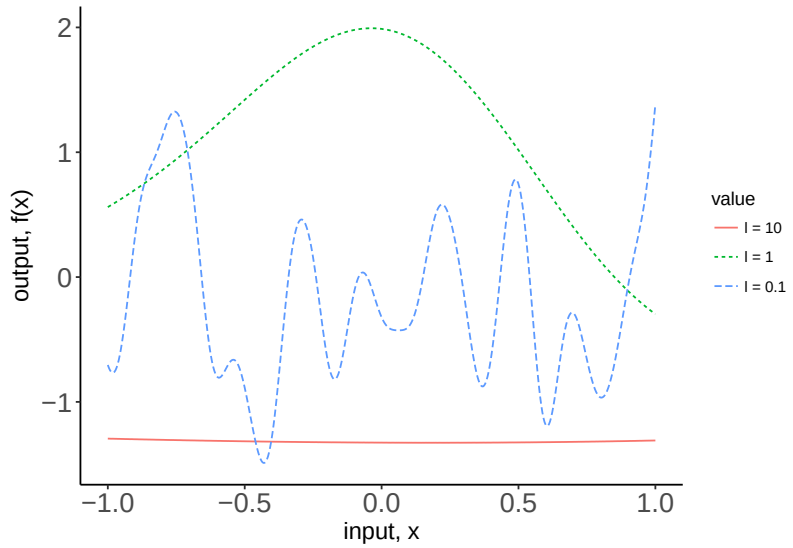


Figure 2.3: RBF with length-scale parameters $\phi = 1$, 0.1, and 0.01 respectively.

2.2.1.3 *Rational quadratic*

The rational quadratic kernel is an infinite sum of RBF kernels with varying length-scales. The two parameters are the length-scale, ϕ, and the scale mixture parameter, $\alpha > 0$, which determines the relative weighting of large-scale and small-scale variations. The rational quadratic function is denoted

$$k(x, x_*) = \left(1 + \frac{|x - x_*|^2}{2\alpha\phi^2}\right)^{-\alpha}.$$

When $\alpha \to \infty$, the rational quadratic is identical to the RBF.

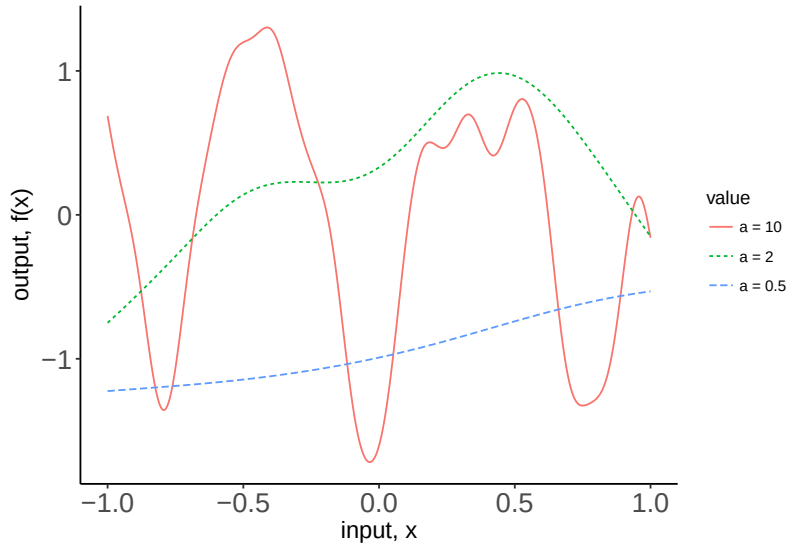


Figure 2.4: Rational quadratic function with a fixed length-scale parameter, $\phi = 1$, and varying scale mixture parameter, with values $\alpha = 10$, 2, and 0.5 respectively.

2.2.1.4 *Exponential-sine squared*

The exponential-sine squared kernel is adopted when capturing periodic trends, such as effects due to seasonality. Parametrised by both a length scale and a periodicity parameter, p where $p > 0$. The kernel is defined as

$$k(x, x_*) = \exp\left[-2\sin^2\left(\frac{\pi}{p}\frac{|x - x_*|^2}{\phi^2}\right)\right].$$

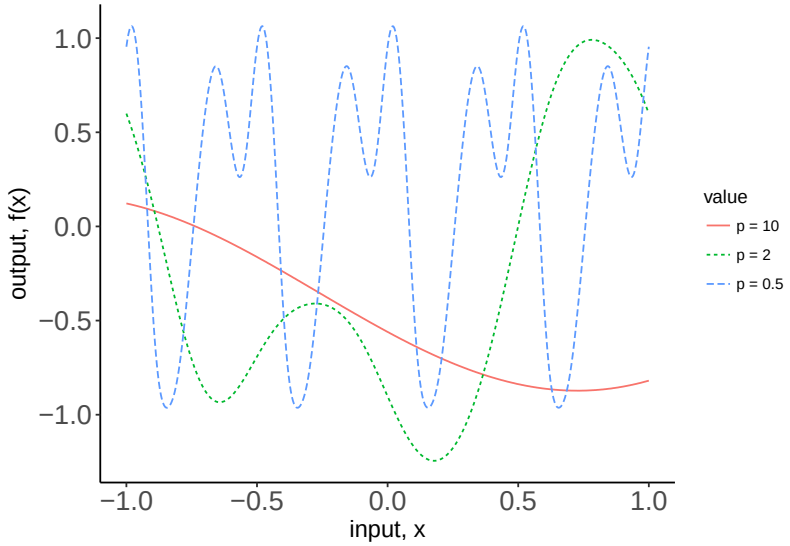


Figure 2.5: Exponential-sine-squared prior with fixed length-scale parameter $\phi = 1$, whilst varying the periodicity parameter $p = 10$, 2, and 0.5

2.2.1.5 *Dot-product*

The dot-product kernel is non-stationary, meaning it is invariant to rotations about the origin, but not translations and is denoted

$$k(x, x_*) = \sigma_0^2 + x \cdot x_*,$$

where $x \cdot x_*$ corresponds to the dot product. This can be obtained from linear regression by assigning $N(0, 1)$ priors on the coefficients of $x_d, d = 1, \ldots, D$ and a prior of $N(0, \sigma_0^2)$ on the error term. There is a single parameter called the bias, σ_0^2, and when this is zero, the kernel becomes the homogeneous linear kernel.

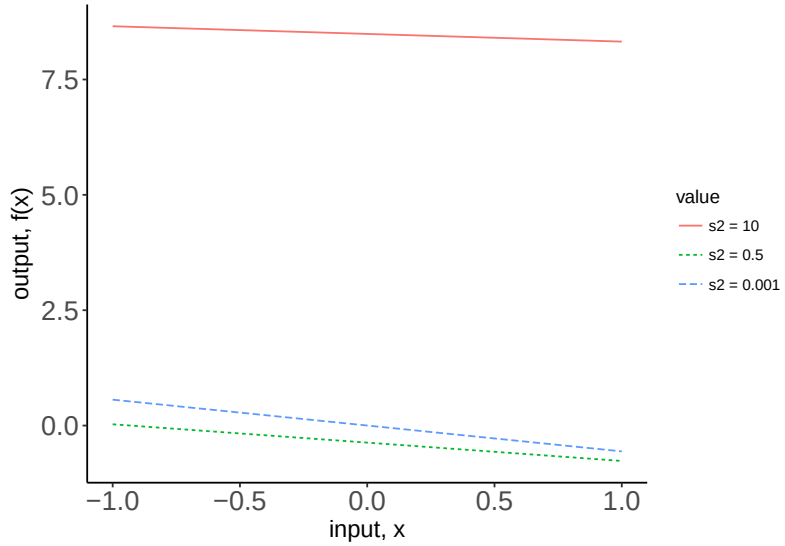


Figure 2.6: Dot product prior with a fixed length-scale parameter $\phi = 1$, whilst varying the variance $\sigma^2 = 10$, 0.5. and 0.001

2.2.1.6 *Matérn*

Named after the work of swedish mathematician Bertil Matérn [61], the Matérn is a common choice of kernel when modelling geo-spatial phenomena. The Matérn kernel is expressed as

$$k(x, x_*) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|x - x_*|}{\phi} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{|x - x_*|}{\phi} \right),$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ϕ and σ^2 are the scale and variance parameters. The Matérn kernel parameter, ν, acts as a smoothing parameter. Given $\nu = 0.5$, the Matérn kernel becomes the exponential kernel, which is very rough, and if $\nu \to \infty$, the Matérn kernel converges to the RBF.

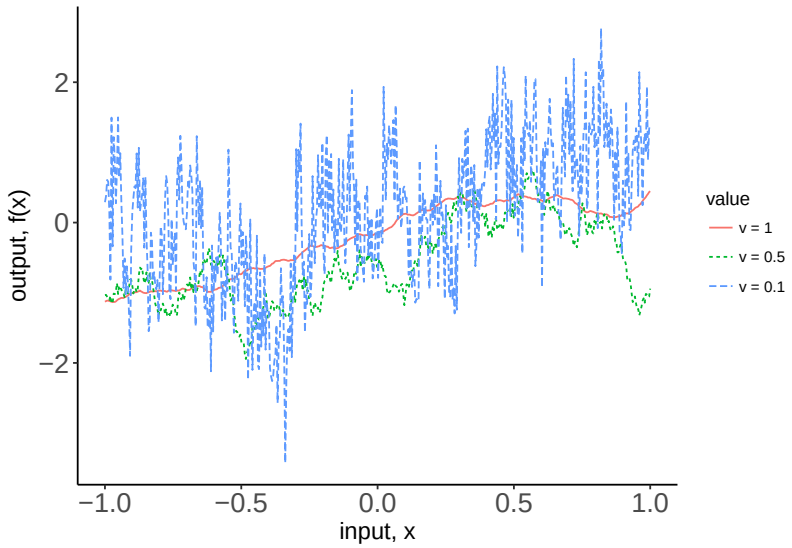


Figure 2.7: Matérn prior with a fixed length-scale parameter of $\phi = 1$, whilst varying the smoothing parameter $\nu = 1$, 0.5 and 0.01.

2.2.2 *GP regression*

Suppose we have some data (x_i, y_i) for $i = 1, \ldots, n$, and we wish to create a model that predicts y from x. With a parametric approach to regression, we assume that the relationship between the mean of y_i and the regressor variables can be represented by some parametric form, such as a linear function. GP regression, on the other hand, is a non-parametric approach, meaning we seek a distribution over all possible functions that are consistent with the data [28].

In describing the GP, we model under the assumption

$$f(\mathbf{x}) \sim \mathrm{GP}(\mu(\cdot), k(\cdot,\cdot)).$$

For any given random vector $\mathbf{x} \in \mathbb{R}^n$, f is Gaussian if it has the density

$$P(f(\mathbf{x})) = \frac{1}{\sqrt{(2\pi)^n |K|}} \exp\left(-\frac{1}{2}(y - f(\mathbf{x}))^\top K^{-1}(y - f(\mathbf{x}))\right),$$

where $|K| = \det(K)$ and $y \in \mathbb{R}^n$ are the observed values of the response variable. Under the proportionality sign, the constant term may be dropped, leaving

$$f(\mathbf{x}) \propto |K|^{-1/2} \exp\left(-\frac{1}{2}(y - f(\mathbf{x}))^\top K^{-1}(y - f(\mathbf{x}))\right),$$

where K is retained as it is a function of the parameters. Considering the case where the observations are noise free, that is assume we are provided with a sample set, $y_i = f(x_i)$,

$$S = \{x_i, f(x_i) | i = 1, \ldots, n\},$$

and wish to predict at new inputs x_*. Then according to the GP prior, the joint distribution of the training outputs given by $\mathbf{f} = f(\mathbf{x}) = (f(x_i), i = 1, \ldots, n)$, and the test outputs $\mathbf{f}_* = f(\mathbf{x}_*)$ is defined as

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\vec{0}, \begin{bmatrix} K & K_*^\top \\ K_* & K_{**} \end{bmatrix}\right),$$

where $K = K(\mathbf{x}, \mathbf{x}) \in \mathbb{R}^{n \times n}$, $K_* = K(\mathbf{x}_*, \mathbf{x}) \in \mathbb{R}^{n_* \times n}$, and $K_{**} = K(\mathbf{x}_*, \mathbf{x}_*) \in \mathbb{R}^{n_* \times n_*}$. To derive the posterior distribution over functions, we restrict this joint prior distribution to contain only those functions which are commensurate with the observed data points, i. e., we need to take the conditional expectation. The multivariate Gaussian distribution has the property that any conditional distribution is also Gaussian. Therefore, the conditional distribution of $\mathbf{f}$ can be fully described with a mean and covariance matrix. From Gaussian conditioning rules, this posterior predictive distribution for noise-free observations is

$$\mathbf{f}_* | f \sim N(K_* K^{-1} \mathbf{f}, K_{**} - K_* K^{-1} K_*^\top). \tag{2.4}$$

To obtain the expected values of our predictions, we simply compute the quantity $K_* K^{-1} \mathbf{f}$. To obtain the uncertainty of each prediction, we compute the quantity $K_{**} - K_* K^{-1} K_*^\top$ and take the square root of the diagonal elements to get the standard errors.

In the case of noisy data, the posterior predictive distribution becomes

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

where $\varepsilon(x) \sim N(0, \tau^2 I)$. The joint distribution becomes

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\vec{0}, \begin{bmatrix} K + \tau^2 I & K_\star^\top \\ K_\star & K_{\star\star} \end{bmatrix}\right),$$

where τ^2 is the noise variance. If we wish to predict the effect in the case of noisy observations, the posterior predictive distribution becomes

$$\mathbf{f}_* | \mathbf{y} \sim N\left(K_* [K + \tau^2 I]^{-1} \mathbf{y}, K_{**} - K_* [K + \tau^2 I]^{-1} K_*^\top\right). \tag{2.5}$$

2.2.3 *Incorporating regressor variables into a GP regression model*

When developing GP models, one may wish to account for fixed effects, in addition to random effects, to improve the model's fit to the data and increase its predictive accuracy. Incorporating fixed effects in addition to random effects simply adjusts the mean function, thereby

providing greater model flexibility. One such model which achieves this is

$$\mathbf{y} = \mathbf{X}^\top \boldsymbol{\beta} + f(\mathbf{x}) + \varepsilon(x), \tag{2.6}$$

$$f(\mathbf{x}) \sim N(0, K),$$

$$\varepsilon(\mathbf{x}) \sim N(0, \tau^2 I),$$

where $\mathbf{y} \in \mathbb{R}^n$ denotes the values of the target variable and $\mathbf{X}$ denotes the design matrix, i. e., the matrix containing the data used for fitting the model, $\boldsymbol{\beta}$ denotes the unknown regression parameters for the fixed effects, K is the variance covariance matrix for the random effect, which varies according to the model, and ε is the overall model error where τ^2 is the noise parameter.

Assuming a stationary GP, the covariance may be decomposed into a correlation matrix R and the GP variance σ^2, giving $K = \sigma^2 R$. As the sum of normally distributed random variables is itself normally distributed, the distribution of the response is closed and is denoted

$$\mathbf{y} \sim N(\mathbf{X}^\top \boldsymbol{\beta}, \sigma^2 [R + \lambda I]), \tag{2.7}$$

where $\lambda = \tau^2/\sigma^2$. This reparametrisation reduces the number of parameters to numerically optimise over by one. After optimisation, we obtain the model variance τ^2 by computing the product $\sigma^2\lambda$. The parameter estimates $\hat{\beta}$ and $\hat{\sigma}^2$ have closed form expressions in terms of λ and R, given in Section 2.2.4

Due to Gaussian conditioning rules, the posterior predictive distribution for the damages, $\mathbf{y}_*$, is denoted

$$\mathbf{y}_* | \mathbf{y} \sim N\left(m(\mathbf{x}_*), \sigma^2 v(\mathbf{x}_*, \mathbf{x}_*)\right),$$

$$m(\mathbf{x}_*) = \mathbf{X}_*^\top \boldsymbol{\beta} + R_*^\top R^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{2.8}$$

$$v(\mathbf{x}_*, \mathbf{x}_*) = R_{**} - Q^\top (\mathbf{X} R^{-1} \mathbf{X}^\top)^{-1} Q, \tag{2.9}$$

where $\mathbf{X}_*$ denotes the dataset used for testing, R_* denotes the correlation matrix between the observed and unobserved values, R_{**} denotes the correlation of the unobserved values and $Q = \mathbf{X}_* - \mathbf{X} R^{-1} R_*$. The model parameters are learnt using optimisation and maximum likelihood, see Section 2.2.4.

2.2.4 *Likelihood and estimation*

To learn the GP parameters,i. e., the scale parameter ϕ and the variance parameter σ^2, we optimise the marginal likelihood, given by the product of normal realisations. In our application, we assume no error

term in the model and $\mu = \mathbf{X}^\top \beta$. Given $\mathbf{y} \sim N(\boldsymbol{\mu}, \sigma^2 R)$, the likelihood takes the form

$$\begin{aligned} L(\mu, \sigma^2, \phi|\mathbf{y}) &= C|\sigma^2 R|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top (\sigma^2 R)^{-1}(\mathbf{y}-\boldsymbol{\mu}))\} \\ &= C(\sigma^2)^{-\frac{1}{2}}|R|^{-\frac{n}{2}} \exp\{-\frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\mu})^\top R^{-1}(\mathbf{y}-\boldsymbol{\mu}))\}, \end{aligned}$$

where $y_i \sim N(\mu, \sigma^2)$ and $\text{Corr}(y_i, y_j) = R_{ij}$. C corresponds to the constants not dependent on the parameters. For mathematical convenience, we model the log-likelihood, denoted

$$\mathcal{L}(\mu, \sigma^2, \phi|\mathbf{y}) = \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log|R| - \frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\mu})^\top R^{-1}(\mathbf{y}-\boldsymbol{\mu}).$$

where the constant terms which do not depend on the model parameters are omitted.

To estimate the parameters from the log-likelihood, $\mathcal{L}$, we use the method of maximum likelihood. In practice, we take derivatives of the log-likelihood with respect to each parameter and solve for the stationary points, having set the derivatives to zero. The derivatives of the log-likelihood with respect to the GP parameters are expressed as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} R^{-1}(\mathbf{y}-\boldsymbol{\mu}), \\ \frac{\partial \mathcal{L}}{\partial \phi} &= -\frac{1}{2}\text{tr}(R^{-1}\frac{\partial R}{\partial \phi}) + \frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\mu})^\top \frac{\partial R^{-1}}{\partial \phi}(\mathbf{y}-\boldsymbol{\mu})) \\ &= -\frac{1}{2}\text{tr}(R^{-1}\frac{\partial R}{\partial \phi}) + \frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\mu})^\top R^{-1}\frac{\partial R}{\partial \phi}R^{-1}(\mathbf{y}-\boldsymbol{\mu})), \end{aligned}$$

where $\text{tr}(\cdot)$ is the sums the diagonal elements of the matrix in the brackets. We have used the relation

$$\begin{aligned} RR^{-1} = I \implies \frac{dR}{d\phi}R^{-1} + R\frac{dR^{-1}}{d\phi} &= 0 \\ \implies R\frac{dR^{-1}}{d\phi} &= -\frac{dR}{dl}R^{-1} \\ \implies \frac{dR^{-1}}{d\phi} &= -R^{-1}\frac{dR}{d\phi}R^{-1}. \end{aligned}$$

The parameters for the linear effects $\boldsymbol{\beta}$ and the model variance σ^2, can be found in closed form. To estimate the parameters $\boldsymbol{\beta}$ and σ^2, we first differentiate the log-likelihood with respect to these parameters and set to zero.

The resulting maximum likelihood estimate for $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}R^{-1}\mathbf{X}^\top\right)^{-1}\left(\mathbf{X}R^{-1}\mathbf{y}\right).$$

Following the same procedure for σ^2, it's maximum likelihood estimate becomes

$$\hat{\sigma}^2 = \frac{R^{-1}(\mathbf{y} - \boldsymbol{\mu})}{n}.$$

We also note that

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2(\mathbf{X}^\top R^{-1}\mathbf{X})^{-1} \tag{2.10}$$

is the variance-covariance matrix of the fixed effects with the diagonals being the variances of $\hat{\boldsymbol{\beta}}$. This allows us to construct $100(1-\alpha)\%$ confidence intervals for $\hat{\boldsymbol{\beta}}$, in particular by taking the square root of the diagonal elements of Equation 2.10 for the standard error terms. A confidence interval for $\hat{\boldsymbol{\beta}}$ takes the form

$$\hat{\boldsymbol{\beta}} \pm z_{1-\frac{\alpha}{2}}\text{SE}(\hat{\boldsymbol{\beta}}), \tag{2.11}$$

where z is the quantile from the standard normal distribution, α denotes the Type 1 error rate, i. e., the probability of rejecting the null hypothesis when true. A typical value of α is 0.05, meaning if we generated samples from the true population, one in twenty of the estimates from those samples will not contain the proposed value. $\text{SE}(\hat{\boldsymbol{\beta}})$ denotes the standard error of $\hat{\boldsymbol{\beta}}$ found from We may also apply this method to estimate confidence intervals for the parameter $\hat{\sigma}^2$.

To estimate the GP parameters $\theta = \{\phi, \lambda\}$, we optimise the profile likelihood using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. We then substitute these estimates into the closed form expressions for $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ to obtain their estimates. To obtain the standard errors of an ML parameter, we compute the inverse of the Fisher information matrix, $\text{Var}(\theta) = [I(\theta)]^{-1}$, where $I(\theta) = -\mathbb{E}[H(\theta)]$ and $H(\theta) = \frac{\partial^2 \mathcal{L}}{\partial\theta\partial\theta'}$ is the matrix of second order derivatives of the negative log-likelihood, also known as the Hessian matrix.

2.2.5 *Optimisation using BFGS quasi-Newton Method*

To identify the GP, we must optimise over the GP parameters. There are several methods of optimisations available, with our application utilising the quasi-newton method [14].

For a given objective function $L(\theta)$ with initial point θ_0 and initial positive definite matrix $\mathbf{B}_0$, the BFGS iterations for $k = 0, 1, 2, \ldots$ are given by

$$\theta_{k+1} = \theta_k + \alpha_k \boldsymbol{\Delta}_k,$$

where $\boldsymbol{\alpha}_k > 0$ is the step-length, $\boldsymbol{\Delta}_k = -\mathbf{B}_k \nabla l(\theta_k)$ and $\mathbf{B}_k$ is a symmetric positive definite matrix given by the iteration

$$\mathbf{B}_{k+1} = (\mathbf{I} - \frac{s_k a_k^\top}{a_k^\top s_k})\mathbf{B}_k(\mathbf{I} - \frac{s_k a_k^\top}{a_k^\top s_k}) + \frac{s_k a_k^\top}{a_k^\top s_k}, \tag{2.12}$$

where $s_k = \theta_{k+1} - \theta_k$ and $a_k = \nabla l(\theta_{k+1}) - \nabla l(\theta_k)$. The corresponding approximate Hessian iteration is given by

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{a_k a_k^\top}{a_k^\top s_k} - \frac{\mathbf{H}_k s_k [\mathbf{H}_k s_k]^\top}{s_k^\top \mathbf{H}_k s_k}. \tag{2.13}$$

Equations (2.12) and (2.13) are implemented iteratively until a stopping criteria is reached or the maximum number of iterations is met. The output of R's `optim` function gives an exit code which reveals the status of the optimisation once complete.

2.2.6 *Gaussian Processes for Large Datasets*

GPs are commonly used for modelling spatial data [16], but are burdened with computational challenges for large scale datasets [77]. Spatial model fitting requires inverting an $n \times n$ matrix for a dataset of size n. Evaluations of the likelihood typically require $\mathcal{O}(n^3)$ operations and $\mathcal{O}(n^2)$ memory, which makes inverting large scale matrices using traditional methods impractical.

There are several methods for invert an $n \times n$ matrix of the form $A + CBC^T$. One such method is the Woodbury identity, denoted

$$(A + CBC^T)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^T A^{-1} C)^{-1} C^T A^{-1}. \tag{2.14}$$

For the Woodbury identity to be computationally efficient, A has to have dimension significantly lower than n, as the inverse of A and the inverse of $(B^{-1} + C^T A^{-1} C)$ is needed.

We also make use of the matrix determinant lemma for increased computational efficiency, denoted

$$\det(A + CBC^T) = \det(B^{-1} + C^T A^{-1} C) \det(B) \det(A). \tag{2.15}$$

In addition, the Cholesky decomposition is applied form matrix inversion. The Cholesky decomposition states that if a matrix A is positive definite, there exists a unique upper triangular matrix, U, such that $A = U^T U$. This makes inversion more practical for large datasets since we are now inverting triangular matrices. The inversion becomes $A^{-1} = U^{-1}(U^T)^{-1}$. Since these triangular matrices are sparse, less computations are required. However, computing the Cholesky decomposition still requires computational expense of $\mathcal{O}(n^3)$.

Unless the kernel has a convenient structure, predictive modelling becomes impractical using traditional matrix inversion techniques. Methods have been developed to overcome this bottleneck. [84] explained the idea of subset of regressors approach, which essentially reduces the dimensionality of the problem through spectral decomposition of the covariance matrix. [73] proposed a predictive model by imputing $f(\cdot)$ conditional on values at certain knot locations but deciding knot locations itself is problematic. The method of linear projections

was adopted by [5] in conjunction with reduced-rank approximations. We now describe some of the large-scale matrix inversion methods used within the literature.

2.2.6.1 *Tapering method*

The idea behind covariance tapering is to treat location pairs as independent if their associated covariance is less than some pre-determined value [46]. If correlations between distant observations are negligible, then the tapering idea is suitable. Tapering allows us to create sparse matrices by multiplying element-wise the a covariance matrix with a compactly supported covariance function, resulting in a positive definite covariance matrix. Computational expense is reduced by replacing negligible covariances with zeros. It is, therefore, important to decide between the level of computational efficiency and predictive performance.

We let $K(\theta)$ be a $n \times n$ covariance matrix with the $(i,j)^{th}$ element equal to $k(|x_i - x_j|;\theta), i,j = 1,\dots,n$. Denoting the tapering function, $k_{tap}(h;\gamma)$, which is a type of compact correlation function, and γ denotes the tapering range and is strictly positive. It has the property $k_{tap}(h;\gamma) = 0$ if $h \geqslant \gamma$. Smaller values of γ give more sparsity, however greater sparsity means reduced accuracy, therefore a suitable balance is required. The tapered covariance function at distance h is denoted by

$$\tilde{k}(h;\theta,\gamma) = k(h;\theta) \cdot k_{tap}(h;\gamma), \qquad h > 0, \tag{2.16}$$

and the tapered covariance matrix is

$$\tilde{K}(\theta,\gamma) = K(\theta) \circ K_{tap}(\gamma), \tag{2.17}$$

where $\circ$ denotes the Hadamard product and $K_{tap}(\gamma)$ denotes the covariance matrix corresponding to $k_{tap}(|x_i - x_j|;\gamma)$.

In light of parameter estimation, covariance matrices in the log-likelihood are now replaced with a tapered covariance, i. e.,

$$\mathcal{L}(\mu,\sigma^2,\phi|\mathbf{y}) \propto \frac{1}{2}\mathbf{y}^\top[\tilde{K}(\theta,\gamma)]^{-1}\mathbf{y} - \frac{1}{2}\log|\tilde{K}(\theta,\gamma)|. \tag{2.18}$$

Using a tapered covariance function in the log-likelihood results in a biased likelihood, meaning there is no guarantee the estimator which maximises Equation (2.18) is asymptotically unbiased when the true covariance matrix is $K(\theta)$

Covariance tapering works well for small-scale dependencies, however, is not as effective for long-range dependencies [77]. With tapering, computational complexity is $\mathcal{O}(nl^2)$, where l is the average number of non-zero entries in each row of $\tilde{K}$[35].

2.2.6.2 *Predictive process*

We first consider a GP model,

$$y(x) = f(x) + \varepsilon(x), \tag{2.19}$$

where $f(x)$ is a GP and $\varepsilon(x)$ is the independent process, with $f(x) \sim GP(0, K(x, x'; \theta))$ and $\varepsilon \sim GP(0, \tau^2 I(x = x'))$. Along with specifying a suitable covariance function for the model, predictive process is dependent on the choice of knot locations, $x_1^*, \ldots, x_m^*$, with $m < n$. [6] identified several open questions regarding the spatial design for placement of knots. Ideally, the specified number of knots are placed in such a way that spatially averaged prediction variance is minimized, noting that a predictive process with smaller predictive variance might be viewed as better approximation to the parent process [32]

Specification of the m knots is required at certain locations. An important consideration is how we decide on the knot locations. We may wish to chose a random set of knots across the input space or we may wish to select our knots around the most concentrated area of data samples. Knots may also be chosen through clustering fingerprints and sampling from each cluster. Not only are the locations of the knots important, but also are the number of knots we wish to model with. Ideally, we wish to reduce the dimensionality of the problem without sacrificing model accuracy. We therefore approximate $\tilde{f}(x)$ by selecting knot locations which accurately represent the input space, whilst reducing the dimensionality.

Considering a set of knots, $\mathcal{X}^* = \{x_1^*, \ldots, x_n^*\}$, which is a subset of the input space, $\mathcal{X}$. From Equation (2.19), it follows $\mathbf{f}^* = [f(x_i^*)]_{i=1}^m \sim N\{0, K^*(\theta)\}$, where $K^*(\theta) = [K(x_i^*, x_j^*; \theta)]_{j=1}^m$ is the $m \times m$ covariance matrix. At a site, x_0, spatial interpolation is defined by $\tilde{f}(x_0) = \mathbb{E}[f(x_0)|\mathbf{f}^*] = k_0^\top(\theta) K^{*-1}(\theta)\mathbf{f}^*$, where $k_0(\theta) = [k_0(x_0\theta)]_{j=1}^m$. The GP $\tilde{f}(x) \sim GP(0, \tilde{K}(\cdot, \cdot))$ for a single site interpolator has covariance

$$\tilde{k}(x, x'; \theta) = k^\top(x; \mathcal{X}^*) K^{*-1}(\theta) k(\mathcal{X}^*, x'). \tag{2.20}$$

We define

$$\tilde{y}(x) = \tilde{f}(x) + \varepsilon(x). \tag{2.21}$$

The process $\tilde{y}(x)$ has covariance matrix

$$\tilde{K}_y = \tilde{K}_f + \tau^2 I,$$

where $\tilde{K}_f = \tilde{k}(x, x; \theta)$. The inverse and determinant of $\tilde{K}_y$ are obtained using Equation (2.14) and (2.15). This method is more efficient for long range dependencies, and reduces computational complexity to $\mathcal{O}(nm^2)$.

The predictive process is advantageous when capturing long-range dependencies, but fails to capture local, small-scale correlations [77].

2.2.6.3 *Full scale approximation*

The full scale approximation is a mixture of both tapering and predictive process which overcomes their individual short comings [77]. This is achieved by decomposing the GP into a reduced rank process to capture the large scale dependencies and apply tapering to capture small scale dependencies. The result is accurate approximations to both small are large scale dependencies [77]. Decomposing the GP in Equation (2.19),

$$f(x) = f_1(x) + f_x(x), \tag{2.22}$$

where the reduced rank approximation of $f(x)$ is denoted by $f_1(x)$ and the residual process is defined to be $f_x(x) = f(x) - f_1(x)$. The predictive process can therefore be expressed as

$$f_1(x) = K^\top(x, \mathcal{X}^*)K^{*-1}\mathbf{f}^*, \tag{2.23}$$

with covariance matrix

$$\begin{aligned} K_1(x, x') &= K^\top(x, \mathcal{X}^*)K^{*-1}K(x', \mathcal{X}^*) && (2.24)\\ &= \mathrm{Cov}(f_1(x), f_1(x')). && (2.25) \end{aligned}$$

The residual process has the following covariance matrix

$$\hat{K}(x, x') = K(x, x') - K^\top(x, \mathcal{X}^*)K^{*-1}K(x', \mathcal{X}^*). \tag{2.26}$$

see (17) from [77]. Therefore, the tapering process becomes:

$$K_x(x, x') = \{K(x, x') - K^\top(x, \mathcal{X}^*)K^{*-1}K(x', \mathcal{X}^*)\} \circ K_{tap}(h; \gamma) \tag{2.27}$$

which encompasses both the predictive process and tapering. We note this takes the form suitable for inversion via the Woodbury identity. Computational complexity is now $\mathcal{O}(nl^2 + nm^2)$.

2.2.7 *Advantages and disadvantages of fitting Gaussian processes*

Like many machine learning methods, GPs are equipped with useful properties and bottlenecks.

Some of their advantages are:

- the closed form expressions for the predictive mean and variance and that data points may be easily predicted that have not yet been observed. If a data point is to be predicted which is unlike any data point already observed, the GP prediction will return high uncertainty for the predicted effect,
- through the use of the covariance function, GPs are adaptable to the data [28]. With GPs, we are able to combine multiple kernels, both additively and multiplicatively, which provides a wealth of models to choose from,

- the prediction of data is probabilistic, which allows us to construct empirical confidence intervals and inform decisions on whether one should refit the prediction in some region of interest [29], and
- working with GPs is made feasible by the marginalisation property, that is, we can marginalise over infinitely many variables, even those we have not seen [72].

Some of the drawbacks of GPs are:

- they lack sparsity, i. e., they use whole feature space, resulting in a computationally expensive process [25],
- they can be computationally expensive in the prediction phase (see Section 2.2.6.) Since we are required to invert the covariance matrix, this becomes computationally impractical for large datasets, as evaluations of the posterior involve $\mathcal{O}(n^3)$ computations [5]. We therefore require alternative matrix inversion methods, and
- they are also limited as the user is required to choose the kernel based on the data type and there are a wealth kernels to choose from [85].

2.3 LATENT-VARIABLE MODELS

Many statistical models contain unobservable variables known as latent variables and act as a way of introducing correlation in the model. For example, suppose two coffee brands are to be reviewed by n reviewers at two different temperatures x_{i1} and x_{i2}. The ith reviewer will then produce a rating for the two different temperatures, denoted y_{i1}, y_{i2}. Ratings between reviewers are considered independent, given a reviewer is not informed of anyone's opinion on the taste, whereas the ratings for the same reviewer will be dependant. To include the unmeasurable effects for each reviewer, we introduce latent variables $u_1, \ldots, u_n$ and assume the observed ratings for reviewer i are independent conditional on the value of u_i. However, the ratings are correlated as we do not observe u_i unconditionally. A possible model in this case would be

$$\begin{gathered} y_{ij}|u_i \overset{iid}{\sim} N(\mu_{ij}, \sigma^2),\ \mu_{ij} = x_{ij}^T\beta + u_i,\ i = 1, \ldots, n,\ j = 1, 2, \\ u_i \overset{iid}{\sim} N(0, \tau^2). \end{gathered} \tag{2.28}$$

This model implies that, marginally, $\mathrm{Cov}(y_{i1}, y_{i2}) = \tau^2$, so indeed, the ratings of the same reviewer are correlated.

Often, statistical models such as the model in Equation (2.28) can be expressed in vector form. Let $\mathbf{y} = (y_{11}, y_{12}, y_{21}, \ldots, y_{n2})$, $\mathbf{u} =$

$(u_1, \ldots, u_n)$, $\boldsymbol{\mu}$ denotes the $2n$ dimensional vector and X the $2n \times p$ design matrix. Then,

$$\begin{aligned} \mathbf{y}|\mathbf{u} &\sim N_{2n}(X\beta + P\mathbf{u}, \sigma^2 I_{2n}), \\ \mathbf{u} &\sim N_n(0, \tau^2 I_n). \end{aligned}$$

In the above, I_n denotes the $n \times n$ identity matrix and $P = I_n \otimes \mathbf{1}_2$, where $\otimes$ is the Kronecker product and $\mathbf{1}_k$ is the k dimensional vector of ones. The latent variable model may be written in vector form,

$$f(\mathbf{y}) = \prod_{i=1}^{n} f(y_i) \tag{2.29}$$

and the joint density of (y, u) is

$$f(\mathbf{y}, \mathbf{u}) = f(\mathbf{y}|\mathbf{u})f(\mathbf{u}) = \prod_{i=1}^{n}\prod_{j=1}^{2} f(y_{ij}|u_i) \prod_{i=1}^{n} f(u_i). \tag{2.30}$$

It is necessary to find the likelihood of the observed data $\mathbf{y}$ for inference. For Equation (2.30), it is more computationally demanding due to the presence of the latent variable $\mathbf{u}$. We, therefore, must compute the integral

$$f(\mathbf{y}) = \int f(\mathbf{y}, \mathbf{u}) du = \int f(\mathbf{y}|\mathbf{u})f(\mathbf{u}) d\mathbf{u} \tag{2.31}$$

In effect we are integrating out the latent variables in the model. In the case of the model in Equation (2.28), we see that both $f(\mathbf{y}|\mathbf{u})$ and $f(\mathbf{u})$ are the PDF of two normally distributed random variables. When the expression for $f(\mathbf{y}|\mathbf{u})$ is non-normal, then Equation (2.30) will not have a closed form expression and we have to resort to numerical methods of approximation, such as Monte-Carlo integration.

2.3.1 *Monte-Carlo methods*

Monte-Carlo integration is performed by computing expectations of functions of simulated random variables. We see that the integral in Equation (2.30) can be expressed as an expectation

$$f(\mathbf{y}) = \mathbb{E}[f(\mathbf{y}|\mathbf{u})] \tag{2.32}$$

where the expectation is taken with respect to $\mathbf{u}$. Then given random samples $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(N)}$ from the density $f(\mathbf{u})$, we approximate

$$\hat{f}_{MC}(\mathbf{y}) = \frac{1}{N}\sum_{l=1}^{N} f(\mathbf{y}|\mathbf{u}^{(l)}). \tag{2.33}$$

Monte-Carlo simulations provide probabilistic insight into each outcome and allow for graphical aids to show a range of possible values. However, since the method relies on a proposed distribution, a poor distributional choice will result in unreliable estimates. McCulloch [62] provided several approaches for maximising the likelihood. In addition, Monte-Carlo methods are notably slow in their application.

2.3.2 *Laplace approximation*

When we have a large dataset to sample from, the Monte-Carlo method can be very accurate, but can also be computationally intensive if the dimension of $\mathbf{u}$ is large. Another approximation to Equation (2.30) is through the Laplace approximation. The exact integral we wish to approximate is the following

$$\begin{aligned} f(\mathbf{y}) &= \int_{-\infty}^{\infty} f(\mathbf{y}|\mathbf{u}) f(\mathbf{u}) d\mathbf{u} && (2.34)\\ &= \int_{-\infty}^{\infty} e^{-(-\log(f(\mathbf{y}|\mathbf{u}) f(\mathbf{u})))} d\mathbf{u} && (2.35)\\ &= \int_{-\infty}^{\infty} e^{-(-\log(f(\mathbf{y}|\mathbf{u}))-\log(f(\mathbf{u})))} d\mathbf{u} \\ &= \int_{-\infty}^{\infty} e^{-g(\mathbf{u})} d\mathbf{u}, \end{aligned}$$

where $g(\mathbf{u}) = -\log(f(\mathbf{y}|\mathbf{u})) - \log(f(\mathbf{u}))$. We assume that $g(\mathbf{u}) \to \infty$ as $n \to \infty$. Applying the Taylor expansion to $g(\mathbf{u})$ in the neighbourhood of its minimiser $\hat{\mathbf{u}}$, we obtain

$$g(\mathbf{u}) \approx g(\hat{\mathbf{u}}) + g'(\hat{\mathbf{u}})(\mathbf{u} - \hat{\mathbf{u}}) + \frac{1}{2}(\mathbf{u} - \hat{\mathbf{u}})^\top g''(\hat{\mathbf{u}})(\mathbf{u} - \hat{\mathbf{u}}).$$

Since the gradient of a function about its maximiser is 0, i. e., $g'(\hat{\mathbf{u}}) = 0$, the second term vanishes, leaving

$$g(\mathbf{u}) \approx g(\hat{\mathbf{u}}) + \frac{1}{2} g''(\hat{\mathbf{u}})(\mathbf{u} - \hat{\mathbf{u}})^\top \hat{H}(\mathbf{u} - \hat{\mathbf{u}}).$$

Here we have made use of the fact that $g''(\hat{\mathbf{u}}) = \hat{H}$, where $\hat{H}$ denotes the Hessian matrix at $\hat{\mathbf{u}}$. Therefore, the integral in Equation (2.35) can be approximated using the expression

$$\begin{aligned} I &\approx \int_{-\infty}^{\infty} e^{-(g(\hat{\mathbf{u}}) + \frac{1}{2}(\mathbf{u}-\hat{\mathbf{u}})^\top g''(\hat{\mathbf{u}})(\mathbf{u}-\hat{\mathbf{u}}))} d\mathbf{u} \\ &= e^{-g(\hat{\mathbf{u}})} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(\mathbf{u}-\hat{\mathbf{u}})^\top \hat{H}(\mathbf{u}-\hat{\mathbf{u}})} d\mathbf{u} \\ &= e^{-g(\hat{\mathbf{u}})} \left(\frac{(2\pi)^n}{|\hat{H}|}\right)^{\frac{1}{2}} \int_{-\infty}^{\infty} \left(\frac{|\hat{H}|}{(2\pi)^n}\right)^{\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{u}-\hat{\mathbf{u}})^\top \hat{H}(\mathbf{u}-\hat{\mathbf{u}})} d\mathbf{u} \\ &= e^{-g(\hat{\mathbf{u}})} (2\pi)^{\frac{n}{2}} |\hat{H}|^{-\frac{1}{2}} \cdot 1 && (2.36) \end{aligned}$$

where the integrand corresponds to the multivariate normal density with mean $\hat{\mathbf{u}}$ and variance-covariance matrix $\hat{H}^{-1}$.

A disadvantage of the Laplace approximation is that it is essentially uncontrolled, i. e., the Hessian may provide poor approximation to the true shape of the posterior. The peak could be much broader or narrower than the Hessian indicates, or it could be a skew peak, while the Laplace approximation assumes it has elliptical contours.

2.4 ORDINAL RESPONSE MODELS

2.4.1 *Multinomial distribution*

To define a multinomial GLM, we must first define a multinomial experiment. A random experiment is called a multinomial experiment with parameters $(m, k, [\pi_1, \dots, \pi_k])$ if it satisfies the following conditions.

1. The number of trials, m, is fixed,
2. all observations are contained within one of the k categories,
3. observations are independent,
4. the probability of observing each category $j, j = 1, \dots, k$, is the same for each trial, and equals π_j,
5. all probabilities sum to one, i.e. $\pi_1 + \pi_2 + \cdots + \pi_k = 1$.

The multinomial distribution is an extension to the binomial distribution, where we have more than two possible outcomes. The random variable, $\mathbf{Y}$, which counts the number of occurrences of each category of a multinomial experiment with parameters $(m, k, [\pi_1, \dots, \pi_k])$ is said to follow the multinomial distribution, usually denoted

$$\mathbf{Y} \sim \text{Multinomial}(m, k, \boldsymbol{\pi}),$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$. If $\mathbf{Y} \sim \text{Multinomial}(m, k, \boldsymbol{\pi})$, then $\mathbf{y} = [y_1, \dots, y_k]$, is a k-dimensional vector where the jth element of the vector is the number of times category j was observed in m trials. This means $y_1 + \cdots + y_k = m$, for each $y_j \in \{0, 1, \dots, m\}$.

The probability mass function (PMF) of a multinomial random variable $\mathbf{Y}$ is denoted

$$\begin{aligned} f(\mathbf{y}; \boldsymbol{\pi}) &= \frac{m!}{y_1! \cdots y_k!} \pi_1^{y_1} \cdots \pi_k^{y_k} \\ &\propto \pi_1^{y_1} \cdots \pi_k^{y_k}, \end{aligned}$$

where the constant terms are omitted under proportionality.

2.4.2 *Multinomial likelihood*

Suppose we observe n outcomes $y_1, \dots, y_n$ from a multinomial experiment with parameters (m, k, π) such that the categories are ordered. The contribution of the ith datum to the log-likelihood for the multinomial distribution is expressed as

$$\mathcal{L}(\pi_i; y_i) = \sum_{j=1}^{k} \mathbf{y}_{ij} \log \pi_{ij},$$

where each y_i is the observed damage and π_i is the vector of probabilities for the ith experiment. The constraints on the observations and probabilities are

$$\sum_j \mathbf{y}_{ij} = m_i \text{ and } \sum_j \pi_{ij} = 1,$$

where there are m replicates. The cumulative probability of observing up to class j in a single trial is denoted γ_j, and π_j gives the probability of observing class j only. Therefore,

$$\begin{aligned} \gamma_1 &= \pi_1, \\ \gamma_2 &= \pi_2 + \pi_1, \\ &\vdots \\ \gamma_j &= \pi_j + \pi_{j-1} + \cdots + \pi_1 \end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned} \pi_1 &= \gamma_1, \\ \pi_2 &= \gamma_2 - \gamma_1, \\ &\vdots \\ \pi_j &= \gamma_j - \gamma_{j-1}. \end{aligned}$$

The derivative of the log-likelihood with respect to the class probabilities is denoted

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = \frac{y_j - m\pi_j}{\pi_j}$$

It is also of interest to find the derivative of the negative log-likelihood with respect to the cumulative probabilities. We perform this by using the chain rule,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \gamma_j} &= \sum_{j'=1}^{k} \frac{\partial \mathcal{L}}{\partial \pi_{j'}} \frac{\partial \pi_{j'}}{\partial \gamma_j} \\ &= \frac{\partial \mathcal{L}}{\partial \pi_1}\frac{\partial \pi_1}{\partial \gamma_j} + \frac{\partial \mathcal{L}}{\partial \pi_2}\frac{\partial \pi_2}{\partial \gamma_j} + \cdots + \frac{\partial \mathcal{L}}{\partial \pi_j}\frac{\partial \pi_j}{\partial \gamma_j} + \frac{\partial \mathcal{L}}{\partial \pi_{j+1}}\frac{\partial \pi_{j+1}}{\partial \gamma_j} \\ &= \frac{\partial \mathcal{L}}{\partial \pi_1} \cdot 0 + \frac{\partial \mathcal{L}}{\partial \pi_2} \cdot (0) + \cdots + \frac{\partial \mathcal{L}}{\partial \pi_j} \cdot 1 + \frac{\partial \mathcal{L}}{\partial \pi_{j+1}} \cdot (-1) \\ &= \frac{\partial \mathcal{L}}{\partial \pi_j} - \frac{\partial \mathcal{L}}{\partial \pi_{j+1}}. \end{aligned}$$

We now compute the second derivative of the log likelihood with respect to π and γ

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \pi_{ij}} &= \frac{\tilde{\mathbf{y}}_i - m\pi_{ij}}{\pi_{ij}} \frac{\partial \mathcal{L}}{\partial \gamma_{ij}} \\ &= \frac{\partial \mathcal{L}}{\partial \pi_{ij}} - \frac{\partial \mathcal{L}}{\partial \pi_{ij+1}} \end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned} &\frac{\partial^2 \mathcal{L}}{\partial \pi_{ij}\partial \pi_{ij'}} = \frac{\tilde{\mathbf{y}}_i}{\pi_{ij}^2}, \qquad j = j', \text{else } 0 \\ &\frac{\partial^2 \mathcal{L}}{\partial \gamma_{ij}\partial \gamma_{ij'}} = 0, j' \notin \{j-1, j, j+1\} \end{aligned}$$

$$
\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial \gamma_{ij} \partial \gamma_{ij-1}} &= \sum_{j'=1}^{k} \frac{\partial}{\partial \pi_{ij'}} \left(\frac{\partial \mathcal{L}}{\partial \gamma_{ij}} \right) \frac{\partial \pi_{ij'}}{\partial \gamma_{ij-1}} \\
&= \sum_{j'=1}^{k} \frac{\partial}{\partial \pi_{ij'}} \left(\frac{\partial \mathcal{L}}{\partial \pi_{ij}} - \frac{\partial \mathcal{L}}{\partial \pi_{ij+1}} \right) \frac{\partial \pi_{ij'}}{\partial \gamma_{ij-1}} \\
&= \sum_{j'=1}^{k} \left(\frac{\partial^2 \mathcal{L}}{\partial \pi_{ij} \partial \pi_{ij'}} - \frac{\partial^2 \mathcal{L}}{\partial \pi_{ij+1} \partial \pi_{ij'}} \right) \frac{\partial \pi_{ij'}}{\partial \gamma_{ij-1}} \\
&= \frac{\partial^2 \mathcal{L}}{\partial \pi_{ij}^2} \frac{\partial^2 \pi_{ij}}{\partial \gamma_{ij-1}} - \frac{\partial^2 \mathcal{L}}{\partial \pi_{ij+1}^2} \frac{\partial^2 \pi_{ij+1}}{\partial \gamma_{ij-1}} \\
&= \frac{\tilde{\mathbf{y}}_i}{\pi_{ij}^2} \cdot (-1) - \frac{\partial^2 \mathcal{L}}{\partial \pi_{ij+1}^2} \cdot (0) \\
&= -\frac{\tilde{\mathbf{y}}_i}{\pi_{ij}^2},
\end{aligned}
$$

where $\tilde{\mathbf{y}}_i = 1$ if the observed class is the jth class.

2.4.3 *Cumulative link models*

Given some observed data $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$, where each y_i is the outcome of a multinomial experiment with parameters $(1, k, \boldsymbol{\pi})$ and the k classes are ordered $1 < \ldots < k$, the objective is to estimate the probability of observing category j, $\pi_{ij} = \pi_j(x_i)$, where $\mathbf{x}_i$ is a vector of p predictors. We denote the predicted probabilities for observing classes $1, \ldots, k$ for a given $\mathbf{x}$ as $\hat{\pi}_1(\mathbf{x}), \ldots, \hat{\pi}_k(\mathbf{x})$. The class containing the highest estimated posterior probability is chosen to be the predicted class.

In order to estimate the probabilities of each class, our model needs to relate the linear predictor, denoted $\eta = \boldsymbol{\beta}^\top \mathbf{x}_i$, to these probabilities, usually through some link function. A model which achieves this is denoted

$$
\begin{aligned}
G(\gamma_{ij}) &= \alpha_j + \boldsymbol{\beta}^\top x_i \qquad j = 1, \ldots, k-1 \\
&= \eta_{ij},
\end{aligned}
\tag{2.37}
$$

where $G(\cdot)$ denotes the link function, $\gamma_{ij} = P(Y_i \leqslant j)$ and we have k ordered classes. We write Y_i for the observed class corresponding to y_i. For example, $y_i = (0, 1, 0)$ implies $\gamma_i = 2$. We note that the intercepts $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{k-1})$ are strictly increasing, a feature specific to ordinal response outcomes. The role of $\boldsymbol{\alpha}$ is to determine the cut-off points for the response variable between each class, which we learn from the data through optimisation.

2.4.4 *Choice of cumulative link function*

For the logit model, cumulative probabilities, i. e., the probability the random variable takes any value up to the j^{th} class, may be expressed as

$$\begin{aligned} P(Y_i \leqslant j) &= \frac{\exp(\eta_{ij})}{1+\exp(\eta_{ij})}, \qquad j = 1, \ldots, k-1 \\ &= \frac{1}{\exp(-\eta_{ij})+1}. \end{aligned}$$

If the response variable is ordinal in nature [1, 62], individual class probabilities are obtained by subtracting the cumulative probabilities of consecutive classes, i. e.,

$$\begin{aligned} P(Y_i = j) &= P(Y_i \leqslant j) - P(Y_i \leqslant j-1) \\ &= \frac{\exp(\eta_{ij})}{1+\exp(\eta_{ij})} - \frac{\exp(\eta_{ij-1})}{1+\exp(\eta_{ij-1})} \\ &= \pi(x_i). \end{aligned}$$

When fitting GLMs to ordinal response outcomes, there are several link functions, denoted by $G(\cdot)$, at one's disposal. Link functions relate the mean of the response variable to the linear predictor. Thier choice is motivated by the behaviour within the data. These behaviours may be non symmetrical, for instance, shifting values by some constant amount at the lower end of the linear predictor may have a different impact on the class assignment than the same shift for greater values of the linear predictor. Table 2.1 shows some common choices for link functions where $\Phi(\gamma_j) = \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\gamma_j} e^{-z^2/2} dz$ and z are the quantiles of the standard normal distribution. The log-log and complementary log-log are asymmetrical links, and are suited for behavioural differences at the extremes of the linear predictor. There are other link functions at

Link	$G(\gamma)$
Logit	$\log \frac{\gamma}{1-\gamma}$,
Probit	$\Phi^{-1}(\gamma)$,
Log-log	$\log(-\log(\gamma))$,
Complementary log-log	$\log(-\log(1-\gamma))$,

Table 2.1: Four link functions used for GLMs. Φ is the standard normal CDF.

one's disposal. However, link functions must be continuous, monotonic and differentiable, so standard CDFs suffice. Figure 2.8 illustrates the different behaviours of the link functions given in Table 2.1. We see the log-log and complementary log-log are asymmetrical functions.

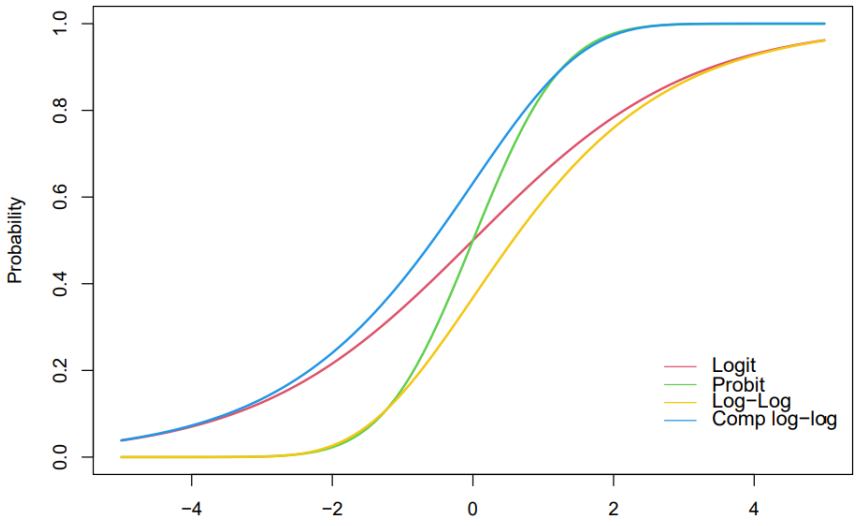


Figure 2.8: Link functions plotted against increasing values of the linear predictor

2.5 ENSEMBLE METHODS

2.5.1 *Decision Trees*

Classification trees, commonly referred to as decision trees, are a simple class of models which make predictions based on a series of decisions. They may be represented as a tree, where the nodes are labelled as features, edges are labelled as either a single value or a set of values, and the leaves are class labels. Decision trees can be regarded as a multistage decision making process, where different subsets of features are the decision criteria and various stages of the tree. These decision trees are somewhat of a feature selection method since they determine the most important features for the classification boundary. Instead of using all features present within the data, subsets of features are chosen at different levels of the tree. The three characteristics of a decision tree are:

- **Root node**: This is usually the top of the tree with no incoming edges
- **Internal nodes**: these contain a single incoming edge and multiple outgoing edges
- **Leaf/terminal node**: these are nodes with a single incoming edges and no outgoing edges

The decision to split at each node is made according to the metric called purity. A node is 100 % impure when a node is split evenly 50/50 and 100 % pure when all of its data belongs to a single class. See Figure 2.9 for an illustrative example of a decision tree.

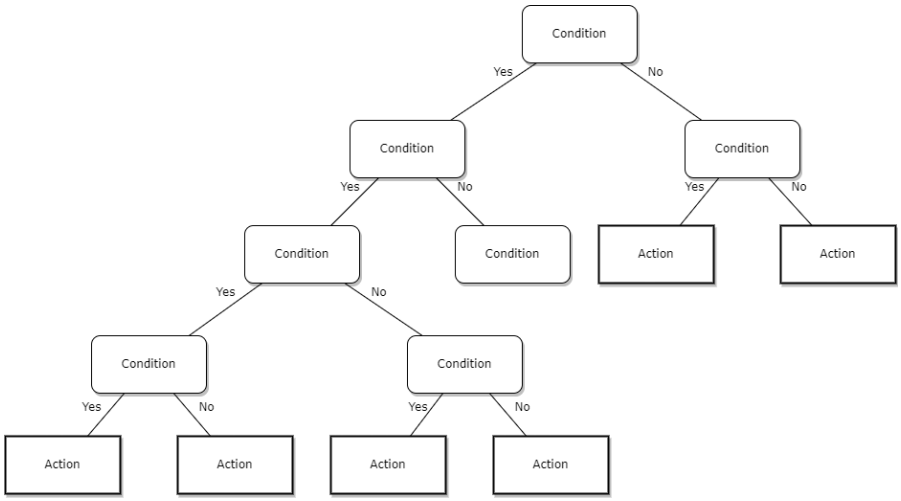


Figure 2.9: A generic decision tree used in a classification setting

2.5.2 *Bootstrap aggregating*

Bootstrap aggregating (bagging) [12] is a popular ensemble method used in classification and regression. The method involves selecting samples with replacement from the training set, called bags, and using these samples as individual training sets. The optimal sample size of the training set for bootstrap aggregating is around 60 %, see Figure 2.10. Having trained the models on the samples, predictions are made on the upsampled data. In a regression setting, predictions are aggregated and a majority vote is made in a classification setting. More formally, given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging selects B random samples with replacement and fits trees to the samples. In regression, a global prediction is made by computing $\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x_*)$ where x_* are the unseen samples and b is a dummy variable denoting the b^{th} function applied to the bootstrapped dataset. In classification, we select the prediction with the most votes.

Bootstrapping has been shown to improve model performance by decreasing the variance of the model without increasing the bias [12]. This is the result of averaging over many decision trees, since a single tree is sensitive to noise in the training set. Estimates of the prediction uncertainty can be calculated as the standard deviation of the prediction of a single decision tree for the unseen samples, x_*.

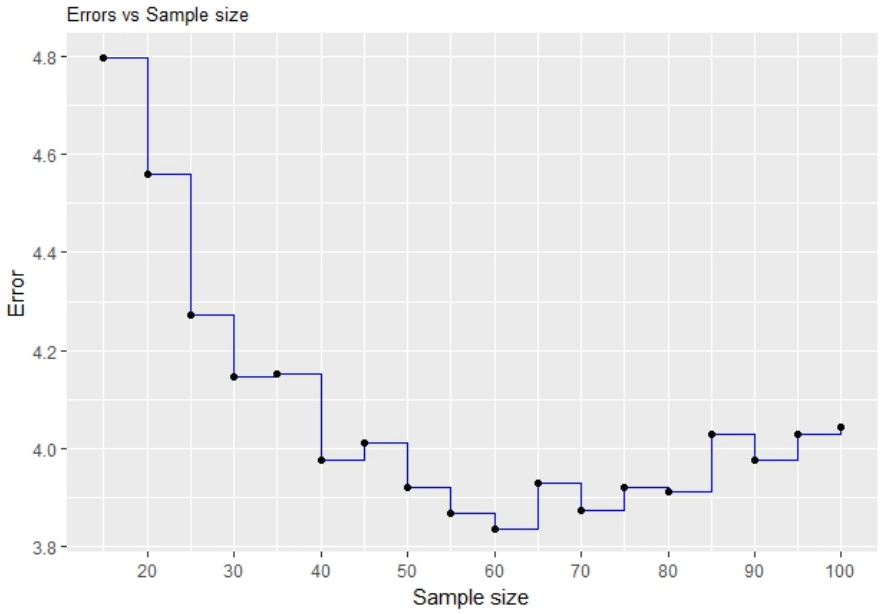


Figure 2.10: Out of bag error rate vs sample sizes with replacement

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x_*) - \hat{f})^2}{B-1}} \tag{2.38}$$

The number of samples, B, can be found through cross validation or through the out of bag error. The out of bag error is the average error on the training samples that were not selected in the bootstrap sample. Depending on the size of the data, it is customary to use several hundred, or even thousands of decision trees.

2.5.3 *Random Forests*

Random forests are a supervised ML technique widely used for classification. The random forest classifier is an ensemble learner built on decision trees. A drawback of decision trees is that they often lead to complex trees and over-fitting. Random forests reduce this risk as it aggregates over many decision trees by making use of bootstrap aggregation. When used for classification, a random forest obtains a class prediction from each decision tree, then classifies based on the majority vote, see Figure 2.11 Decision trees are suitable for bagging as they are able to capture complex interaction structures within the data with relatively low bias (given the tree if sufficiently deep.)

In decision trees, nodes are generated by using the best split among all regressors, assessed through the Gini index or entropy, whereas in random forests the nodes are constructed by the best split among a sample of regressors. Selecting a subset of features is sometimes

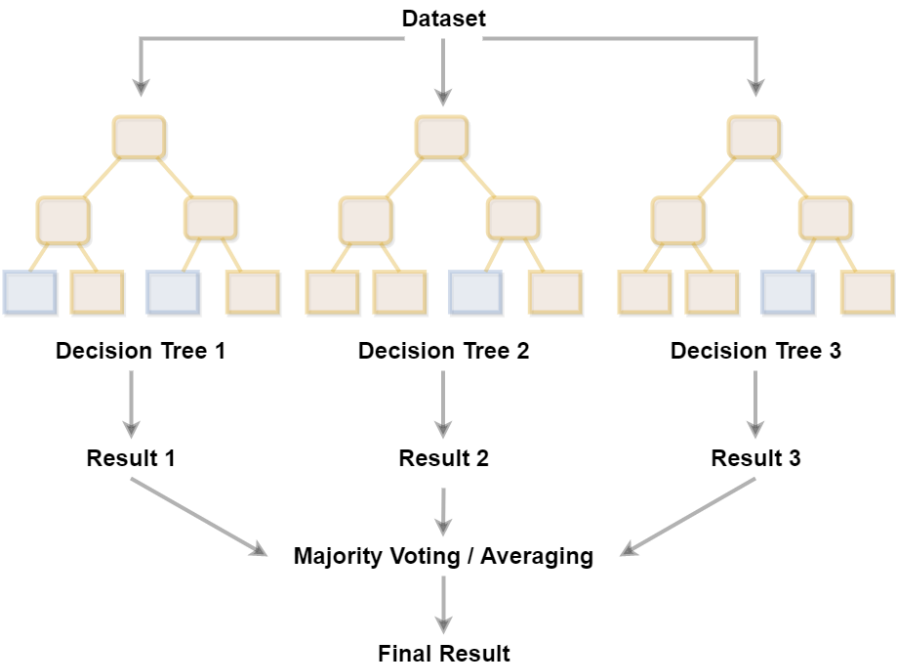


Figure 2.11: A series of decision trees forming a random forest

referred to as "feature bagging". Random forests have shown comparable performance to other classifiers such as SVMs and neural network (NN)s, [56] and are immune to over-fitting [13].

2.5.4 *Model Averaging/ Stacking*

Given several models are under consideration, a single classifier may be formed using the predictions of each individual model which has the ability to outperform the performance of the individual models in question. This approach is motivated by Bayesian model averaging. We may wish to make inference on a quantity of interest, Δ , which may be a future observation, $\mathbf{y}_*$, a regression coefficient, $\boldsymbol{\beta}_j$, the indicator of the regression coefficient being non-zero j, or the posterior density of the regression coefficient $\Pr(\boldsymbol{\beta}_j|\mathbf{y})$, where $\mathbf{y}$ are the observed data. The posterior density for Δ is obtained as a weighted average for the densities for Δ under each model $\mathcal{M}_i$ and the weights are the posterior probabilities of the models $\mathcal{M}_i$ [45]. This may be expressed as

$$\Pr(\Delta|y) = \sum_{i=1}^{M} \Pr(\Delta|\mathcal{M}_i, y) \Pr(\mathcal{M}_i|y), \tag{2.39}$$

where M models are under consideration. Models with high probability receive more weight, while models with less probability are

discounted. We may also find the posterior expected value for Δ using a similar relation,

$$\mathbb{E}(\Delta|y) = \sum_{i=1}^{M} \mathbb{E}[\Delta|\mathcal{M}_i, y] \Pr(\mathcal{M}_i|y). \tag{2.40}$$

Both Equation (2.39) and (2.40) have the form of a weighted average over models, hence the term Bayesian model averaging. If our quantity of interest are the predicted values $\Delta = \mathbf{y}_*$, the predictions using Bayesian model averaging are denoted $\mathbf{y}_* = \sum_{i=1}^{M} \mathbf{y}_{*i} \Pr(\mathcal{M}_i|y)$, where $\mathbf{y}_{*i}$ are the fitted values under each model. If each model $\mathcal{M}_i$ has parameters θ_i, we write

$$\Pr(\mathcal{M}_i|X) \propto \Pr(\mathcal{M}_i) \Pr(X|\mathcal{M}_i) \tag{2.41}$$

$$\propto \Pr(\mathcal{M}_i) \int \Pr(X|\theta_i, \mathcal{M}_i) \Pr(\theta_i|\mathcal{M}_i) d\theta_i. \tag{2.42}$$

One may wish to specify priors $\Pr(\theta_i|\mathcal{M}_i)$ and compute the posterior probabilities numerically. However, this may not be worth the extra effort over computing a simpler Bayesian information criterion (BIC) approximation [45].

We may also wish to apply the same principle from a frequentist perspective. Given some fitted values $\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_M(x)$, under squared error loss we seek to calculate weights $\mathbf{w} = w_1, \ldots, w_M$ such that

$$\hat{w} = \underset{w}{\operatorname{argmin}} \mathbb{E}_{\mathcal{P}}[\mathbf{y} - \sum_{m=1}^{M} w_m \hat{f}_m(x)]^2, \tag{2.43}$$

where the data X are distributed according to $\mathcal{P}$. This is solved through the population linear regression of $\mathbf{y}$ on $\hat{F}(x)^\top \equiv [\hat{f}_1(x), \ldots, \hat{f}_M(x)]$:

$$\hat{w} = \mathbb{E}_{\mathcal{P}}[\hat{F}(x)\hat{F}(x)^\top]^{-1} \mathbb{E}_{\mathcal{P}}[\hat{F}(x)\mathbf{y}]. \tag{2.44}$$

It is also possible to implement other loss functions.

Since the full regression has smaller error than an individual model, combining models will not reduce predictive performance at population level [45], i. e.,

$$\mathbb{E}_{\mathcal{P}}[\mathbf{y} - \sum_{i=1}^{M} w_i \hat{f}_i(x)]^2 \leqslant \mathbb{E}_{\mathcal{P}}[\mathbf{y} - \hat{f}_i(x)]^2 \forall i. \tag{2.45}$$

Weights are chosen so that the expected loss is minimised.

Rather than optimise over the weights, it is possible to use the Akaike information criterion (AIC) as model weights [91]. To compute this, we derive weights, w_i, based on the equation

$$w_i(\text{AIC}) = \frac{\exp\left\{-\frac{1}{2}\Delta_i(\text{AIC})\right\}}{\sum_{k=1}^{K} \exp\left\{-\frac{1}{2}\Delta_k(\text{AIC})\right\}}, \tag{2.46}$$

where K models form the ensemble and $\Delta_i(\text{AIC})$ is the model AIC minus the minimum AIC of all models. These weights are then multiplied by the predictions from each model.

2.6 MODEL METRICS

The type of metric used to evaluate mode performance is determined by the nature of the response variable. Functions of residuals are commonly used in a regression setting, e. g., mean squared error (MSE), whereas classification performance factors the posterior class probabilities, e. g., score function [40]. The AIC may be used in both a regression and classification setting to determine the trade-off between fit and model complexity. The likelihood ratio test may also be used for both regression and classification problems for comparison of nested models. A nested model is a model whose parameters are a subset of the parameters within the competing model.

When assessing predictive performance on unobserved data, in both a classification and regression setting, one typically uses a train-test split. This is performed by randomly splitting the data into two sets, a common choice being $80:20$ split, called training and testing sets. The training set is used to learn the model parameters and the left out data are predicted from the trained model. The model which has the greatest accuracy is typically preferred, subject to time constraints.

In some cases, a validation set may be included, for instance in the train, test, validation ratio of $80:10:10$. The validation set is used when we have identified the best performing model on the test set and are interested in its performance on a separate dataset, i. e., the validation set. However, the performance on the validation set may, in fact, be worse than the performance on the test set.

2.6.1 *K-fold cross validation*

K-fold cross validation is a method to assess a model's predictive performance on untrained data, which simultaneously prevents overfitting in the process. Like the train-test split, K-fold cross validation involves partitioning the data into K roughly even folds. The data contained within a single fold is left out for prediction, whilst the data within the $K-1$ folds are used for model training. Choices of K vfary, with typical values being 5 or 10. Figure 2.12 illustrates the partition into 10 folds for cross validation.

2.6.2 *Regression model metrics*

To assess the performance of a regression model, there are several common metrics available at ones disposal. These include the mean squared error,

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

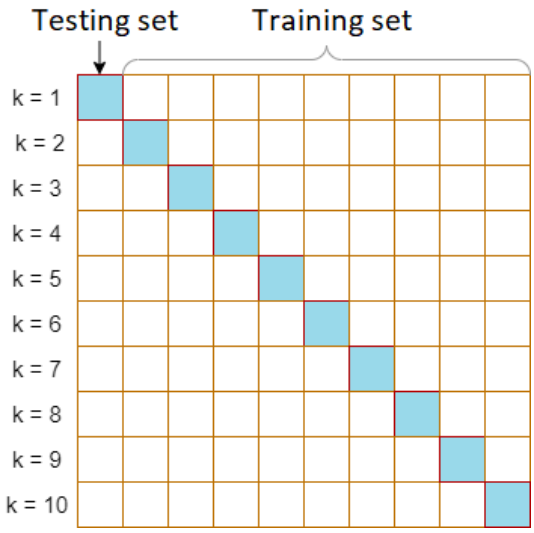


Figure 2.12: Data partitioned into 10 folds for cross validation.

and the mean absolute error

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

where n is the number of points in the test set, y_i is the observed value of the ith experiment and $\hat{y}_i$ is the fitted value of the ith experiment. A lower model error indicates greater predictive accuracy. The difference between these is referred to as the residual, given as $\epsilon_i = y_i - \hat{y}_i$, which mey either be positive or negative. The MSE operates by reflecting the residuals on the positive axis and squaring the distance. This has the effect of inflating larger differences, thereby penalising models with less accuracy. The mean absolute error (MAE) is similar metric which acts on the magnitude of the difference.

Another useful metric for regression modelling, which may be used for non-nested models, is the coefficient of determination, typically denoted,

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\bar{y}$ is the mean of the response variable. R^2 gives an idea of how correlated with predictors are with the response variable and may take negative values. The closer R^2 is to one, the more correlated the predictors are with the target variable. Typically, the adjusted coefficient of determination is used for model selection.

Another metric used with a regression setting is the score function, with the family of methods referred to as proper scoring rules [40]. The score function factors in the whole distribution, meaning predictions with low variance that are inaccurate will be penalised more than inaccurate predictions with high variance. The score function

essentially acts as a level of loss from each model. In a regression setting, a choice of score is the probability density function (PDF) of the log-normal distribution, denoted

$$\text{Score} = C - \frac{1}{2}\log \sigma^2 - \frac{(\mathbf{y} - \tilde{\mathbf{y}})^2}{2\sigma^2}, \tag{2.47}$$

where C are the constant terms in the log-normal PDF, $\tilde{\mathbf{y}}$ is the vector of the fitted value, and σ^2 is the variance of the fitted value. The scores of all data points are then averaged to obtain an average loss for each data point. The closer the score is to zero, the greater the model accuracy.

One may also consider the AIC for spatial models [48]. The idea behind AIC is to act as a trade-off between model fit, through the likelihood term, and model complexity, through a penalty parameter. In model selection, we choose a combination of regressors which minimises the AIC. The AIC is assessed when fitting the model to the entire dataset, and is denoted

$$\text{AIC} = -2\log L_z(\hat{\Psi}) + 2p, \tag{2.48}$$

where the model $\hat{\Psi}$ contains p explanatory variables and is evaluated at n data points.

2.6.3 *Classification model metrics*

The metrics for a classification model vary as we are now modelling the mean of the response and not the residuals. The metric varies according to the number of responses and are typically functions of predicted class probabilities.

2.6.3.1 *Binary response*

When the response is binary, there are several metrics at one's disposal, and are, typically, some construct of the confusion matrix. The confusion matrix is a visual way of assessing the fitted values compared with the observed values. There are three main constructs of the confusion matrix, namely

- **Sensitivity/ recall**: $\frac{TP}{TP+FN}$ $\left(\frac{\text{Number of true positive assessments}}{\text{Number of all positive assessments}}\right)$
- **Specificity**: $\frac{TN}{TN+FP}$ $\left(\frac{\text{Number of true negative assessments}}{\text{Number of all negative assessments}}\right)$
- **Accuracy**: $\frac{TN+TP}{TN+TP+FN+FP}$ $\left(\frac{\text{Number of correct assessments}}{\text{Number of all assessments}}\right)$

These metrics should be assessed amongst each other, as increasing one usually results in the decrease of another. Model performance may also be assessed visually through the receiver operating characteristic

receiver operating characteristic curve (ROC) curve. This method assess the true positive rate against the false positive rate when varying the threshold of classification, typically set to 0.5. This means that once transforming the linear predictor using, say, the *logit* transformation, any value above 0.5 is classed as a 1 and values below 0.5 are classed as a 0. The ROC curve identifies the different true positive to false positive rates while the area under the curve (AUC) identifies the optimal ROC curve which has the greatest area underneath.

There are two heuristic approaches to assessing multi-class classification performance whereby we reduce multiple classes to binary comparisons. This involves reducing the number of classes to binary sub-problems, then evaluating several models. The are two main approaches are:

One-vs-one Here we divide the K class problem into $\frac{K(K-1)}{2}$ binary classifiers. A classifier, $\hat{f}_{jk}$, is constructed by coding the j^{th} class as positive and the k^{th} class as negative, with $j, k = 1, \ldots, K, j \neq k$. For a new data point, x, we average the votes in each class and assign x to the class with the highest number of votes. This method is less sensitive to the problems of imbalanced datasets, however it has greater computational expense.

One-vs-all In this approach, we have to train a separate classifier for each different pair of labels. This approach involves dividing the K-binary classification problem into K-class sub-problems, comparing the k^{th} class with the not k^{th} class, for $k = 1, \ldots, K$. A classifier, $\hat{f}_k$, is then constructed so that the instances in the k^{th} class are positive and the union of all other classes are negative. A new data point, x, is then assigned to the class with the largest value of $\hat{f}_k(x)$, $k = 1, \ldots, K$, where $\hat{f}_k(x)$ is the optimal solution to the binary problem.

Setting up multiple binary classification problems may result in extra computational burden as well as having too many models to compare, depending on the number of classes in the response. It may, therefore, be preferable to evaluate performance from a single model

2.6.3.2 *Multinomial response*

There are several methods to assess multi-class classification performance without reducing the number of classes. AIC is one such method for non-nested models. Scoring rules are another popular choice.

In defining a scoring rule, suppose M models were fitted, each with different link or correlation functions, and estimated probabilities for input x $\hat{\boldsymbol{\pi}}_m(\mathbf{x})$, $m = 1, \ldots, M$ and corresponding data y_i, for $i = 1, \ldots, n$, the overall loss for the m^{th} model is

$$\sum_{i=1}^{n} \text{Loss}(\boldsymbol{\pi}^{(m)}, y_i),$$

where $\text{Loss}(\boldsymbol{\pi}, y_i)$ is the loss function for experiment i, and $\boldsymbol{\pi}$ is the probabilistic forecast of the observed class. There are several choices

of loss functions, such as the logarithmic,Brier, and spherical losses, defined as

$$\begin{aligned} \textit{log} :\text{Loss}(\boldsymbol{\pi}_i, y_i) &= \log \pi_{i,y_i}, \\ \textit{Brier} :\text{Loss}(\boldsymbol{\pi}, y_i) &= -2\pi_{i,y_i} + \sum_{j=1}^{C} \pi_{ij}^2 + 1, \\ \textit{Spherical} :\text{Loss}(\boldsymbol{\pi}, y_i) &= \frac{\pi_{i,y_i}}{||\pi_i^2||}, \end{aligned}$$

where y_i is the observed class of the i^{th} experiment. We seek the models with a lower loss as they tend to assign the highest probabilities to the correct classes.

3

GP MODEL FOR COMPOUND DATA ANALYSIS AND DISCOVERY

3.1 FINGERPRINTS AS A REPRESENTATION OF THE CHEMICAL SPACE

Fingerprints are a widely used concept for assessing molecular substructures and are represented as bit vectors based on their chemical graphs. Each feature within the fingerprint indicates the presence of some atomic substructure, e.g., a vertex or a cycle [52]. Most of the molecular fingerprints have been developed to describe molecular structures associated with biological activities based on synthetic compounds [80] .

The Tanimoto (Jaccard) similarity is a measure of closeness between fingerprints. Consider two vectors of the form $c_r = (c_{r1}, c_{r2}, \ldots, c_{r\kappa})$ where c_{ri} is either 0 or 1, and not all 0, denoting the presence of feature (atomic substructure) i in the rth compound, $i = 1, \ldots, \kappa$. The Tanimoto similarity $S_{rs} = S(c_r, c_s)$ for a pair of compounds c_r, c_s is defined to be the number of features in common between the two compounds over the number of features in either. More specifically,

$$S_{rs} = \frac{\langle c_r, c_s \rangle}{\langle c_r, c_r \rangle + \langle c_s, c_s \rangle - \langle c_r, c_s \rangle}, \tag{3.1}$$

where $\langle c_r, c_s \rangle = \sum_{i=1}^{\kappa} c_{ri} c_{si}$. By definition, $0 \leqslant S_{rs} \leqslant 1$. When the two compounds have no features in common, the Tanimoto similarity is zero, i. e., $S_{rs} = 0$, and when the compounds have identical features, the Tanimoto similarity is one, i. e., $S_{rs} = 1$. The $m \times m$ matrix with elements S_{rs}, $r, s = 1, \ldots, m$, is positive definite [11]. This allows us to apply the Tanimoto similarity as a valid covariance function, denoted Jaccard in Table 3.1.

Subtracting the Tanimoto similarity from one converts the similarity into a distance [41, 42], with the Tanimoto distance between compounds c_r and c_s denoted by

$$T(c_r, c_s) = T_{rs} = 1 - S_{rs}. \tag{3.2}$$

Some authors [31] used the Tanimoto distance directly within a Gaussian kernel to model the correlation of a Gaussian process. Although the Tanimoto distance is a metric, it is non-Euclidean, and can result to non-positive definite correlations when used with spatial kernels. This result allows us to create a vast catalogue of correlation functions based on the Tanimoto distance, inspired by the correlation functions used in the GP literature, which allow the GP model to have certain

Correlation	$R(t, \phi)$
independent	$\mathbb{1}(t = 0)$
exponential	$\exp\{-\sqrt{t}/\phi\}$
Gaussian	$\exp\{-t/\phi^2\}$
Jaccard	$1 - t$

Table 3.1: Correlation functions based on the Tanimoto metric at distance t with scaling parameter ϕ.

properties. Table 3.1 lists several choices of the GP correlation, $R(t, \phi)$, corresponding to compounds with Tanimoto distance t. The independent correlation corresponds to what is commonly referred to as the mixed effects model, and is used for reference to assess the improvement when incorporating correlation.

As an example, consider the chemical space $\mathcal{C} = \{c_1 = (0,1,1), c_2 = (1,0,1), c_3 = (1,1,0), c_4 = (1,1,1)\}$. The matrix of pairwise Tanimoto distances, T, and the corresponding correlation matrix R with elements $R_{rs} = \exp(-T_{rs}^2)$, are given by

$$T = \begin{pmatrix} 0 & 2/3 & 2/3 & 1/3 \\ & 0 & 2/3 & 1/3 \\ & & 0 & 1/3 \\ & & & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 1 & 0.6412 & 0.6412 & 0.8948 \\ & 1 & 0.6412 & 0.8948 \\ & & 1 & 0.8948 \\ & & & 1 \end{pmatrix}$$

Note that the distances given in T cannot correspond to distances in some Euclidean space. To see this, suppose there exist points $\varepsilon_1, \ldots, \varepsilon_4$ on some Euclidean space with pairwise distances given by T. Then, as $T_{14} + T_{24} = T_{12}$, $T_{14} + T_{34} = T_{13}$, and $T_{24} + T_{34} = T_{23}$, the point ε_4 must lie simultaneously in the middle of the edges of the equilateral triangle formed by ε_1, ε_2, and ε_3, which is impossible. Note also that the correlation matrix R is not positive definite as its lowest eigenvalue is about -0.036.

Next, we discuss the use of the Tanimoto distance with well-known spatial kernels.

Definition 1. Let $(\mathcal{C}, d)$ be a metric space. The metric d is called Euclidean if for any set of points $c_1, \ldots, c_m \in \mathcal{C}$, there exist $\varepsilon_1, \ldots, \varepsilon_m \in \mathbb{R}^\alpha$ (α depends on m), such that $d(c_r, c_s) = \|\varepsilon_r - \varepsilon_s\|$ for all $r, s = 1, \ldots, m$, where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^\alpha$. In this case, we say that the points $\{c_1, \ldots, c_m\}$ can be isometrically embedded in a Euclidean space of dimension α.

The following theorem, appearing in Gower [43], can be used to show that a metric is Euclidean. We denote the $m \times m$ identity matrix by I_m, and the $m \times m$ matrix of ones by J_m.

Theorem 1. *Let* $(\mathcal{C}, d)$ *be a metric space.*

1. *The metric* d *is Euclidean if and only if, for any set of points* $c_1, \ldots, c_m \in \mathcal{C}$*, the* $m \times m$ *matrix* $B = HAH$ *is positive semi-definite, where* $H = I_m - m^{-1}J_m$*, and* A *is the* $m \times m$ *matrix with elements* $A_{rs} = -d(c_r, c_s)^2/2$, $r, s = 1, \ldots, m$.

2. *Furthermore, let* $\alpha = \text{rank}(B)$*. Then, the points* $\{c_1, \ldots, c_m\}$ *can be isometrically embedded in a Euclidean space of dimension* α*, and* α *is the lowest dimension for which this is possible.*

Now consider the chemical space $\mathcal{C} = \{c_1, \ldots, c_m\}$ with the metric $d(c_r, c_s) = \sqrt{T(c_r, c_S)}$. The matrix B from Theorem 1 is $B = -\frac{1}{2}H(J_m - S)H = \frac{1}{2}HSH$, where S is the $m \times m$ matrix with elements given by (3.1). As S is positive definite, B is positive semi-definite and $\text{rank}(B) = m - 1$, therefore, the points $\mathcal{C}$ can be embedded in a $(m-1)$-dimensional Euclidean space. Mardia, Kent, and Bibby [60, Section 14.2.2] provide an algorithm for finding the points $\varepsilon_1, \ldots, \varepsilon_m$ in the Euclidean space. In the example given earlier, $\varepsilon_1 = (-1/\sqrt{6}, -1/\sqrt{18}, -1/12)$, $\varepsilon_2 = (1/\sqrt{6}, -1/\sqrt{18}, -1/12)$, $\varepsilon_3 = (0, 2/\sqrt{18}, -1/12)$, $\varepsilon_4 = (0, 0, 1/4)$ have pairwise Euclidean distances given by the square root of the elements of T.

3.2 SAMPLE VARIOGRAM AS A METHOD OF ASSESSING CORRELATION

The Tanimoto distance is a suitable metric if there exists spatial dependence. To confirm dependence, we may resort to the variogram, or the semi-variogram. For a stochastic process $f(x)$, a variogram is defined to be the variance of the difference between two field values (draws from the GP) at locations x_1 and x_2, defined by their fingerprints. The variogram is expressed as

$$\begin{aligned} 2\gamma(x_1, x_2) &= \text{Var}(f(x_1) - f(x_2)), \\ &= \mathbb{E}[(f(x_1) - f(x_2))^2] - \mathbb{E}[f(x_1) - f(x_2)]^2, \\ &= \mathbb{E}[(f(x_1) - f(x_2))^2]. \end{aligned} \tag{3.3}$$

The second term in (3.3) vanishes as we assume $\mathbb{E}[f(x)] = 0$ for a GP. To compute the the variogram, we first generate pairwise Tanimoto similarities between all compounds. We then create a matrix for the associated differences of residuals from a regression model of choice, one such option being the random effects model. We fit a suitable model and look at the residuals since the assumption $\mathbb{E}[f(x)] = 0$

might not hold for the actual process, but it holds for the residuals. We map

$$\begin{bmatrix} t(c_1,c_1) & t(c_1,c_2) & \dots & t(c_1,c_n) \\ t(c_2,c_1) & t(c_2,c_2) & \dots & t(c_2,c_n) \\ \vdots & \vdots & \ddots & \vdots \\ t(c_n,c_1) & t(c_n,c_2) & \dots & t(c_n,c_n) \end{bmatrix} \rightarrow \begin{bmatrix} r_1 - r_1 & r_1 - r_2 & \dots & r_1 - r_n \\ r_2 - r_1 & r_2 - r_2 & \dots & r_2 - r_n \\ \vdots & \vdots & \ddots & \vdots \\ r_n - r_1 & r_n - r_2 & \dots & r_n - r_n \end{bmatrix}$$

where r_i corresponds to the ith residual. The procedure is applied to all elements in the matrix distance h. The variogram is computed at various distances. This makes sense only for isotropic GP when the variogram is only a function of h. We then find all the associated differences in residuals that correspond to compounds of distance h. Because of the symmetry, we only need to observe either the upper or lower triangle. Once we have selected the residuals corresponding to distance h, we calculate the variance of these residuals using the formula

$$2\hat{\gamma}(h) = \frac{1}{n_h} \sum_{i,j} (r_i - r_j)^2 \tag{3.4}$$

where the sum is over all pairs (i,j) such that $t(c_i,c_j) = h$ and n_h corresponds to the number of pairwise distances equal to h.

We then select the number of bins used for the variogram, which contain all distances that lie within h. This will increase the sample size used for estimating $2\gamma(h)$ for different values of h. In practice, there are only few data with a given distance h, which makes the estimator unreliable. To improve the estimate, we use distances within $h - \epsilon, h + \epsilon$ for the calculation of the variogram at distance h. We proceed by plotting the variance of the residuals against the mid-point of each bin. This then creates points for the scatter plot, known as the sample variogram. Spatial dependence exists if the covariance will decrease as the distance increases, i. e., anything other than a straight line. A straight line indicates that no matter the distance of $x_i - x_j$, the covariance will be constant.

3.3 GPS FOR ORDINAL OUTCOMES

For the GP classification model, we consider a chemical space $\mathcal{C} = \{c_1, \dots, c_m\}$ of m distinct compounds. In practice, m is large, but only a small number of them will be used in experiments. We assume observed data $(\mathbf{x}_1, y_1, c_{l_1}), \dots, (\mathbf{x}_n, y_n, c_{l_n})$, where $y_i \in \{1, 2, \dots, C\}$, with $1 < 2 < \dots < C$, is the class response, $\mathbf{x}_i$ are the testing conditions, and $l_i \in \{1, \dots, m\}$ indicates which compound is used in the ith experiment among m distinct compounds in $\mathcal{C}$, $i = 1, \dots, n$. The objective is to predict the outcome y_* given experimental conditions

$\mathbf{x}_*$ with unobserved compound c_*, i. e., to estimate the probabilities $\Pr(y_* = j|\mathbf{y})$ for each class $j \in \{1, \ldots, C\}$, where $\mathbf{y} = \{y_1, \ldots, y_n\}$.

Let $T(\cdot, \cdot)$ be a distance in the chemical space. Define $u : \mathcal{C} \mapsto \mathbb{R}$ to be a GP on $\mathcal{C}$, such that for any finite collection of compounds $\mathbf{u} = (u(c_1), \ldots, u(c_m))$ is distributed according to the m-dimensional multivariate normal distribution with mean 0 and variance covariance matrix K. We write the (r, s)th element of the matrix K, $k_{r,s}$, $r, s = 1, \ldots, m$, corresponding to compounds c_r and c_s as $k_{rs} = \sigma^2 R(T(c_{l_r}, c_{l_s}), \phi)$, where $R(t, \phi)$ denotes the correlation function at distance t with scaling parameter ϕ, and σ^2 denotes the variance parameter.

Let y denote the outcome of an arbitrary experiment under conditions $\mathbf{x}$ with compound c, and let $\gamma_j = \Pr(y \leqslant j|u(c))$, with $\gamma_C = 1$. Our model assumes that

$$\begin{aligned} G(\gamma_j) &= \alpha_j + \boldsymbol{\beta}^\top \mathbf{x} + u(c), \; j = 1, \ldots, C-1, \\ &= \eta_{jc}, \end{aligned}$$

where $G : (0, 1) \mapsto \mathbb{R}$ is the link function, $\boldsymbol{\beta}$ denotes the regressor coefficients, and $\alpha_1, \ldots, \alpha_{C-1}$ the intercepts. Note that this model is an extension of the model presented in Section 2.4.

Let $\gamma_{ij} = \Pr(y_i \leqslant j|u(c_i))$, be the cumulative probabilities for up to class j, $j = 1, \ldots, C$, and $\pi_{i1} = \gamma_{i1}$, $\pi_{ij} = \gamma_{ij} - \gamma_{i,j-1}$, $j = 2, \ldots, C$ be the individual class probabilities. We assume that the distribution of each y_i is conditionally independent of $y_{i'}$ for $i' \neq i$ given $u(c_{l_i})$. Thus our model can be described by

$$\begin{aligned} y_i|u(c_{l_i}) &\overset{\text{ind}}{\sim} \text{Categorical}(\boldsymbol{\pi}_i), \; i = 1, \ldots, n, \\ \mathbf{u} &\sim N_m(0, K), \end{aligned} \tag{3.5}$$

where $\boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iC})$ and $\mathbf{u}$ is the value of the GP at the m distinct compounds.

Within our application, decreasing values of u results in the estimation of higher high-class probabilities when the link function is increasing. To demonstrate this, we consider the odds ratio $(1-\gamma_j)/\gamma_j$ for $j = 1, \ldots, C$ and its behaviour as a function of u. We can see that $(1-\gamma_j)/\gamma_j = 1/\gamma_j - 1 = 1/G^{-1}(\eta_{jc}) - 1$, where $\eta_{jc} = \alpha_j + \beta^\top \mathbf{x} + u(c)$. If G is increasing, then so is G^{-1}, so the odds ratio is a decreasing function of u.

3.3.1 *Estimation of model parameters*

Let $\theta = (\alpha_1, \ldots, \alpha_{C-1}, \boldsymbol{\beta}, \sigma^2, \phi)$ denote the model parameters. We use the symbol $f(\cdot)$ to represent the probability density/mass function

of the expression in the brackets. Given the model in (3.5), we have, excluding any factors that do not depend on θ or $\mathbf{u}$,

$$f(\mathbf{y}|\mathbf{u};\theta) \propto \prod_{i=1}^{n}\prod_{j=1}^{C} \pi_{ij}^{\mathbb{1}(y_i=j)}, \tag{3.6}$$

$$f(\mathbf{u};\theta) \propto |K|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{u}^\top K^{-1}\mathbf{u}\right). \tag{3.7}$$

Here, $\mathbb{1}(\cdot)$ denotes the indicator function. The likelihood, based on data $\mathbf{y}$, is then

$$L(\theta|\mathbf{y}) = f(\mathbf{y};\theta) = \int f(\mathbf{y}|\mathbf{u};\theta) f(\mathbf{u};\theta)\, d\mathbf{u}. \tag{3.8}$$

The integral in (3.8) does not have a closed-form solution, so obtaining the maximum likelihood estimates of θ by direct maximisation of the likelihood is not possible. To compute the likelihood, we apply Laplace approximation, a technique which enables approximations to integrals of the form $\int e^{-g(\mathbf{u})}\, d\mathbf{u}$. Letting

$$g(\mathbf{u}) = -\log[f(\mathbf{y}|\mathbf{u};\theta) f(\mathbf{u};\theta)],$$

where $\hat{\mathbf{u}}$ denotes the point at which the function $g(\mathbf{u})$ is minimised, and $\hat{H}$ denote the Hessian matrix of $g(\mathbf{u})$ at $\hat{\mathbf{u}}$, we may express the second order Taylor expansion of $g(\mathbf{u})$ as

$$g(\mathbf{u}) \approx g(\hat{\mathbf{u}}) + \frac{1}{2}(\mathbf{u}-\hat{\mathbf{u}})^\top \hat{H}(\mathbf{u}-\hat{\mathbf{u}}). \tag{3.9}$$

By substituting (3.9) into (3.8), we obtain the approximation to the likelihood

$$f(\mathbf{y};\theta) \propto e^{-g(\hat{\mathbf{u}})} |\hat{H}|^{-\frac{1}{2}}. \tag{3.10}$$

We obtain $\hat{\theta}$ by maximising (3.10) with respect to θ. Furthermore, recognising that the exponential of (3.9) is proportional to a multivariate normal density, leads to the approximation

$$\mathbf{u}|\mathbf{y} \sim N_m(\hat{\mathbf{u}}, \hat{H}^{-1}) \text{ approximately as } n \to \infty. \tag{3.11}$$

For this approximation to be valid, we must have $n \to \infty$ and that $g = O(n)$. Shun and McCullagh [82] commented on the error of the Laplace approximation when $m \to \infty$. For the error of the Laplace approximation to be small, we require $m/n \to 0$ as $n \to \infty$. In our application, m increases as the number of features in the fingerprint increases, so the number of experiments, n, must increase as well.

3.3.2 *Likelihood of GP classification model*

The logarithm of the probability mass function for $\mathbf{y}|\mathbf{u}$, from (3.6), is given by

$$\begin{aligned}\ell(\mathbf{y}|\mathbf{u};\theta) &= \sum_{i=1}^{n}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\log(\pi_{ij})\\ &= \sum_{i=1}^{n}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\log(\gamma_j-\gamma_{j-1})\\ &= \sum_{i=1}^{n}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\log(G(\eta_{i,j})-G(\eta_{i,j-1}))\end{aligned}$$

where $\eta_{i,j} = \alpha_j + \beta^\top \mathbf{x} + u(c_{l_i})$ and we define $\alpha_0 = -\infty$, $\gamma_0 = 0$. Therefore

$$\begin{aligned}\frac{\partial \ell}{\partial u(c)} &= \sum_{i=1}^{n}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\eta\mathbb{1}(c_{l_i}=c),\\ \frac{\partial^2 \ell}{\partial u(c)\partial u(c')} &= \sum_{i=1}^{n}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\left\{\eta'-\eta^2\right\}\mathbb{1}(c_{l_i}=c)\mathbb{1}(c_{l_i}=c'),\end{aligned}$$

where

$$\begin{aligned}\eta &= \frac{G'(\eta_{i,j})-G'(\eta_{i,j-1})}{G(\eta_{i,j})-G(\eta_{i,j-1})},\\ \eta' &= \frac{G''(\eta_{i,j})-G''(\eta_{i,j-1})}{G(\eta_{i,j})-G(\eta_{i,j-1})}.\end{aligned}$$

Overall, we can write

$$\frac{\partial \ell}{\partial \mathbf{u}} = P^\top \Psi_1, \qquad \frac{\partial^2 l}{\partial \mathbf{u}\partial \mathbf{u}^\top} = P^\top \Psi_2 P$$

where P is an $n \times m$ binary matrix where its i^{th} row is 0 everywhere except at l_i which equals 1, and Ψ_1 is an n-dimensional vector and Ψ_2 is an $n \times n$ diagonal matrix with elements

$$\begin{aligned}\Psi_{1i} &= \sum_{j=1}^{C}\mathbb{1}(y_i=j)\eta,\\ \Psi_{2ii} &= \sum_{j=1}^{C}\mathbb{1}(y_i=j)\left\{\eta'-\eta^2\right\},\end{aligned}$$

respectively, for $i = 1, \ldots, n$. To find $\hat{\mathbf{u}}$ used in the Laplace approximation, we solve

$$K^{-1}\hat{\mathbf{u}} - P^\top \hat{\Psi}_1 = 0, \tag{3.12}$$

and the Hessian is $\hat{H} = K^{-1} - P^\top \hat{\Psi}_2 P$, where $\hat{\Psi}_1$ and $\hat{\Psi}_2$ denote Ψ_1 and Ψ_2 evaluated at $\hat{\mathbf{u}}$.

3.3.3 *Prediction*

The approximation in (3.11) enables predictions of the class damage, y_*, for an unseen compound. We begin by evaluating the conditional distribution $\mathbf{u}_*|\mathbf{y}$ to estimate the unobserved effects of the GP. Using the conditional independence of $\mathbf{u}_*$ and $\mathbf{y}$ given $\mathbf{u}$, we observe that

$$f(\mathbf{u}_*|\mathbf{y}) = \int f(\mathbf{u}_*|\mathbf{u}) f(\mathbf{u}|\mathbf{y})\, d\mathbf{u} \approx \int f(\mathbf{u}_*|\mathbf{u}) \hat{f}(\mathbf{u}|\mathbf{y})\, d\mathbf{u} =: \hat{f}(\mathbf{u}_*|\mathbf{y}),$$

so the density $f(\mathbf{u}_*|\mathbf{y})$ can be approximated by a Gaussian density $\hat{f}(\mathbf{u}_*|\mathbf{y})$, whose mean and variance can be computed using the law of total expectation and variance. In doing so,

$$\begin{aligned}
\mathbb{E}[\mathbf{u}_*|\mathbf{y}] &= \mathbb{E}[\mathbb{E}[\mathbf{u}_*|\mathbf{u}]|\mathbf{y}] \\
&= \mathbb{E}[K_* K^{-1}\mathbf{u}|\mathbf{y}] \\
&= K_* K^{-1}\hat{\mathbf{u}} \qquad (3.13) \\
\mathrm{Var}[\mathbf{u}_*|\mathbf{y}] &= \mathbb{E}[\mathrm{Var}[\mathbf{u}_*|\mathbf{u}]|\mathbf{y}], + \mathrm{Var}[\mathbb{E}[\mathbf{u}_*|\mathbf{u}]|\mathbf{y}] \\
&= \mathbb{E}[K_{**} - K_*^\top K^{-1} K_*|\mathbf{y}] + K_*^\top K^{-1} \mathrm{Var}[\mathbf{u}|\mathbf{y}] K_* K^{-1} \\
&= K_{**} - K_*^\top K^{-1} K_* + K_*^\top K^{-1} \hat{H}^{-1} K_* K^{-1} \qquad (3.14)
\end{aligned}$$

where $K_* = \mathrm{Cov}(\mathbf{u}_*, \mathbf{u})$, and $K_{**} = \mathrm{Cov}(\mathbf{u}_*, \mathbf{u}_*)$. Here we have made use of the well known relations $\mathbb{E}[\mathbf{u}_*|\mathbf{u}] = K_* K^{-1}\mathbf{u}$ and $\mathrm{Var}[\mathbf{u}_*|\mathbf{u}] = K_{**} - K_*^\top K^{-1} K_*$ from Gaussian conditioning rules.

Let y_* denote the outcome of a future experiment under conditions $\mathbf{x}_*$ using compound c_*. To obtain the predicted outcome, we require the probabilities $\Pr(y_* = j|\mathbf{y})$ for $j = 1, \ldots, C$. This can be estimated as follows.

$$\begin{aligned}
\Pr(y_* = j|\mathbf{y}) &= E[\mathbb{1}(y_* = j)|\mathbf{y}] \\
&= E[E[\mathbb{1}(\mathbf{y}_* = j)|\mathbf{y}, \mathbf{u}_*]|\mathbf{y}] \\
&= E[E[\mathbb{1}(\mathbf{y}_* = j)|\mathbf{u}_*]|\mathbf{y}] \\
&= E[\pi_{*j}|\mathbf{y}] \\
&= \int \pi_{*j} f(u_*|\mathbf{y})\, du_* \\
&\approx \int \pi_{*j} \hat{f}(u_*|\mathbf{y})\, du_*. \qquad (3.15)
\end{aligned}$$

Equation (3.15) is evaluated using numerical integration. In this paper, we use the Gauss-Hermite quadrature method [30] with 21 integration points.

3.3.4 *Variance corrections to parameter uncertainty*

The formula for $\mathrm{Var}[\mathbf{u}_*|\mathbf{y}]$ given in the previous section is a function of the model parameters, θ. In practice, θ is unknown and is replaced by its estimate $\hat{\theta}$, effectively assuming that the true value of θ is $\hat{\theta}$.

This ignores the uncertainty in the value of θ. Booth and Hobert [10] provided a correction to the prediction variance for generalised linear mixed models with *independent* random effects. We follow a similar approach here.

Let $\mathbf{u}_*$ be the true value and let $\hat{\mathbf{u}}_*(\mathbf{y}, \theta) = \mathbb{E}[\mathbf{u}_*|\mathbf{y}]$ be the prediction with known θ. We want to assess the error $\hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \mathbf{u}_*$, where $\hat{\theta}$ denotes the maximum likelihood estimator for θ and $\mathcal{I}(\theta)$ is the corresponding Fisher information matrix.

We write $\hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \mathbf{u}_* = \hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \hat{\mathbf{u}}_*(\mathbf{y}, \theta) + \hat{\mathbf{u}}_*(\mathbf{y}, \theta) - \mathbf{u}_* = e_1 + e_2$, where $e_1 = \hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \hat{\mathbf{u}}_*(\mathbf{y}, \theta)$ is the additional error due to the uncertainty in θ and $e_2 = \hat{\mathbf{u}}_*(\mathbf{y}, \theta) - \mathbf{u}_*$ is the error had θ been known. Note that, e_1 is a function of $\mathbf{y}$, but not of u_*, and $\mathbb{E}[e_2|\mathbf{y}] = \hat{\mathbf{u}}_*(\mathbf{y}, \theta) - \mathbb{E}[\mathbf{u}_*|\mathbf{y}] = 0$. Then,

$$\begin{aligned}\mathbb{E}[e_1 e_2] &= \mathbb{E}[\mathbb{E}[e_1 e_2|\mathbf{y}]] \\ &= \mathbb{E}[e_1 \mathbb{E}[e_2|\mathbf{y}]] = 0.\end{aligned}$$

Furthermore,

$$\begin{aligned}e_1 &= \hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \hat{\mathbf{u}}_*(\mathbf{y}, \theta) \\ &\approx \nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta)^T (\hat{\theta} - \theta) \\ \Rightarrow \mathrm{Var}(e_1) &\approx \nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta)^T \mathcal{I}(\theta)^{-1} \nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta).\end{aligned}$$

Then,

$$\begin{aligned}\mathbb{E}[(\hat{\mathbf{u}}_*(\mathbf{y}, \hat{\theta}) - \mathbf{u}_*)^2] &= \mathbb{E}[(e_1 + e_2)^2] \\ &= \mathrm{Var}(e_1 + e_2) \\ &= \mathrm{Var}(e_1) + \mathrm{Var}(e_2) \\ &\approx \nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta)^T \mathcal{I}(\theta)^{-1} \nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta) + \mathrm{Var}[\mathbf{u}_*|\mathbf{y}].\end{aligned} \tag{3.16}$$

The second term in (3.16) is given by (3.14), while the first term is the variance correction due to estimation in θ. To compute the derivatives $\nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta)$, note that, by (3.13), $\nabla_\theta \hat{\mathbf{u}}_*(\mathbf{y}, \theta) = K_* K^{-1} \nabla_\theta \hat{\mathbf{u}}(\mathbf{y}, \theta)$, where $\hat{\mathbf{u}}(\mathbf{y}, \theta)$ is the solution to (3.12). By differentiating both sides of (3.12) with respect to elements of θ, we are able to compute $\nabla_\theta \hat{\mathbf{u}}(\mathbf{y}, \theta)$ algebraically.

3.3.5 *Estimating standard errors through bootstrapping*

Bootstrapping is a statistical procedure that resamples a single dataset to create many simulated samples. This process allows you to calculate standard errors, construct confidence intervals, and perform hypothesis testing for numerous types of sample statistics. In defining the bootstrap, suppose we have a data-generating mechanism $f(y|\theta)$ from which we can sample independent data, $y_1, \ldots, y_n \sim f(y|\theta_0)$, where θ_0 denotes the 'true' value of the parameter θ. We assume we know

the true value of the model $f(y|\theta)$, but we don't know the true value θ_0. Using the data, our goal is to estimate the true value of θ by fitting the model to the observed data. Let $\hat{\theta} = g(\mathbf{y})$ be the estimated value of θ based on data $\mathbf{y} = \{y_1, \ldots, y_n\}$. The notation $g(\mathbf{y})$ is used to indicate that $\hat{\theta}$ is a function of the observed data $\mathbf{y}$. Bootstrap helps us obtain measures of the bias and variability of the estimator $\hat{\theta}$. The idea of bootstrap estimator is to generate new data from a distribution close to $f(y|\theta)$, then fit the same model to the new data to obtain a new estimate of θ. Repeating this process a large number of times, say B, will provide us with a sample of size B from the distribution of $\hat{\theta}$ which we can use to compute the estimates. Suppose $\hat{f}(y|\mathbf{y})$ is a distribution close to $f(y|\theta_0)$ that can generate new data. We use a distribution based on the observed data $\mathbf{y}$ and not based on θ_0 as the latter is unknown, but the former is observed. We then generate $\mathbf{y}_b = y_{b1}, \ldots, y_{bn}$ as an independent sample from $\hat{f}(y|\mathbf{y})$, for $b = 1, \ldots, B$. Let $\hat{\theta}_b = g(\mathbf{y}_b)$ be the estimate of θ based on the sample $\mathbf{y}_b$. Then $\hat{\theta}_b, \ldots, \hat{\theta}_B$ is and independent sample from the distribution of $\hat{\theta}$.

Using $\hat{\theta}_b, \ldots, \hat{\theta}_B$ allows us to compute

1. the bias of $\hat{\theta}$ as $\text{bias}(\hat{\theta}) = \bar{\theta} - \hat{\theta}$,
2. the variance of $\hat{\theta}$ as $\text{Var}(\hat{\theta}) = (1/(B-1)) \sum_{b=1}^{B} (\hat{\theta}_b - \bar{\theta})^2$,

where $\bar{\theta} = (1/(B)) \sum_{b=1}^{B} (\hat{\theta}_b)$ is the average bootstrap estimate of θ. The above formulae can be used to provide the bias-corrected estimate of θ by $\hat{\theta} - \text{bias}(\hat{\theta}) = 2\hat{\theta} - \bar{\theta}$, and the standard error of $\hat{\theta}$ as $\sqrt{\text{Var}(\hat{\theta})}$. It remains to choose the distribution $f(y|\mathbf{y})$ from where the data are sampled. There are two ways to do that:

1. Non-parametric bootstrap: sample from the empirical distribution of the sample. This corresponds to sampling with replacement n observations among the observed data.
2. Parametric bootstrap: sample from $f(y|\hat{\theta})$

Application to regression

In regression, the data are $(x_1, y_1), \ldots, (x_n, y_n)$. If non-parametric bootstrap is used, the we are resampling each pair (x_i, y_i). For parametric bootstrap, we resample or choose arbitrary x values and then sample the y values from the fitted model.

Application to correlated data

The above methods assume that the observed data are independent and identically distributed. If the data are correlated, then non-parametric bootstrap is difficult to implement, but parametric bootstrap is applied the same way. Parametric bootstrap is comprised of the following steps.

1. Fit the model to the original data to obtain $\hat{\theta}$.
2. Generate B new data from the model using $\hat{\theta}$.
3. Fit the model to each of the new data to obtain $\hat{\theta}_b, \ldots, \hat{\theta}_B$.
4. Compute bootstrap bias and variance.

3.4 OPTIMISATION METHODS FOR DRUG DISCOVERY

Exploration the chemical space is vital when discovering new and effective compounds. Due to the space of all possible compounds being so large, it is impossible to assess all combinations of features to discover the ideal compound [8]. Therefore, we would like techniques that guide us on the interesting regions to explore, as . Optimisation methods allow exploration of such chemical spaces and are suitable for the discovery of new drugs. They are a tool used to identify the key features with the compounds, i. e., the region within the chemical space where the molecule possess herbicidal features. We require discrete optimisation algorithms since the chemical space is discrete. Our aim is to minimise $u(c)$ over $\mathcal{C}$, thereby finding the most active compound. Optimisation methods may also be used to propose hypothetical compounds of high efficacy. We present a few methods employed for these pursuits which allow wider regions of the objective functions to be explored, thereby discovering several optima.

3.4.1 *Simulated annealing*

Simulated annealing is an optimisation method which seeks to approximate a solution to a given objective function. It is employed when one is interested in a global solution, rather than a local solution. Newton and BFGS optimisation algorithms are only used when the search space is continuous and are not suitable for discrete problems such as optimising over the chemical space. Unlike the Newton method of optimisation where one perpetually moves in the direction of the steepest gradient, there are two kinds of acceptance of the proposed move, namely a regular acceptance if the solution is improved upon, and a probabilistic acceptance for when a non-improving solution is proposed. Enabling the search in a direction which does not lead to an improved solution allows for the greater regions to be explored.

In our application, simulated annealing no longer searches over a continuous space, since the features of a fingerprint are dichotomous. Instead, we move in a high-dimensional, discrete space by single steps. These steps equate to either adding or removing arbitrary substructures, regarded as a 'switch', within the compound. Every switch results in a novel compound from which we may estimate its effect. If the newly proposed compound's effect is lower, i. e., it has greater

predicted potency than the predeceasing compound, then our options are two-fold: we remain in the current state, i. e., the same location of the chemical space and randomly select another feature to switch, or, with some probability, we accept the newly proposed compound and move to this location within the chemical space, see algorithm 1 for the Pseudocode. The effect of accepting a new compound with less predicted activity allows for broader exploration of the chemical space.

Given a candidate solution c', the probability of moving to c' is given by

$$P(\text{Accept } c') = \min\left(1; \exp\left(-\frac{\Delta E}{T_i}\right)\right)$$

where $\Delta E = u_0 - u'$ is the difference in the effect of the current solution u_0 and the effect of the newly proposed solution u'. The parameter T_i represents the current temperature and, in our application, takes the form

$$T_i = \frac{\lambda}{\log(\lfloor \frac{i-1}{\tau} \rfloor + e)}. \tag{3.17}$$

where λ is the maximum temperature, τ is the number of iterations until the temperature changes, and i is ith iteration of the optimisation. The operator $\lfloor \cdot \rfloor$ is the floor function.

When optimising using SA, the parameters of the temperature, T_i, should be chosen in such a way that allow you to obtain a range of probabilities. The value of ΔE should be taken into account for this. If λ is too high compared to ΔE, then the acceptance probability will be very close to 1, so movements are simply random. If λ is too small compared to ΔE, then the acceptance probability will be very close to 0, so movements will occur when a better compound is found, thus being unable to explore the chemical space sufficiently.

In choosing λ, one may take all fingerprints used to fit the data and change one feature from each, say the first active feature. This gives an idea on the different values of ΔE that are encountered. One may proceed by setting λ to some value close to the maximum of these ΔE values. Then select λ so that towards the end, the acceptance probability is lower than 0.1, for example. Algorithm 1

3.4.2 *Genetic algorithm*

A GA is a stochastic optimisation technique which follow the "Darwinian" model of natural selection, and may be applied to both continuous and dichotomous data. GAs act as a feature selection tool by choosing the optimal combination of features based on the strength of the parent features. The GA is an adaptive strategy and a global optimization technique. As an evolutionary algorithm, it belongs to

Algorithm 1 Pseudocode for Simulated Annealing

Input: Fingerprintlength, $iterations_{max}$, $temp_{max}$, Model
Output: S_{best}
$C_0 \leftarrow$ `CreateInitialSolution`(Fingerprintlength)
$C_{best} \leftarrow C_0$
for $i = 1$ **to** $iterations_{max}$ **do**
  $C' \leftarrow C_0$
  $u' =$ `Predict`(`Model`, C')
  if $u' \leqslant u_0$ **then**
    $C_0 \leftarrow C'$
    if $u' \leqslant u_{best}$ **then**
      $C_{best} \leftarrow C_i$
    end if
  else if `Exp` $\left(\frac{u_0 - u'}{T_i}\right) <$ `Rand`() **then**
    $C_0 \leftarrow C_i$
  end if
end for
return C_{best}

the broader study of Evolutionary Computation. GA are capable of solving for a near-optimal solution for multivariable functions without the mathematical requirements of strict continuity, differentiability, convexity and other properties [17].

The main tuning parameters within thw GA are the crossover rate and the mutation rate, which play a central role in diversifying individuals and exploring the search space to discover new solutions s [39, 94]. See Figure 3.1 for an illustration of the crossover and mutation roles and algorithm 2 for the Pseudocode.

3.5 SIMULATION STUDY

We perform a simulation study to examine whether the method of scoring identifies the true model from which the data are sampled. If the scoring identifies the true model, then we may assume it is a suitable method for model identification using our novel approach.

The chemical space is formed by combining 5 features, producing a total of $2^5 - 1 = 31$ distinct compounds, excluding the vector where no feature is present. The data consist of $n = 310$ realisations, where each of the 31 compounds was tested under 10 different experimental conditions. Let $y_{ik} \in 1, 2, 3$, $i = 1, \ldots, 10$, $k = 1, \ldots, 31$, denote the observed outcome at the ith experiment with compound k, which can be among $C = 3$ categories. The model for the cumulative probabilities is

$$\operatorname{logit} \Pr(y_{ik} \leqslant j) = \alpha_j + \beta x_i + u_k, \; j = 1, 2,$$

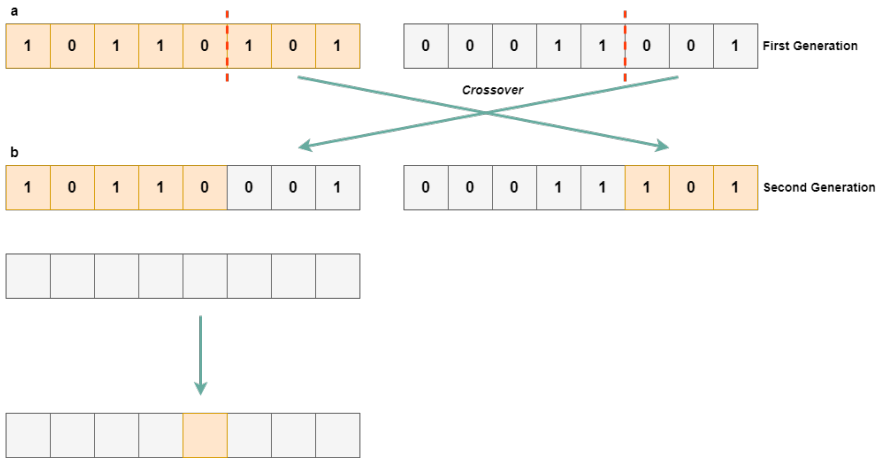


Figure 3.1: Methods to induce diversity in the population of individuals (candidate solutions). (a) During crossover, a feature within the fingerprint is exchanged by another feature of another compound. (b) When mutating, one or more features within a fingerprint are converted to a different one. This results in the survival of the fittest features.

Algorithm 2 Pseudocode for Genetic Algorithm

Input: $\mathrm{Population}_{size}, \mathrm{Problem}_{size}, \mathrm{P}_{crossover}, \mathrm{P}_{mutation}$
Output: S_{best}
Population $\leftarrow$ `InitialisePopulation`$(\mathrm{Population}_{size}, \mathrm{Problem}_{size})$
`EvaluatePopulation`(Population)
$\mathrm{S}_{best} \leftarrow$ `GetBestSolution`(Population)
while `StopCondition`() **do**
  Parents $\leftarrow$ `SelectParents`$(\mathrm{Population}, \mathrm{Population}_{size})$
  Offspring $\leftarrow \emptyset$
  for each $\mathrm{Parent}_1, \mathrm{Parent}_2 \in$ Parents **do**
    $\mathrm{Offspring}_1, \mathrm{Offspring}_2 \leftarrow$ `Crossover`$(\mathrm{Parent}_1, \mathrm{Parent}_2, \mathrm{P}_{crossover})$
    Offspring $\leftarrow$ `Mutate`$(\mathrm{Offspring}_1, \mathrm{P}_{mutation})$
    Offspring $\leftarrow$ `Mutate`$(\mathrm{Offspring}_2, \mathrm{P}_{mutation})$
  end for
  `EvaluatePopulation`(Offspring)
  $\mathrm{S}_{best} \leftarrow$ `GetBestSolution`(Offspring)
  Population $\leftarrow$ `Replace`(Population, Offspring)
end while
return S_{best}

	α_1	α_2	β	$\hat{\sigma}^2$
True	-1	0	1	0.1
Estimate	-0.9963	-0.0004	0.9965	0.0725
St dev	0.3028	0.2931	0.3889	0.1262

Table 3.2: True and average estimated parameter values and their standard deviations for the simulation study.

with a single covariate $x_i = (i-1)/9$. We choose $\alpha_1 = -1$, $\alpha_2 = 0$, and $\beta = 1$. The GP model for $\mathbf{u}$ consists of the Tanimoto covariance, with variance parameter $\sigma^2 = 0.1$.

We conduct 500 simulations in total. The parameter estimates for the logit-Tanimoto model, along with their standard deviations, were averaged across the 500 simulations and are presented in Table 3.2. We observe in Table 3.2 the parameters of the simulated data lie particularly close to the true estimates.

We fit 17 different models to each realised dataset. The different models consists of different choices of link and correlation function, as shown in Table 3.3, plus a random forest model. For computing the loss of each of these models, we partition the experiments into training and test sets in the ratio 80 : 20. We train the models on 80% of the simulated data and test the models' predictive accuracy on the remaining 20% through the method of scoring, using either the spherical score or the log score. The score of each model was averaged across the 500 simulations. Table 3.3 illustrates the classification performances of the 16 GP models as well as a random forest model. We see the model achieving the greatest classification accuracy is the GP model with logit link and Tanimoto covariance, that being the model from which the data were sampled. This demonstrates that scoring rules identifies the true model, thereby affirming its suitability for model selection.

Link	Correlation	Spherical	Log
logit	Tanimoto	−0.6007	1.0592
probit	Tanimoto	−0.6006	1.0593
log-log	Tanimoto	−0.6003	1.0601
probit	Gaussian	−0.6000	1.0606
logit	Gaussian	−0.6000	1.0606
logit	independent	−0.5999	1.0607
logit	exponential	−0.5999	1.0608
probit	exponential	−0.5998	1.0609
probit	independent	−0.5998	1.0610
C-log-log	Tanimoto	−0.6000	1.0612
log-log	Gaussian	−0.5997	1.0615
log-log	exponential	−0.5995	1.0616
log-log	independent	−0.5995	1.0617
C-log-log	Gaussian	−0.5990	1.0634
C-log-log	exponential	−0.5989	1.0637
C-log-log	independent	−0.5987	1.0642
RF		−0.5120	

Table 3.3: Average spherical and logarithmic loss for each model on the simulated data, ordered from highest to lowest in terms of their accuracy.

4

APPLICATION TO SYNGENTA'S DATA

4.1 DESCRIPTION AND PRESENTATION OF DATA

4.1.1 *Syngenta's testing process*

Every year, thousands of compounds undergo testing at Jealott's Hill Research Centre, to discover new and effective herbicides. Every compound is subjected to numerous stages of approval, namely screen tests, where compounds where compounds are tested for herbicidal properties. The most successful compounds will progress onto field trials for agricultural testing. Examples of herbicides produced by Syngenta include *Acuron*, *Axial*, *Gramoxone* and *Touchdown*, all of which have different means of addressing crop protection.

Syngenta externally source compounds for experimental research. Groups of experiments are stored into what are referred to as 'Projects'. Projects contain compounds posessing similar biological activity, i. e., share a common mode/method of action. For the purposes of this thesis, we will conduct our analysis using compounds from a single project.

Herbicidal performance is influenced by a number of factors including temperature, humidity, species, and developmental stage of the weed [50]. Important considerations when producing herbicides include efficacy and bystander safety. In light of this, it may be preferable to apply a lower dose, as this also tends to reduce the level of herbicidal resistance [44]. When determining the optimal dose of a herbicide, factors such as the crops' yield potential, the price of output, and the rotation practised are considered [38]. Before progressing onto field trials, herbicides typically undergo numerous stages of testing. In this project, we are analysing data from the initial stages, that is EPS. During EPS and PPS, compounds are tested on a mixture of weeds and crops contained in a trough with an untreated row left as the control group. Each trough is labelled with a compound descriptor and a date of application, with applications most commonly occurring at $250\,\mathrm{g\,ha^{-1}}$, $500\,\mathrm{g\,ha^{-1}}$, or $1\,\mathrm{kg\,ha^{-1}}$ (one hectare equates to $100\,\mathrm{m} \times 100\,\mathrm{m}$). Tests are performed either prior to the plant emerging (pre-emergence) or once the plant is partially or fully established (post-emergence).

Within two weeks of application, a trained biologist visually assesses the damage and ranks the herbicidal effect accordingly. Damages are recorded as percentages in multiples of 10. A score of 0 % indicates no herbicidal effect, whilst a score of 100 % indicates complete necrosis of

the plant. Damage, however, presents itself in multiple ways, such as a reduced pigmentation, or stunted growth. This also leads to variability in the biologists opinion of levels of damage. The pigmentation of the leaves is compared with colour codes to assist the biologists' assessments.

The following provides a summary of EPS and PPS trials:

1. **EPS trial**: EPS is preliminary stage of testing where around 8000 compounds are tested yearly to distinguish between inactive and active herbicides. Compounds are tested on various weeds and crops in climate controlled glass houses with a constant temperature of 24 °C and regular overhead irrigation. Compounds inflicting high levels of damage on the plant will progress to PPS. However, herbicides showing high levels of damage may not progress for other reasons, such as notable levels of toxicity. Within the glasshouse data, roughly 48 % of the compounds tested in EPS are present in PPS.

2. **PPS trial**: compounds demonstrating herbicidal potential during EPS will undergo further testing on more established weeds and crops and are subjected to a more rigorous set of criteria. The objective in PPS is to determine overall effectiveness and consistency in damage and to identify other interesting properties, such as selectivity, i. e., whether damage is inflicted on the weeds alone or acts on all crops. During PPS, we expect the average performance of compounds to be greater than in EPS. However, crops that are resilient to certain herbicides are subject to testing in PPS to further distinguish toxicity.

The data from PPS occurs from 1997, which is much earlier than 2007, the first year we have data for EPS. 35470 tests occurred during EPS and 238396 during PPS. There are also a vast number of compounds (745) being tested on few species (15) in EPS, which signifies the discovery aspect of the EPS stage. During PPS, we see less compounds are tested (361) which supports the claim that more potent compounds are tested during PPS. It is important to note that there are 10 years worth of data for EPS compared with 21 years for PPS. Interestingly, 98 % of the EPS experiments were conducted on weeds and not crops, as opposed to 77 % during PPS. This suggests EPS assess herbicidal vigour rather than selectivity.

Figure 4.1 shows there are two main periods of testing for PPS for the given project, one around 2003 and the other around 2013. Most of the EPS tests in our data occur around 2012 whereas in PPS is around 2013. It is important to note the times of the tests are for the first batch of data received and may not be reflective of the total number of experiments conducted during this time.

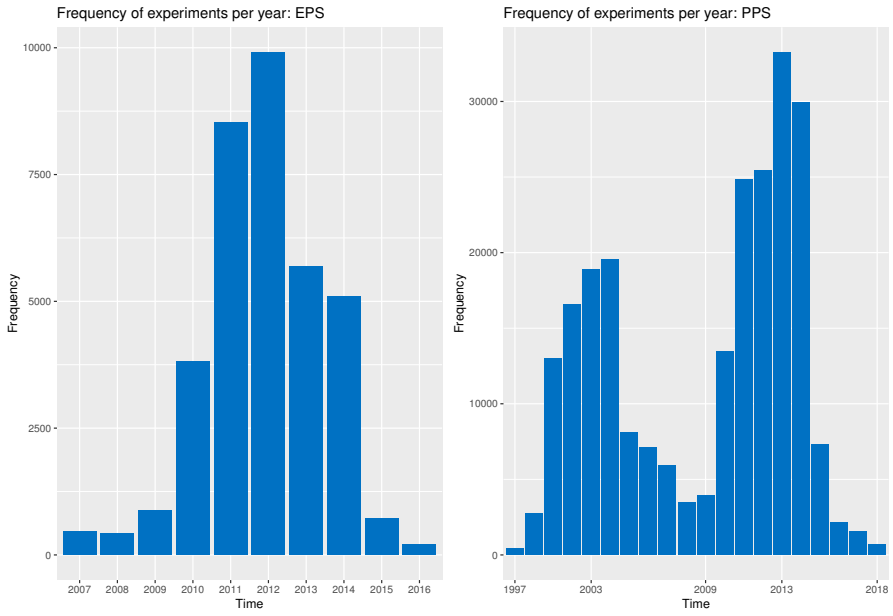


Figure 4.1: Histogram depicting the number of herbicidal tests per year for EPS and PPS.

	EPS:Pre	EPS:Post	PPS:Pre	PPS:Post
Species tested	13	15	43	43
Compounds tested	745	745	310	361
Distinct tests	193	193	406	407
Total experiments	17811	17929	107470	130926

Table 4.1: Summary statistics of the herbicide data by stage and method of application.

4.1.2 *Tanimoto similarities*

We now illustrate the distribution of the normalised Tanimoto similarities for the compounds in EPS and PPS. We observe in Figure 4.3 there is an average Tanimoto similarity of $\mu \approx 0.35$ with slightly less variation for the experiments in PPS

4.1.3 *Frequency of experiments*

In this section, we present summary statistics of the frequency of experiments for EPS and PPS data.

Table 4.1 shows 15 species were tested for EPS post-emergence and 13 for EPS pre-emergence. In PPS, 43 species were subject to testing for both pre- and post-emergence Out of the 35,740 tests in EPS, 193 were conducted during distinct times. In PPS, of the 238,396 tests, there were 410 conducted at a distinct time. Table 4.1 also shows that most of the compounds were applied during post-emergence, perhaps

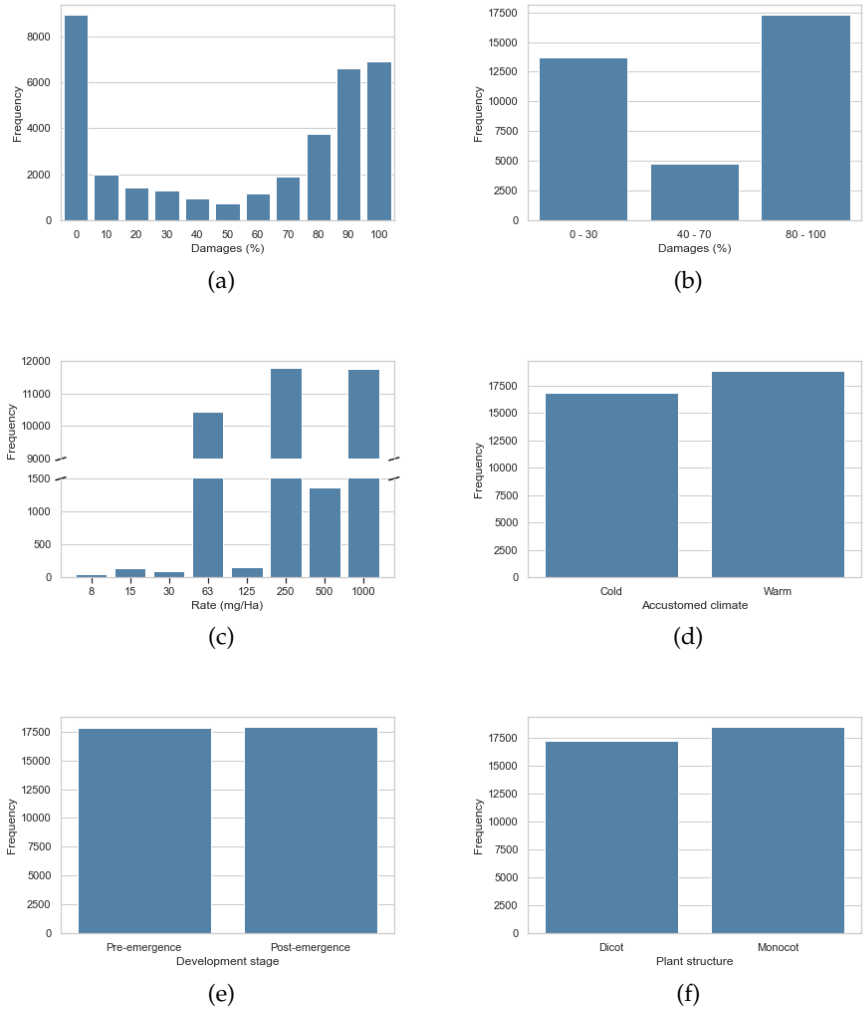


Figure 4.2: Distribution of raw damages, transformed damages, the rate of application of the herbicide, cold and warm acclimatised plants, pre- and post-emergence plants, and dicot and monocot plants for the EPS data

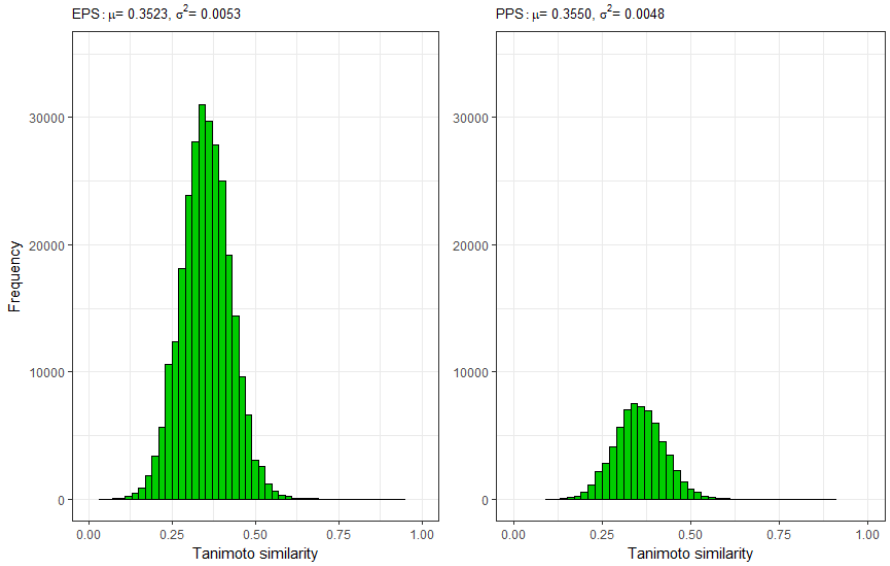


Figure 4.3: Tanimoto similarity for compounds tested within EPS and PPS

reflecting the greater demand to eliminate established weeds in an agricultural setting.

4.1.4 *Dose-response*

Some compounds had been tested substantially more than others. For example, the leading compound was applied in 142,480 tests, indicating this compound may be the commercial standard. Compounds are commonly tested against the commercial standard to assess relative potency. Figure 4.4 illustrates the dose-response effect of the eading compound for both pre- and post-emergence.

We selected species 37 and 15 for testing since these appeared most frequently in testing. Figure 4.4 illustrates a generalised additive model (GAM), [93], fitted to the data within the R package `ggplot2` [92]. The points indicate the damages at given rates, with a single data point representing multiple points. We see in Figure 4.4 for fixed species and methods of application, we see a positive trend between rate of application and damage for both the pre- and post-emergence plants. These plots suggest the herbicide has a stronger effect when applied during pre-emergence for both species. We also note the rate of damage increase for pre-emergence compared with post-emergence. There appears to be a steady increase for post-emergence, with a slight fall in damage as we reach the maximum rate of application. This suggests that pre-emergence plants reach the highest level of damage sooner, again indicating their susceptibility to the herbicide.

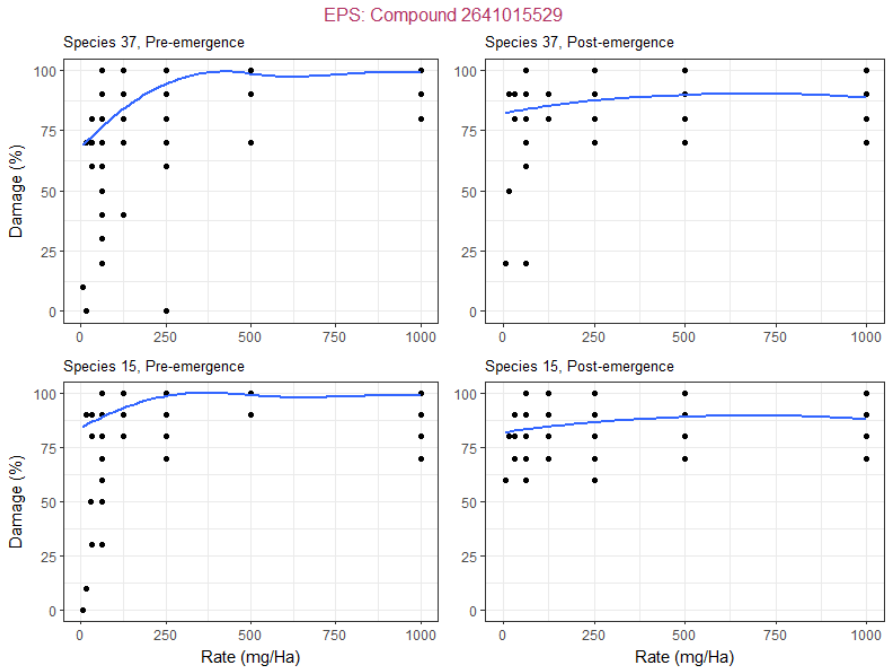


Figure 4.4: The effect of compound C_* on species 37 and 15, with applications occurring during pre- and post-emergence. The blue line represents a generalised additive model

4.1.5 *Species effect on Damage*

In Figure 4.7a, we have plotted the observed damages for selected species as this may provide insight to their susceptibility. We have divided the data into pre- and post- emergence for both EPS and PPS and all compounds are included in assessing the effect of herbicide on the species. These particular species have been selected since they were most frequently used in testing. Other species may show a different distribution of damages.

For the sake of analysis, understanding the susceptibility of each plant will help in the predictive phase, since this may be factored in when theoretically testing a compound on one of the species. We may also test a single compound on all species and aggregate the effects accordingly.

Figure 4.7a shows that Species 3, which is a monocot weed suited to a cold environment, has had many damages recorded as 0 % and no damages at 100 %. From this we may suspect that Species 3 is more resilient to the herbicidal effects than other varieties. On the other hand, Species 1, which is a dicot weed suited to a warm environment, has had many damages recorded above 80 % and few damages at 0 %. From this we infer species 3 is a less resilient variety of weed. These results motivate further analysis on causation of susceptibility, since being in a warm environment, or being a dicot may result in an increased susceptibility to herbicides.

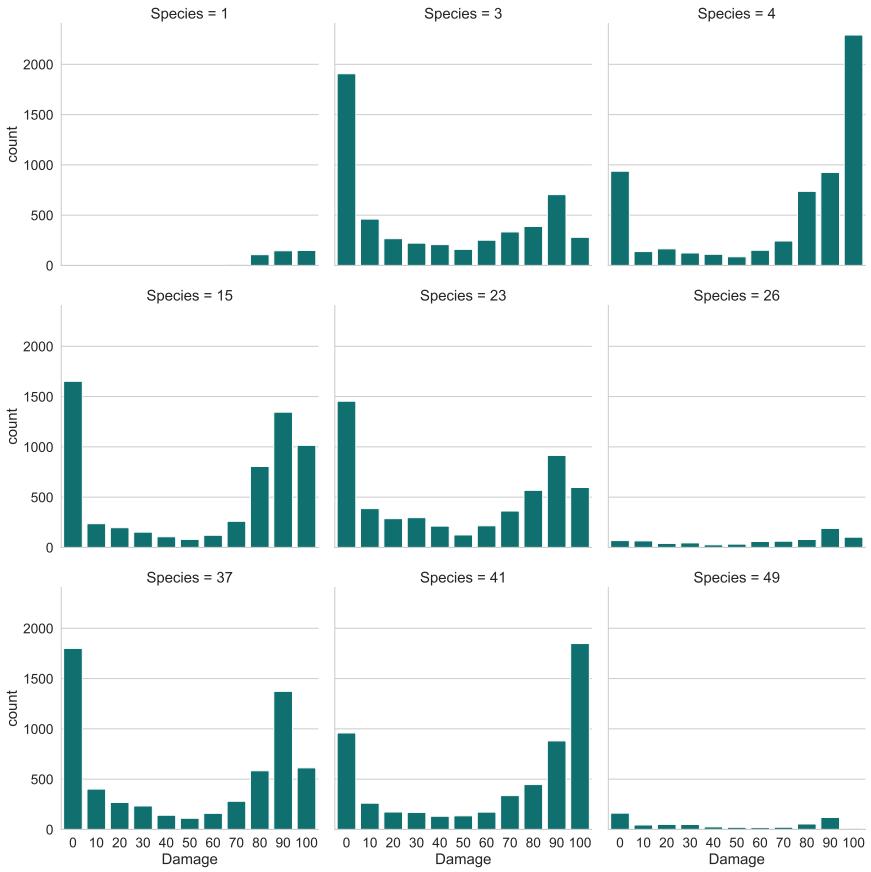


Figure 4.5: Distribution of damage rates by species. Note that species 5, 12, 14, 16, 28, and 45 were excluded as combined they only appeared 49 times.

An approach to modelling the responses may be to subdivide the damages into three classes, red, amber and green, red being compounds scoring a low damage, and green being compounds scoring a high damage. Since we are only interested in the compounds which are either inactive or successful, we may wish to only consider the compounds in the red and green zone and discard compounds in the amber category. This also reduces the dimensionality of the problem to a binary response and will result in greater computational efficiency. The colour coded system was a suggestion made by the computational chemists at Syngenta.

Knowing the impact of species properties to the damage is also of interest, since the type of plant and its natural climate may have an impact on the herbicidal effect. We have therefore decided to show the change in damage as we distinguish between the species properties.

Figure 4.6 shows bar charts for the frequency of damages for the species information. We notice that many more experiments were conducted on weeds than crops. This may be partly due to prioritising the development of non-selective herbicides. We note that for both weeds and crops, many of the damages were recorded at zero, which may indicate either the resilience of the particular plant or the lack of potency in the herbicide. What is evident between the two graphs is a very similar distribution exists for the damages. This may suggest the similarity of herbicidal effect on both weeds and non-weeds. In total, there were $54,720$ experiments on crops and $219,416$ on weeds.

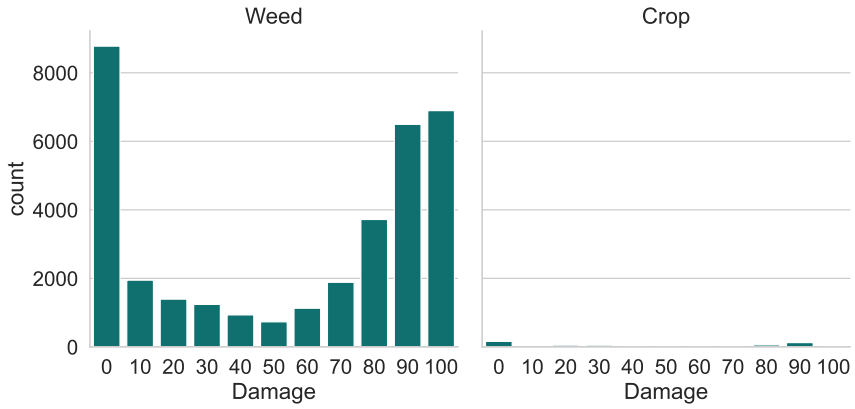
In contrast, the effect of herbicides on monocots and dicots varies quite noticeably. Based on the data, it appears there is a greater susceptibility to the herbicides for dicot species than for monocots. This is shown by the increased number of zeros for monocot and also the increase in the number of high damages for dicot. There were a total of $140,220$ experiments for monocot plants and $133,916$ for dicot plants.

In assessing the herbicidal effect on cold and warm plants, Figure 4.6 indicates plants in a warm environment have an increased sensitivity to the herbicidal effect. This is shown by the greater number of damages recorded at $80\,\%$ and above. There are a total of $107,251$ experiments for cold climate plants and $166,885$ for warm climate plants. We now wish to take this analysis further by studying the histograms for all the possible combinations of the species characteristics.

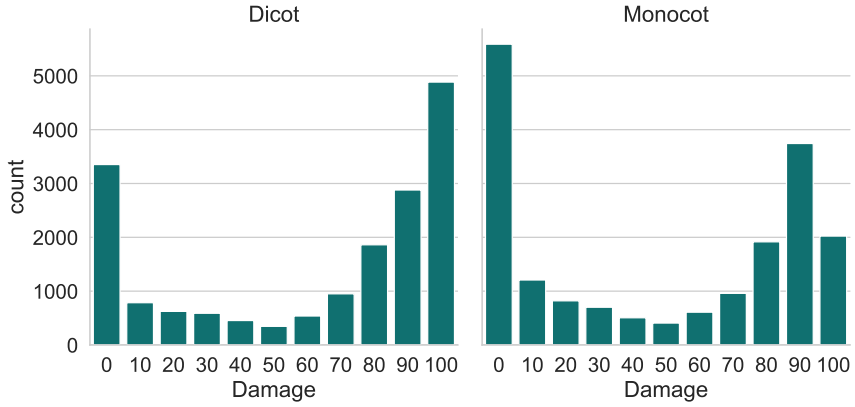
Figure 4.7 illustrates the effect of the different combinations of species characteristics to the damage. Monocot plants in a warm environment appear to be more sensitive to herbicidal effects than in a cold environment.

4.2 COMPOUND EFFECT ON DAMAGE

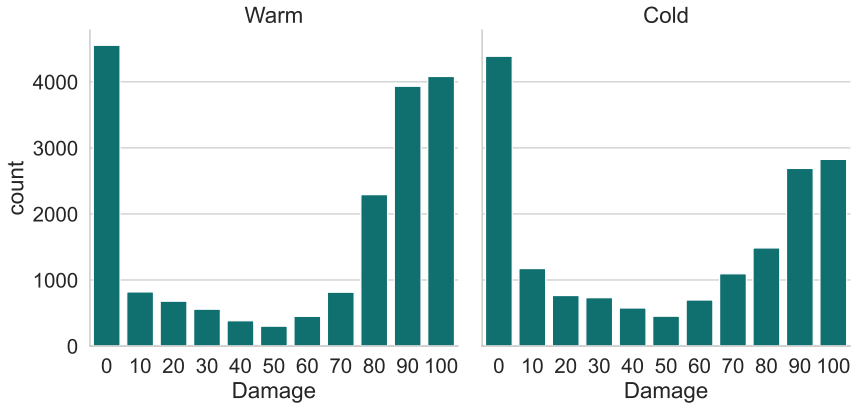
We now observe the compound effect on the damage, whilst including all species and rates of application in the analysis. We do this to gain



(a)

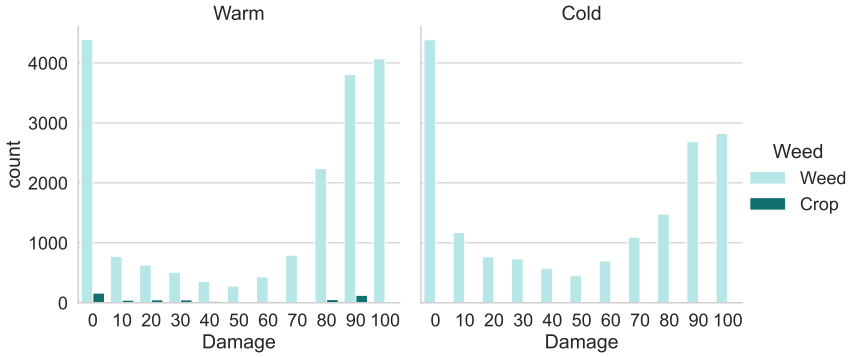


(b)

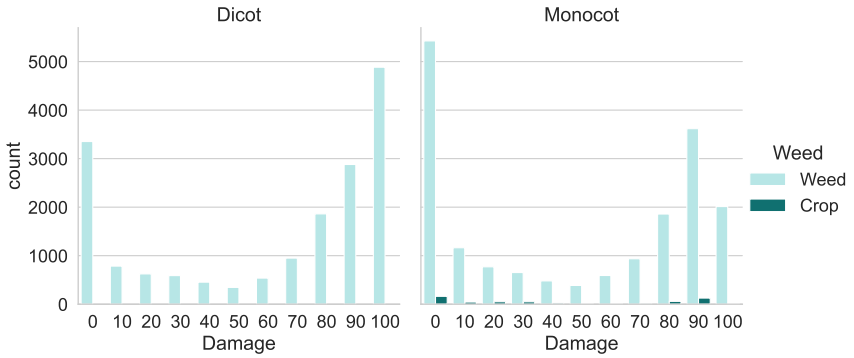


(c)

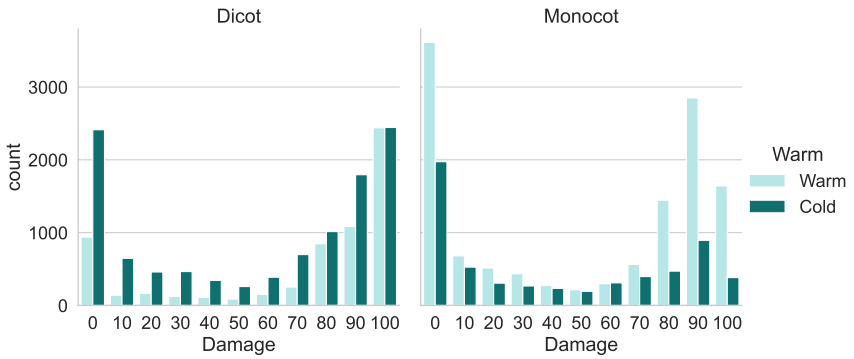
Figure 4.6: Panels (a), (b), and (c) show damage rates for different species characteristics weed, monocot, and warm respectively.



(a)



(b)



(c)

Figure 4.7: Damage rates for different combinations of species characteristics.

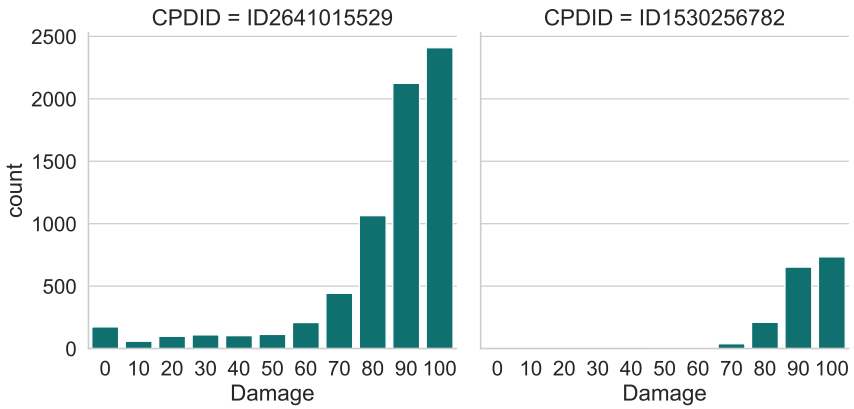


Figure 4.8: The frequency distribution of the observed damages for the two most tested compounds in the EPS data.

insight into the potency of each compound. We illustrate the effect of the compounds through histograms in Figure 4.8. These compounds have been selected since they were most frequently tested. As with the analysis of species, we divide the groups into pre- and post-emergence for both EPS and PPS. Taking compound 2641015529 for example, we see that many of the recorded damages are greater than 80 %, which illustrates the potency of this compound. This compound was however most frequently tested in the four groups, suggesting this may be the commercial standard. Compound 2528308950 on the other hand appears to be lacking in herbicidal vigour, with only around 350 tests being scored at 100 % and around 1,000 being scored at 0 %. With other compounds, it appears they are either mostly successful or quite the opposite, suggesting the experimental nature of herbicide production.

4.2.1 *Dicot vs monocot susceptibility*

We wish to explore the effect of the species type, either monocot or dicot, on the damage of the plant. In analysing all the data from the glass-house experiments, Figure 4.9 illustrates that higher damages are recorded for dicot species (red) than for monocots (blue). This result aligns with prior belief that dicots are more susceptible to the effect of herbicides. Table 4.2 shows the average damage for monocot is consistently less than for dicot, thus supporting the statement that dicots are presumed to be more susceptible to herbicides.

4.2.2 *Toxicity of compounds*

The compounds in our dataset are applied at rates between $1\,\mathrm{mg\,ha^{-1}}$ and $1000\,\mathrm{mg\,ha^{-1}}$. The maximum damage inflicted on a plant where

Figure 4.9: Damages for dicot and monocot species. The horizontal axis are indices of experiments

Rate ($mg\,ha^{-1}$)	62.5	125	250	500	1000
Monocot	35	41	49	53	59
Dicot	53	67	72	75	72

Table 4.2: Average damage for dicot and monocot species for commonly applied rates

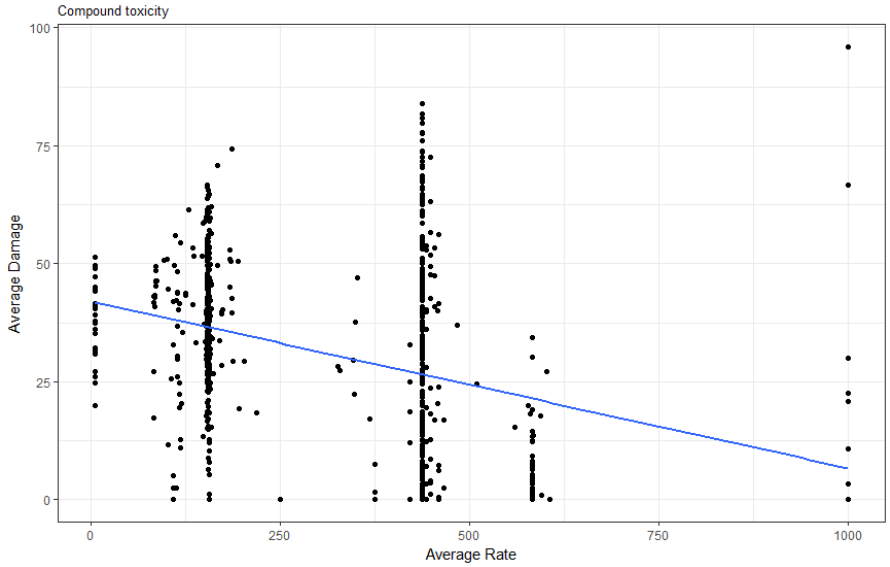


Figure 4.10: Average damage of compound for increasing average rate.

$1\,\mathrm{mg\,ha^{-1}}$ of compound is applied is 100 %. This was achieved by compound 2302716872. This is the only compound to score 100 % damage when applied at $1\,\mathrm{mg\,ha^{-1}}$. There are 11 distinct compounds which scored 90 % damage when applied at $1\,\mathrm{mg\,ha^{-1}}$.

On the other hand, the least damage inflicted on a plant when a compound is applied at $1000\,\mathrm{mg\,ha^{-1}}$ is 0 %. In total, 356 compounds have shown this lack of herbicidal vigour. This contrast illustrates the range of toxicity of compounds prevalent within the data set. When ranking the compounds later in the project, we will factor in this toxicity, as compounds which demonstrate high damage at low doses will be ranked highly.

Figure 4.10 illustrates the toxicity of compounds. We average the compounds in the data set by toxicity and damage recorded. Each point resembles a compound. We see that as the average dosage increases, compounds on average perform less effective in the experiments. The blue line is a least squares fit to the data, which illustrates the downward association between average rate and average damage.

4.3 EXAMINATION OF THE CHEMICAL SPACE

In this section, we apply clustering methods to the herbicide data. Clustering the compounds enables one to distinguish between certain properties, such as average potency and rate applied. One may also depict the chemical space in this way, by projecting the compounds onto their principal components. We also cluster the species to identify any patterns and similarities.

4.3.1 *Varying the number of clusters*

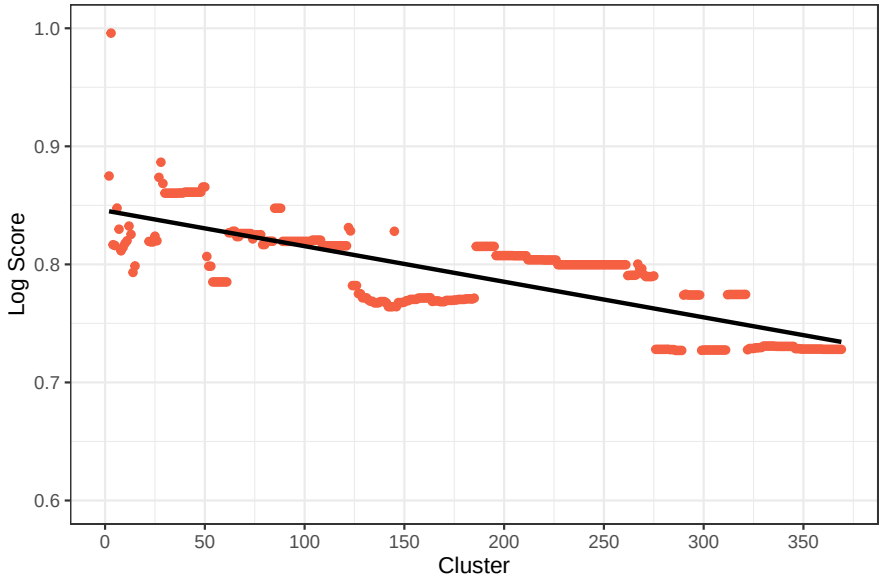
Here, we present further cluster analysis of the data. We use the cluster membership variable in-place of the compound id, resulting in fewer parameters to optimise over. Other variables in the model are Rate, pre- and -post emergence, warm, and if the plant was a weed or not. We initially assigning the compounds to two clusters. The data are split in a test-train ratio of 80 : 20. A decision tree is fit to the training data using the R package `rpart` [89]. We then assess the models predictive ability on the test set through the log score. We are interested in seeing the decay of the model error as we increase the number of clusters.

Figure 4.11a shows the log-score error as we increase the number of clusters the compounds can fall into. We observe a gradual decline in error as we increase the number of clusters, as expected. However, the error is minimised when the number of clusters is between 275-360, where 360 is the maximum number of trees permitted in the R package `rpart`. We also observe the variability in the error for fewer clusters which levels out as we have 60 clusters. Figure 4.11b depicts the times taken to fit the decision tree model. This is evidently increase with more clusters. However, there appears to be a certain numbers of clusters in the model where the time taken is considerably greater.

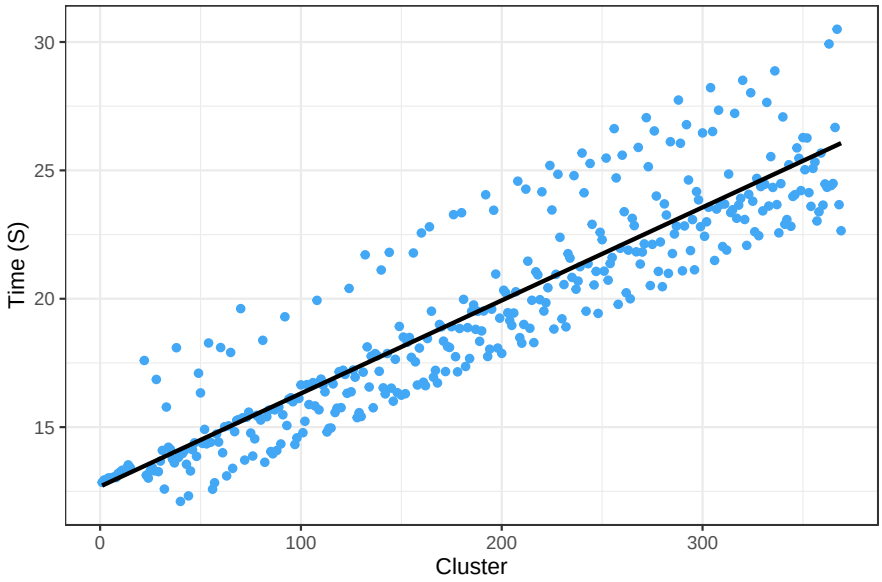
4.3.2 *Hierarchical clustering of compounds*

Hierarchical clustering is an unsupervised learning method used for the clustering of compounds. This method aims to group items most similar in nature through their cluster membership. Also know as farthest neighbour clustering, the process begins by assigning each element to its own cluster. The clusters are then sequentially combined into larger clusters by finding the minimum distance between elements of disjoint clusters. This is repeated until all elements are grouped into a single cluster. Clusters within the chemical space are calculated by identifying all pairwise Tanimoto distances, stored in a $n \times n$ matrix, where n is the number of compounds. We then pass this matrix on to the R function `hclust` which implements hierarchical clustering using the 'complete linkage' method.

In determining the optimal number of clusters within a Euclidean setting, we assess the within cluster sums of squares and the average silhouette width. The within cluster sums of squares measures the squared average distance of all the points within a cluster to the cluster centroid. Within the chemical space, we instead use the Tanimoto metric. The average silhouette method determines how well each object lies within each cluster and measures the level of confidence in determining the clustering of the data [81]. The closer the value is to one, the better the samples have been clustered. The closer the



(a)



(b)

Figure 4.11: Panel (a) shows log scores of a decision tree model when increasing the number of clusters. A least squares line of best fit is provided. Panel (b) Times taken to fit decision tree modes while increasing cluster membership

value is to -1, the greater the misclassification of the clustering is. The silhouette width is defined to be

$$
\begin{aligned}
S(i) &= \frac{b_i - a_i}{\max(b_i, a_i)}, \\
a_i &= \frac{1}{n_i} \sum_{j \in C_i} \text{dist}(i,j), \\
b_i &= \min_{i \notin C_k} \sum_{j \in C_k} \frac{\text{dist}(i,j)}{n_k}
\end{aligned}
$$

where a_i is the average distance between i and all other observations in the same cluster, b_i is the minimum average distance between observations at the other clusters, C_i is the cluster containing observation i, $\text{dist}(i,j)$ is the Tanimoto distance between observations i and j, and n_c is the cardinality of cluster C, i. e., the number of samples in the cluster.

To determine the optimal number of clusters for the data, we refer to the dendrogram, scree plot, and silhouette width demonstrated in Figure 4.12, Figure 4.13a and Figure 4.13b respectively. Using the dendrogram, we select the optimal number of clusters when there exists the greatest horizontal split with the greatest drop in height. For the scree-plot, we select the number of clusters using the elbow method, i. e., when levelling off occurs.

The dendrogram in Figure 4.12 illustrates optimal separation of the compounds being two clusters. Figure 4.13a and Figure 4.13b also show the optimal number of clusters is two. We will therefore continue our analysis using two clusters.

Having assigned the compounds to two clusters, we find the cluster sizes are 254 and 529. Using the compounds from each two clusters, we find the average damage on the logit-scale, as well as the average rate applied. The average damages were -2.42 and -1.44, with average rates 365 and 281 respectively. Based on this information, it appears the potency of the second cluster is greater than the first, since the average rate of application is lower, and the average damage is higher. To reiterate, 50 % damage equates to 0 damage on the logit-transformed scale. Interestingly, 19 compounds from cluster one were present in PPS, which amounts to 17 % of the compounds. One the other hand, 281 compounds from cluster two progressed to PPS, which amounts to 44 %. This indicates that the second cluster is grouped by herbicidal potency.

4.3.2.1 *Hierarchical clustering of species*

Overall there is information on 49 species of plant. The variables used to describe species are weed, monocot, and warm. each species is characterised by having at least one of these traits. That means there are $3! = 6$ possible combinations of characteristics a plant can have.

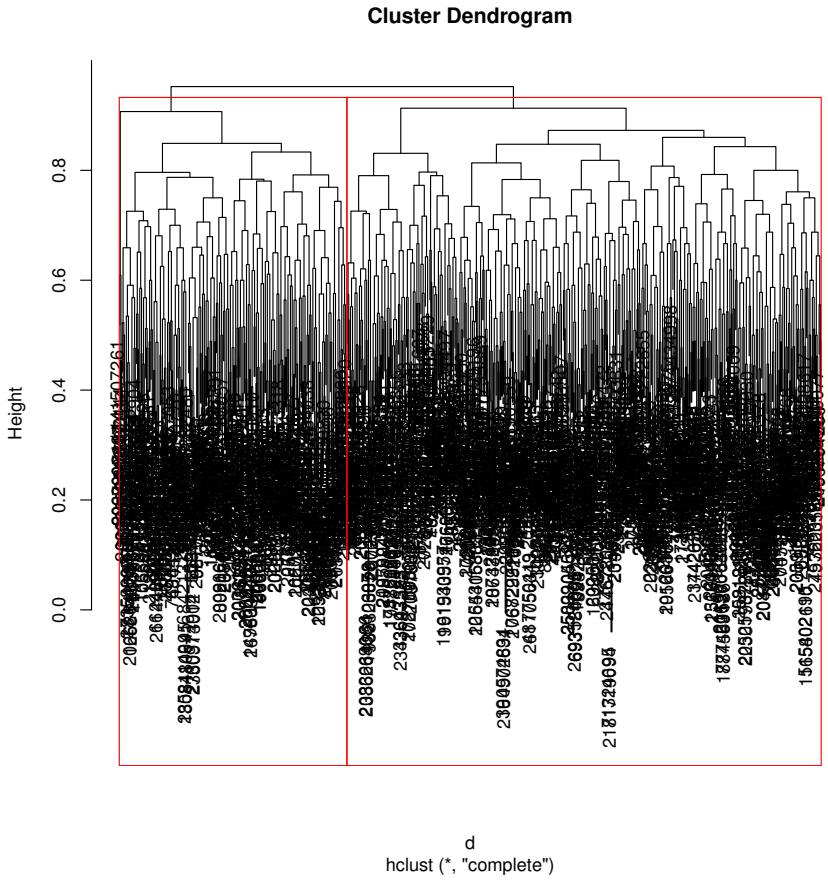
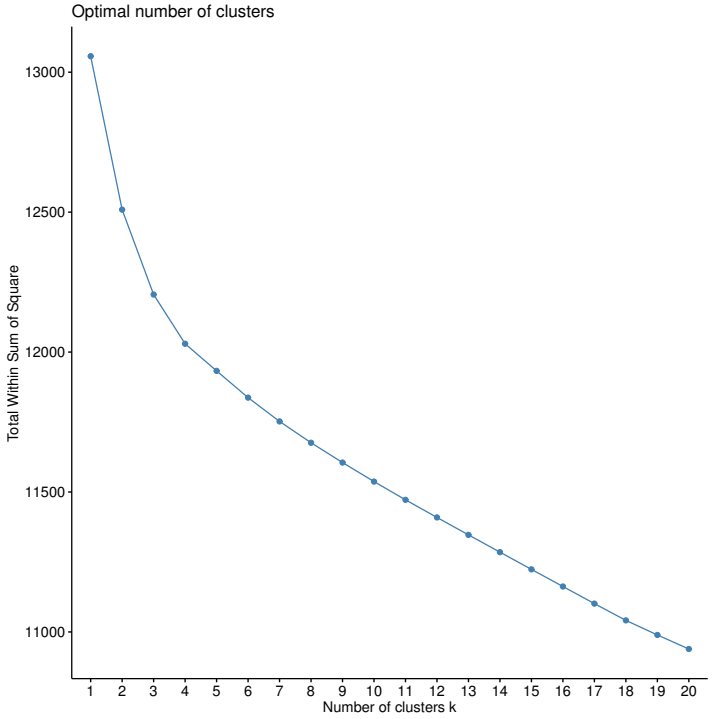
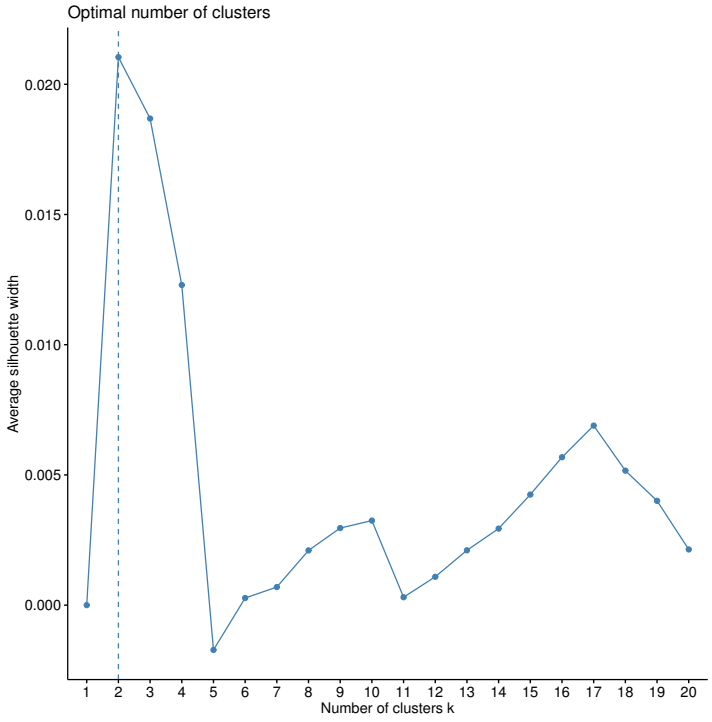


Figure 4.12: Dendrogram depicting two clusters as being optimal using hierarchical clustering



(a)



(b)

Figure 4.13: Panel (a) shows the total within-cluster sum of squared distances against the number of clusters. Panel (b) shows the average silhouette width against the number of clusters.

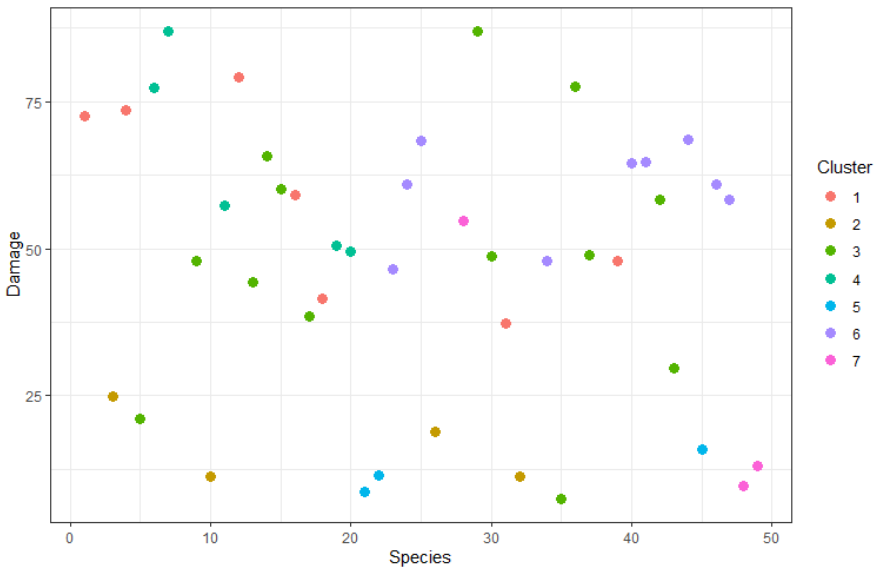


Figure 4.14: Hierarchical clustering according to the 7 different species of crop

The number of clusters should be less than the possible number of configurations of the species characteristics.

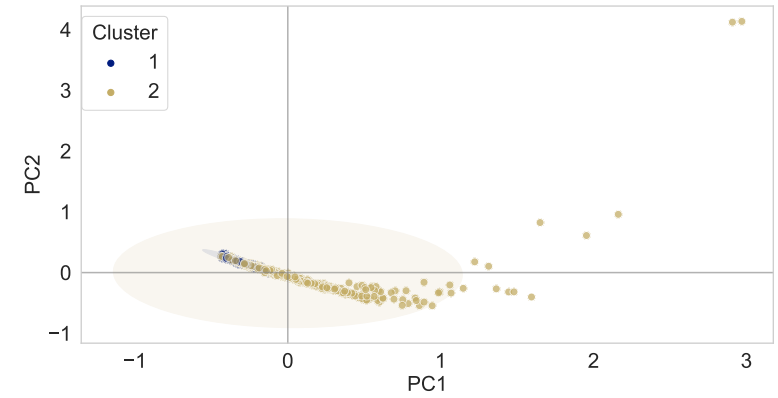
Upon inspection of the data, there are 15 species of plant used in EPS whereas there are 43 species in PPS. Species 41 was the only species to appear in EPS and not PPS. This means there were 29 new species used in PPS.

Using complete linkage hierarchical clustering, we group the species into 6 clusters and assess both the average damage and rate for each cluster. We make use of the full dataset in our analysis. From Figure 4.14, it is not so clear the distinction between species according to their damages.

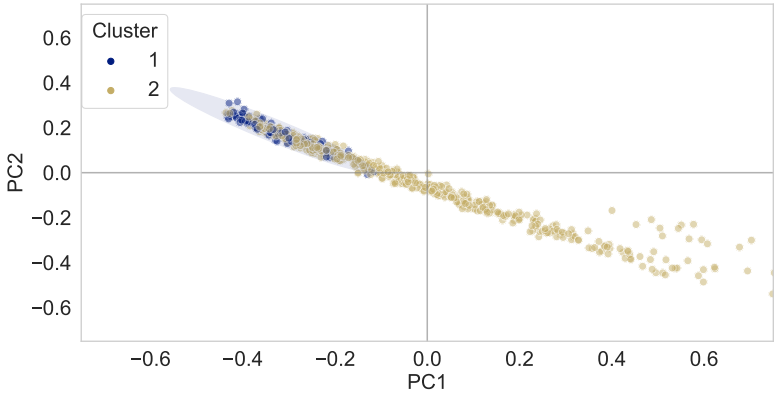
4.3.2.2 *Principal component analysis for compounds*

Principal component analysis may be used to identify lower-dimensional representations of the chemical space. This is performed by projecting the compound onto their first two principal components, for example.

Figure 4.15 shows the first two principal components with each compound plotted according to their cluster membership, see Section 4.3. The first two components account for approximately 26 % of the explained variation in the fingerprints. We see that the compounds within first cluster share similar properties, as indicated by the blue confidence ellipse. The compounds from the second cluster present greater variation and may be the more experimental compounds. The two outlying compounds in panel (a) have identical testing conditions and are both tested 36 times. Their average damages are 28.8 % and 48.0 %, which is lower than the overall average across all com-



(a)



(b)

Figure 4.15: Panel (a) shows the first two principal components plotted across the full range of the compounds in the EPS dataset, and are divided into two clusters. Confidence ellipses are plotted to one standard deviation's width. Panel (b) shows (a) zoomed in to distinguish the clusters.

pound, that being 55 %. This may inform us that one or both of the components are a contrast of potent and inactive compounds.

4.4 GP REGRESSION MODEL

We apply both a regression model as well as a classification model to the glasshouse data. The classification model is applied due to the novelty of this approach and that the damages from the experiments are naturally classes, so a classification model is preferred.

4.4.1 *Regression results*

We now present the parameters and predictive performance for the GP regression model. We assume the following relationship

$$\begin{aligned} \mathbf{y} &= \mathbf{X}^\top \boldsymbol{\beta} + P\mathbf{u} + \boldsymbol{\varepsilon}, \\ \mathbf{u} &\sim N_m(0, \sigma^2 R), \\ \varepsilon(\mathbf{x}) &\sim N(0, \tau^2 I), \end{aligned} \tag{4.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ are the logit-transformed damages and the extremes of 0 and 100 have been mapped to 1 and 99 respectively to eliminate singularities. This, however, is an arbitrary choice,and other methods of scaling are possible. In addition, $\mathbf{X}$ is the design matrix accounting for the fixed effects, $\boldsymbol{\beta}$ are regressor effects, P is the indicator matrix mapping the vector of unique random effects $\mathbf{u} \in \mathbb{R}^m$ to their location in the dataset, and ε is the residual error. The continuous covariate is scaled using the log transformation, having divided the variable by the maximum value of 1000. This helps to ensure numerical stability. The dichotomous variables in the data are weed, monocot, and warm. The parameters in the GP regression model are shown in Table 4.4 and the kernels used to calculate R are shown in Table 3.1, all being a function of Tanimoto distances.

From Table 4.3, we can identify the relationships between the predictors and the levels of damage from the glasshouse experiments. We see the that the effect of rate is positive, meaning an increase in rate by 0.01 leads to a unit increase of damage on the logit transformed scale. The effect of monocot is negative, meaning as we go from a dicot to a monocot, the damage decreases. The scale parameter is 0.46 for the exponential model and 0.47 for the Gaussian model. We also observe that the residual variation when applying Tanimoto covariance is much higher. This indicates less flexibility in comparison to the models accounting for the scale parameter. We also observe that the time taken to optimise the likelihood is much greater for the exponential and Gaussian models.

θ	independent	exponential	Gaussian	Tanimoto
β_0	−0.55	−0.23	−0.20	0.05
β_1	0.001	0.001	0.001	0.001
β_2	−1.91	−1.91	−1.91	−1.91
β_3	0.37	0.37	0.37	0.37
σ^2	1.68	4.74	4.66	17.31
ϕ	-	0.46	0.47	-
λ	0.45	0.01	0.13	0.13
Time(s)	0.28	2.16	2.07	0.75

Table 4.3: Parameter estimates from the GP regression models.

Parameter	Description
β_0	Intercept
β_1	Log(Rate/1000)
β_2	Monocot, Dicot = 0
β_3	Warm, Cold = 0
σ^2	compound variance
ϕ	length scale
λ	model variance to compound variance, $\frac{\tau^2}{\sigma^2}$
t	time (seconds) to optimise the likelihood

Table 4.4: Parameters used within the GP regression model. The categorical predictors indicated are referred to when the dummy takes the value 1.

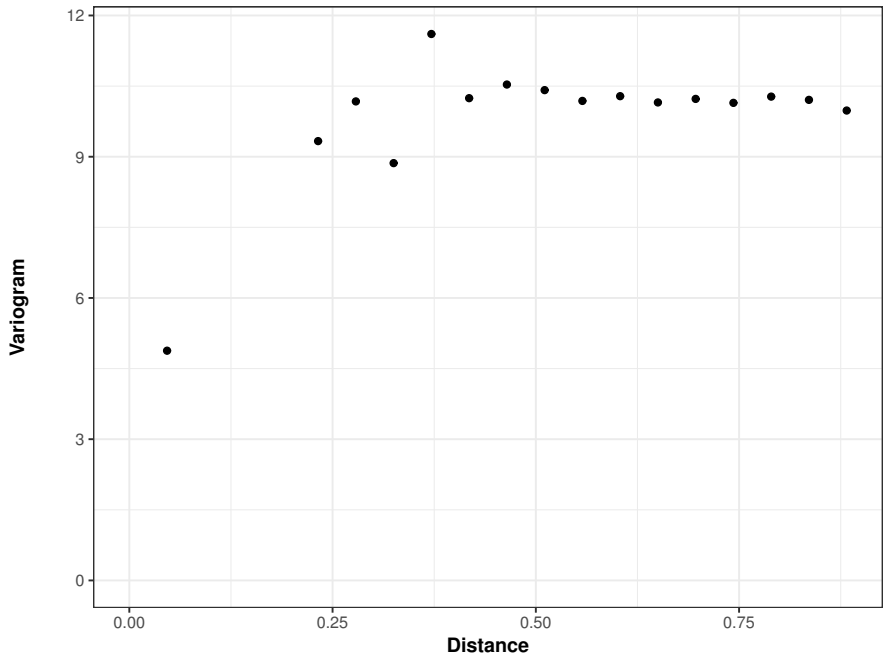


Figure 4.16: Variogram capturing spatial dependence of the Tanimoto similarity.

4.4.2 *Spatial dependence and Tanimoto similarity*

Based on the methodology described in Section 3.2, we now illustrate the results the findings results of the variogram. In Figure 4.16, we show the Tanimoto distance plotted against the variogram. The residuals are based on the GP regression model fit.

We create 20 evenly spaced intervals (bins) in our analysis, of which the 591 distinct Tanimoto similarities will be assigned to. Figure 4.16 shows there is some correlation in the Tanimoto similarity, i. e., the further distances become, the less spatial dependence is exhibited. This validates the spatial dependence of Tanimoto metric. This however is purely an exploratory experiment as it does not take into account the effect of the other variables.

4.5 CLASSIFICATION MODEL

In this section, we present our analysis to the glasshouse data. Since the damages are presented in classes, it is natural to develop a classification model. Before conducting analysis, the number of classes in the response variable are reduced from from 11 to 4 classes according to the following ranges $[0:0-20, 1:30-50, 2:60-80, 3:90-100]\%$. This was based on a suggestion by the computational chemists at Syngenta to help reduce bias due to human factors. We then fitted the proposed GP model with various link and correlation functions as

Parameter	Description
$G(\gamma_j)$	Link function
k_{rs}	Correlation function
$\hat{\alpha}_1$	Intercept 1
$\hat{\alpha}_2$	Intercept 2
$\hat{\alpha}_3$	Intercept 3
$\hat{\beta}_1$	Log(Rate/1000)
$\hat{\beta}_2$	Stage, Pre-emergence = 0
$\hat{\beta}_3$	Warm, Cold = 0
$\hat{\alpha}_4$	Monocot, Dicot = 0
$\hat{\phi}$	GP scale parameter
$\hat{\sigma}$	GP variance

Table 4.5: Parameters used within the GP classification model. The categorical predictors indicated are referred to when the dummy takes the value 1.

shown in Table 4.6 as well as a random forest model. We ensure the intercepts of the models are strictly increasing by defining $\theta_0 = \alpha_0$ and $\alpha_k = \alpha_{k-1} \exp \theta_k$, where $k \in \{1, \ldots, 3\}$. A high performance computer was used for carrying out the computations.

In estimating the model parameters, we fit the 16 GP models to the full dataset consisting of 35,740 distinct experiments. The parameter estimates of each model are presented in Table 4.6 along with their definitions given in Table 4.5. The standard errors of the parameters are presented inTable 4.7 From Table 4.6, we observe the signs of the parameters for the effects of rate, warm, and monocot are negative. Due to the way the model is defined, we can infer that increasing the rate of application in an experiment increases the odds of the damage falling within a higher class. We also infer that the warm variety of crops and those that are monocots are more susceptible to damage than dicots and those that are accustomed to a colder environment. We also observe that the model with Tanimoto covariance has a considerably higher variance parameter than the competing models. This is due to the absence of the scale parameter which controls for the correlation.

In assessing the classification accuracy of each model, we split the data into training and test sets and implement ten-fold cross validation. We take a random sample of 80% from each compound to form the training set and the remaining 20% are used within the test set. We implement a random forest model for further comparison, where the features of the fingerprints are used as predictor variables. The random forest was tuned using a grid search method. The parameters after tuning were found to be `mtry` = 30, `ntree` = 450, and `nodesize` = 29. The results of the classification performances, as as-

$G(\gamma_j)$	k_{rs}	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\alpha}_4$	$\hat{\phi}$	$\hat{\sigma}$
logit	independent	-1.415	-0.418	1.471	-0.9	1.938	-1.113	-1.076		9.32
logit	exponential	-1.415	-0.418	1.471	-0.9	1.938	-1.113	-1.076	0	9.32
logit	Gaussian	-1.415	-0.418	1.471	-0.9	1.938	-1.113	-1.076	0	9.319
logit	Tanimoto	-1.011	-0.014	1.874	-0.9	1.938	-1.113	-1.075		36.701
probit	independent	-0.794	-0.23	0.84	-0.498	1.076	-0.619	-0.577		2.976
probit	exponential	-0.556	0.008	1.078	-0.498	1.077	-0.619	-0.577	0.583	3.646
probit	Gaussian	-0.632	-0.068	1.002	-0.498	1.077	-0.619	-0.577	0.501	2.973
probit	Tanimoto	-0.967	-0.403	0.667	-0.499	1.077	-0.619	-0.577		11.728
log-log	independent	-0.215	0.336	1.435	-0.464	0.914	-0.589	-0.467		3.591
log-log	exponential	0.07	0.621	1.72	-0.464	0.915	-0.588	-0.467	0.601	4.46
log-log	Gaussian	-0.015	0.535	1.634	-0.464	0.915	-0.588	-0.467	0.505	3.595
log-log	Tanimoto 0.222	0.772	1.871	-0.464	0.915	-0.588	-0.466	13.953		
C-log-log	independent	-1.511	-0.813	0.463	-0.6	1.445	-0.792	-0.777		4.348
C-log-log	exponential	-1.083	-0.388	0.894	-0.607	1.465	-0.804	-0.781	0.565	5.249
C-log-log	Gaussian	-1.143	-0.449	0.833	-0.605	1.46	-0.801	-0.784	0.493	4.172
C-log-log	Tanimoto	-1.019	-0.325	0.955	-0.606	1.462	-0.802	-0.78		16.51

Table 4.6: Parameters of the GP models fit to the herbicide testing data. $\hat{\alpha}_1$, $\hat{\alpha}_2$, and $\hat{\alpha}_3$ are the ordered intercepts, $\hat{\beta}_1 = \log(\text{Rate}/1000)$, $\hat{\beta}_2$= Stage, $\hat{\beta}_3$= Warm, and $\hat{\beta}_4$=Monocot are the regressor variables, and $\hat{\sigma}^2$ and $\hat{\phi}$ are the variance and scale parameters.

$G(\gamma_j)$	k_{rs}	$SD(\hat{\alpha}_1)$	$SD(\hat{\alpha}_2)$	$SD(\hat{\alpha}_3)$	$SD(\hat{\beta}_1)$	$SD(\hat{\beta}_2)$	$SD(\hat{\beta}_3)$	$SD(\hat{\alpha}_4)$	$SD(\hat{\phi})$	$SD(\hat{\sigma})$
logit	independent	0.019	0.017	0.013	0.006	0.016	0.01	0.015		0.127
logit	exponential	0.019	0.017	0.013	0.006	0.016	0.01	0.015	0	0.128
logit	Gaussian	0.019	0.017	0.013	0.006	0.016	0.01	0.015	0	0.127
logit	Tanimoto	0.009	0.014	0.018	0.006	0.016	0.01	0.015		0.601
probit	independent	0.01	0.009	0.007	0.003	0.009	0.007	0.008		0.039
probit	exponential	0.17	0.17	0.17	0.003	0.009	0.007	0.008	0.009	0.055
probit	Gaussian	0.018	0.018	0.018	0.004	0.009	0.007	0.008	0.002	0.038
probit	Tanimoto	0.004	0.005	0.008	0.003	0.009	0.006	0.008		0.195
log-log	independent	0.011	0.009	0.008	0.004	0.011	0.008	0.009		0.043
log-log	exponential	0.412	0.415	0.415	0.004	0.011	0.008	0.009	0.024	0.089
log-log	Gaussian	0.019	0.017	0.018	0.004	0.011	0.008	0.009	0.003	0.041
log-log	Tanimoto	0.522	0.524	0.525	0.004	0.011	0.008	0.009		0.206
C-log-log	independent	0.011	0.012	0.01	0.004	0.009	0.008	0.007		0.096
C-log-log	exponential	0.032	0.031	0.034	0.005	0.009	0.008	0.008	0.014	0.087
C-log-log	Gaussian	0.007	0.009	0.01	0.005	0.012	0.011	0.008	0.008	0.109
C-log-log	Tanimoto	0.004	0.008	0.01	0.005	0.009	0.008	0.008		0.354

Table 4.7: Standard deviations of GP parameters obtained from bootstrapping the cross validation estimates

$G(\gamma_j)$	k_{rs}	Spherical	Log	AIC	Time (h)
C-log-log	Gaussian	−0.75913	0.74656	44,643.00	1.97
C-log-log	independent	−0.75910	0.74714	44,759.76	1.21
C-log-log	exponential	−0.75897	0.74860	44,647.45	2.06
probit	Gaussian	−0.75764	0.75149	44,828.16	2.13
probit	independent	−0.75764	0.75169	44,943.45	1.08
probit	exponential	−0.75750	0.75312	44,831.50	2.23
logit	exponential	−0.75691	0.76755	44,867.34	1.29
logit	Gaussian	−0.75691	0.76755	44,867.34	1.26
logit	independent	−0.75691	0.76755	44,865.34	1.14
C-log-log	Tanimoto	−0.75505	0.77549	44,766.58	1.88
probit	Tanimoto	−0.75443	0.77658	44,952.41	1.86
log-log	Gaussian	−0.74958	0.78669	46,602.51	2.35
log-log	independent	−0.74954	0.78708	46,718.09	1.51
log-log	exponential	−0.74927	0.78940	46,605.19	2.40
log-log	Tanimoto	−0.74407	0.82072	46,725.58	2.04
logit	Tanimoto	−0.74348	0.83215	44,875.88	1.77
RF		−0.63530			0.02

Table 4.8: Comparison of the classification performances of the random forest and GP models, ordered from highest to lowest in terms of their scores. The AIC and the time taken to optimise the likelihood are also given.

sessed through the log and spherical scores, as well as the models' AIC, are shown in Table 4.8. From Table 4.8 we see that the model with the complementary log-log link and Gaussian covariance has the greatest classification accuracy according to the log and spherical scores, and has an optimisation time of 1.97 hours. The model with the lowest AIC is the model with the probit link and Gaussian covariance. The model with the lowest classification accuracy is the random forest model, however, this has the greatest computational time of approximately .02 hours. A log score is omitted for the random forest model since forecasted probabilities contain 0, resulting in undefined values of the score.

4.5.1 *Ranking compounds*

Having fit the models, we now rank the compounds according to their predicted effect. We display the top 10 performing compounds based ont heir predicted effect in Table 4.9. These effects are based on the model with logit link and exponential covariance. We see in Table 4.9 that the compounds applied were tested in similar circumstances, i. e.,

according to the recorded damage across all experiments, rate applied, and number of experiments they appear in. According to our model, the top performing compound is ID1692213733, with a predicted effect of −8.01. This compound had a total of 38 active features within its fingerprint. Within the glasshouse experiments, this compound had an average damage of 95.4 across all experiments it was tested in, as well an average rate of 155 on the natural scale. This compound was tested in 24 experiments, somewhat less than many of the other top-performing compounds. We see the commercial standard, compound ID1530256782, is perhaps less effective as a herbicide, however, ranks in third place according to our model. Since this was applied in 1656 experiments, we believe this to be the commercial standard. This compound was applied at a much higher average rate of 443 across all glasshouse experiments, indicating this has less toxicity than the top performing compound.

Compound	Effect	Damage	Rate	N
ID1692213733	−8.01	95.4	155.0	24
ID20175182147	−7.90	97.5	436.7	36
ID1530256782	−7.38	92.1	443.0	1656
ID0536275554	−7.18	94.4	436.7	36
ID0445671389	−7.17	95.0	436.7	36
ID0056683832	−7.14	95.3	436.7	36
ID2003280057	−7.13	94.2	436.7	36
ID0033643583	−6.67	93.9	436.7	36
ID1277426043	−6.65	94.2	436.7	36
ID0291647191	−6.63	94.2	436.7	36

Table 4.9: 10 best predicted compounds from estimated effects using the logit link with exponential covariance. Summary information on their glasshouse experiments is also provided.

Table 4.10 shows the worst performing compounds according to our model. Compound ID2035763344 has the worst effect of 4.2. The average damage across all experiments is 1.7 %, indicating the compounds lack of herbicidal potency. The average rate is also very high, with the mean average rate across all experiments being 583.3. Rate in this case is the rate of application of a herbicide in its raw form. This compound was tested in 36 experiment in total. Compound ID0228086210 had an average damage of 0 %. This compound had a total of 23 active features within its fingerprint. Since damages can only be positive, that means this compound scored 0 % across all experiments, indicating the lack of any herbicidal efficacy. We observe that no commercial standard appears in the 10 worst predicted compounds. Between the top

performing compound ID1692213733 and the compound which scored 0 %, ID0228086210, there were only two fingerprints in common.

Compound	Effect	Damage	Rate	N
ID2035763344	4.20	1.7	583.3	36
ID2006139222	4.19	4.7	583.3	36
ID1141507261	4.12	1.4	436.7	36
ID1496344124	4.05	1.9	583.3	36
ID0115790621	4.04	2.5	583.3	36
ID2833335762	4.03	0.8	583.3	36
ID2678094703	4.03	1.1	583.3	36
ID0301843169	4.00	2.2	436.7	36
ID0228086210	3.99	0.0	436.7	36
ID1910205245	3.99	2.8	436.7	36

Table 4.10: 10 worst predicted compounds, logit link with exponential covariance. Summary information on their glasshouse experiments is also provided.

To illustrate the top performing and worst performing compounds, we illustrate their predicted effect, for an arbitrary glasshouse experiment, when varying the rate of application. This is shown in Figure 4.17. We see that from the lowest rate of application, the best performing compound starts in class 3 and the worst performing compound starts in the lowest class of 1. As we increase the rate for both classes, the probability of the top performing compound being in class 3 increases, whereas the probability of the worst performing compound of being in class 1 decreases. This illustrates the toxicity, and lack of, for both these compounds.

4.5.2 *Discovering potent compounds*

We next focus on searching the chemical space to identify potent compounds and present the results from GA when applied to the herbicidal fingerprints. The predicted effects from the 16 GP models are used as target variables and the features within the fingerprints as predictors. The GA were tuned using a grid search method over the mutation and crossover rates. The GA was run using 200 iterations, with the results displayed in Table 4.11. $\hat{u}_{GA}$ and $SD(\hat{u}_{GA})$ give the predicted effect from the optimal solution along with its standard error, and N_{GA} gives the number of features within the proposed solution. $\hat{u}_{GP}$ and $SD(\hat{u}_{GP})$ give the best solution and its standard error found by the corresponding GP model. Pr(crossover) and Pr(mutation) are the tuned crossover and mutation rates associated with the proposed solution.

$G(\gamma_j)$	k_{rs}	$\hat{u}_{GA}$	$SD(\hat{u}_{GA})$	N_{GA}	$\hat{u}_{GP}$	$SD(\hat{u}_{GP})$	Pr(cr)	Pr(mt)
logit	Tanimoto	-12.24	5.94	33	-7.31	5.83	0.4	0.7
C-log-log	Tanimoto	-8.54	3.13	35	-5.28	2.99	0.7	0.4
probit	Tanimoto	-6.68	2.48	33	-3.77	1.85	0.7	0.4
log-log	Tanimoto	-6.64	2.81	33	-3.36	1.94	0.4	0.1
C-log-log	Gaussian	-4.3	1.81	42	-5.26	1.55	0.1	0.4
C-log-log	exponential	-3.9	1.89	39	-5.37	1.79	0.1	0.4
log-log	Gaussian	-3.81	1.67	41	-4.34	1.04	0.1	0.4
probit	Gaussian	-3.52	1.51	41	-4.18	0.97	0.1	0.4
probit	exponential	-3.14	1.55	39	-4.26	1.08	0.1	0.4
log-log	exponential	-2.65	1.71	39	-3.59	1.18	0.4	0.1
logit	exponential	0	3.05	48	-7.28	3.37	0.1	0.1
logit	Gaussian	0	3.05	48	-7.27	3.37	0.1	0.1
logit	independent	0	3.05	48	-7.27	3.37	0.1	0.1
probit	independent	0	1.72	48	-4.14	1.05	0.1	0.1
log-log	independent	0	1.9	48	-4.24	1.12	0.1	0.1
C-log-log	independent	0	2.09	48	-5.11	1.76	0.1	0.1

Table 4.11: Comparison of GA solutions with the estimated effects from 16 GP models used as the target variable. Pr(cr) is the crossover rate and Pr(mt) is the mutation rate.

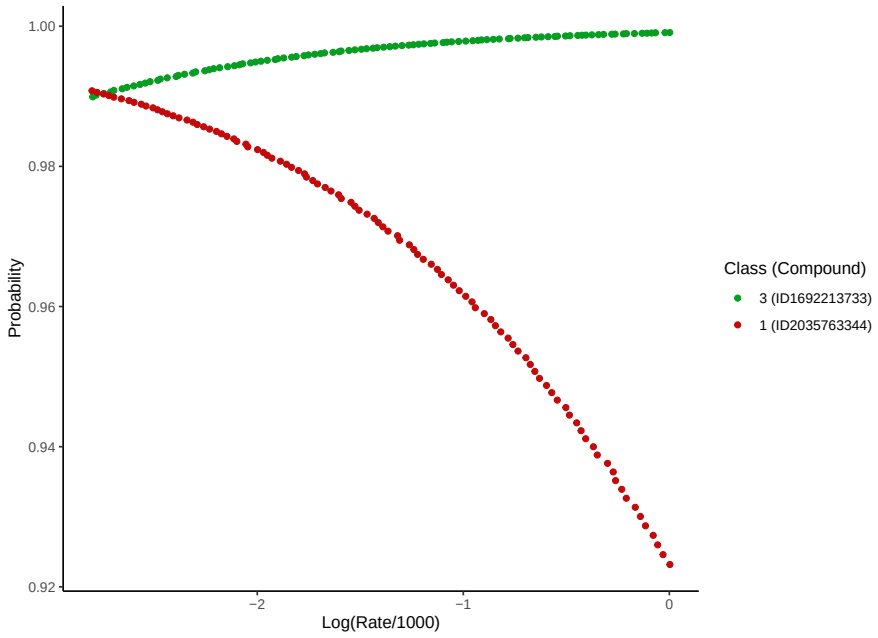


Figure 4.17: Predicted class probabilities for the highest and lowest class for best and worst compounds with varying rate. Predictions are based on the model with logit link function and exponential covariance.

We observe in Table 4.11 the model which proposes the greatest performing solution is the model with logit link and Tanimoto covariance, with $\hat{u}_{GA} = -12.24$ which is considerably greater than the corresponding GP solution, $\hat{u}_{GP} = -7.31$. This GA solution, however, has the most uncertainty, with $SD(\hat{u}_{GA}) = 5.94$, as well as the least amount of features retained, with $N_{GA} = 33$. We observe the models with a zero scale parameter are unable to predict the effect of the proposed solutions.

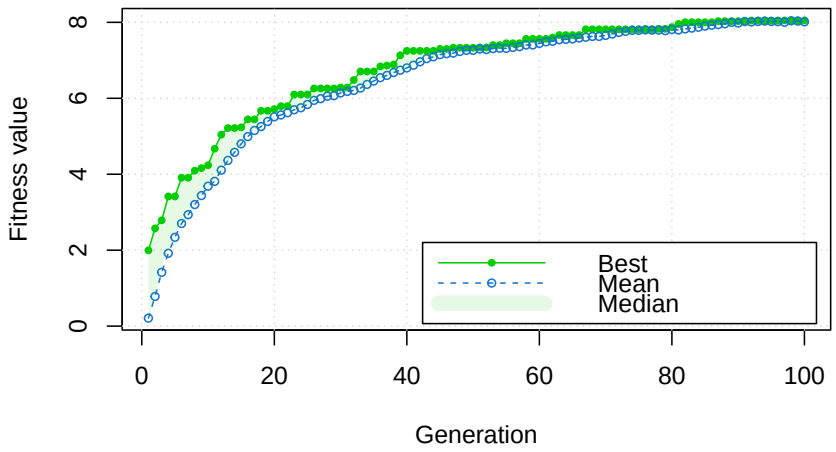


Figure 4.18: GA plot depicting fitness value (positive compound effect) against the surviving generations

5

CONCLUSION AND FUTURE WORK

The core aims of this project were to incorporate correlation within the chemical space within a statistical model through the use of GPs. Our hypothesis is that compounds which live close within the chemical space will exhibit similar herbicidal properties. In our application, we defined distances in the chemical space through the Tanimoto distance, a proper metric on the chemical space. Until now, GLM have not been able to account for correlated random effects, a feature attributed to GPs. Therefore, in our application, existing approaches would treat the effects of herbicides as independent. Our belief is that by incorporating correlated random effects through GPs, we can improve predictive performance. This could be achieved within a classification setting. A notable contribution of the project is that by using information on the seen compounds, we are able to make predictions on the effect and damages of unseen compounds, which is not yet possible with current models. In addition to being able to predict the effect of the compounds, we can predict the uncertainty of their effect.

To develop the classification model we employed the Laplace approximation to provide a normal approximate the likelihood. Another contribution of this project is the use of the Laplace approximation for estimation and prediction of ordinal data. This has yet to be applied within the literature.

When developing the GP models, we implemented several covariances, including the Tanimoto covariance. This in itself is a novel approach to GPs and can be regarded as another contribution of this project. In addition to the covariances, we introduced several link functions, including symmetrical and non-symmetrical link functions. These models were compared with the random forest.

Having developed a model in a classification setting, we developed a ranking system for the compounds. This was based on the predicted effect from the GP model. We demonstrated the top performing compounds according to our model, as well as the worst performing compounds.

We also conducted a simulation study, simulating data from a GP model an using all models under consideration to see if the true model could be identified. Our simulation study confirmed the suitability of proper scoring rules as a means of assessing model performance. We showed the best performing model was identified as the true model through proper scoring rules, hence validating this metric's suitability for assessing the classification model performance.

This project also implemented several optimisation methods for the facilitation of drug-discovery. This was performed by using the effects estimated from the GP model as the target variable and the fingerprint features as covariates. We showed that GAs could propose compounds of the highest efficacy.

5.1 LIMITATIONS AND FUTURE WORK

The issue of the lack of data accounting for the middle class damages meant the GP model could not predict well for certain outcomes. One possible remedy would be to take interim readings after the initial application so that the progress of the compound is recorded. In-turn, this may provide an indication the the speed of action, which could then be accounted for within the model to improve fit. This would also mean that the damage classes of the GP model would not have to be reduced to fewer classes, thereby providing a more accurate estimate of the compounds performance in the middle class damages.

One may also critique the manner in which the damages are recorded by the biologists at Syngenta. One could argue that the damages assigned by the biologists at Syngenta have some variability since not all opinions will consistent. A colour code is given to eliminate this variation, however not all damage is presented in the same way, as some show lack of pigment, while others show a stunt in growth or even necrosis. To eliminate the variation from the biologist's perception of damage, an image classifier could be trained to consistently rank the class of damage resulting from a glasshouse experiment. This would require many pictures of damages and have the biologists manually label their opinion of what constitutes the class damage. Given enough data, and variability in the data, are provided, this could provide a more accurate, and more cost effective approach.

A notable criticism of our methodology is how we encode the molecule into the mathematical expression of a fingerprint. When geometry of a compound is neglected, only the atom connectivity information is retained. Modelling chemicals as graphs therefore significantly reduces the complexity of the compound [49].

Possible directions for future work may include incorporating 3D molecular structures, as well as 4D structures, to capture more information about the compounds location and construct within the chemical space. When working with 2D representations, information on the compounds structure is neglected, such as positional arguments. This information can be captured with the distance measure to gain a more accurate indication of how close chemical compounds are with each other within the chemical space.

Another criticism of our methodology is within the use of the Tanimoto covariance. Since this lacks the scale parameter that adjusts for the correlation in the data, the fit of the model is reduced. To

remedy this, one may incorporate a parameter in the exponent whilst ensuring the positive definiteness is retained.

Further analysis could incorporate other binary metrics, such as the cosine similarity or the dice coefficient within the covariance of GPs to assess the improvement of model predictions. One may also wish to explore other link functions.

GPs don't work well for large data due to the presence of large matrices. This is due to needing the inversion and determinant of these matrices. Some of the methods to deal with this issue have been proposed in this thesis. A sparse matrix based on the Tanimoto similarity can be constructed by reducing the number of features in the fingerprints, hence making compounds more dissimilar and forcing some of the similarities to 0.

The GP model facilitates the application of Bayesian optimisation for exploration and exploitation of the chemical space. Bayesian optimisation can be used to propose new compounds to test in sequential experiments. The idea is to choose to test the compound that will result in the largest expected improvement towards identifying the optimal compound. Optimising the expected improvement can be done using SA or GA as demonstrated.

BIBLIOGRAPHY

[1] Alan Agresti. *Analysis of ordinal categorical data*. Vol. 656. John Wiley & Sons, 2010.

[2] Syed M. Ali, Michael Z. Hoemann, Jeffrey Aubé, Gunda I. Georg, Lester A. Mitscher, and Lalith R. Jayasinghe. "Butitaxel Analogues: Synthesis and Structure-Activity Relationships." In: *Journal of Medicinal Chemistry* 40 (2 Jan. 1997), pp. 236–241. ISSN: 0022-2623. DOI: 10.1021/jm960505t. URL: https://pubs.acs.org/doi/10.1021/jm960505t.

[3] Jürgen Bajorath. *Chemoinformatics and Computational Chemical Biology*. Ed. by Jürgen Bajorath. Vol. 672. Humana Press, 2011. ISBN: 978-1-60761-838-6. DOI: 10.1007/978-1-60761-839-3. URL: http://link.springer.com/10.1007/978-1-60761-839-3.

[4] Dávid Bajusz, Anita Rácz, and Károly Héberger. "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" In: *Journal of Cheminformatics* 7 (1 Dec. 2015), p. 20. ISSN: 1758-2946. DOI: 10.1186/s13321-015-0069-3. URL: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-015-0069-3.

[5] A. Banerjee, D. B. Dunson, and S. T. Tokdar. "Efficient Gaussian process regression for large datasets." In: *Biometrika* 100 (1 Mar. 2013), pp. 75–89. ISSN: 0006-3444. DOI: 10.1093/biomet/ass068. URL: https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/ass068.

[6] Sudipto Banerjee, Alan E. Gelfand, Andrew O. Finley, and Huiyan Sang. "Gaussian predictive process models for large spatial data sets." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70 (4 Sept. 2008), pp. 825–848. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2008.00663.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2008.00663.x.

[7] Andreas Bender and Robert C. Glen. "Molecular similarity: a key technique in molecular informatics." In: *Organic & Biomolecular Chemistry* 2 (22 2004), p. 3204. ISSN: 1477-0520. DOI: 10.1039/b409813g. URL: http://xlink.rsc.org/?DOI=b409813g.

[8] Regine S. Bohacek, Colin McMartin, and Wayne C. Guida. "The art and practice of structure-based drug design: A molecular modeling perspective." In: *Medicinal Research Reviews* 16 (1 Jan. 1996), pp. 3–50. ISSN: 0198-6325. DOI: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6. URL: https:

//onlinelibrary.wiley.com/doi/10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6.

[9] Danail Bonchev. *Chemical graph theory: introduction and fundamentals*. Vol. 1. Taylor & Francis, 1991. ISBN: 9780856264542. URL: https://books.google.co.uk/books?id=X0AG7HhiccoC.

[10] James G Booth and James P Hobert. "Standard errors of prediction in generalized linear mixed models." In: *Journal of the American Statistical Association* 93.441 (1998), pp. 262–272.

[11] Mathieu Bouchard, Anne-Laure Jousselme, and Pierre-Emmanuel Doré. "A proof for the positive definiteness of the Jaccard index matrix." In: *International Journal of Approximate Reasoning* 54.5 (2013), pp. 615–626.

[12] Leo Breiman. "Bagging predictors." In: *Machine Learning* 24 (2 Aug. 1996), pp. 123–140. ISSN: 0885-6125. DOI: 10.1007/BF00058655. URL: http://link.springer.com/10.1007/BF00058655.

[13] Leo Breiman. "Random forests." In: *Machine learning* 45 (1 2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[14] C. G. Broyden. "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations." In: *IMA Journal of Applied Mathematics* 6 (1 1970), pp. 76–90. ISSN: 0272-4960. DOI: 10.1093/imamat/6.1.76. URL: https://academic.oup.com/imamat/article-lookup/doi/10.1093/imamat/6.1.76.

[15] Gaspar Cano, Jose Garcia-Rodriguez, Alberto Garcia-Garcia, Horacio Perez-Sanchez, Jón Atli Benediktsson, Anil Thapa, and Alastair Barr. "Automatic selection of molecular descriptors using random forest: Application to drug discovery." In: *Expert Systems with Applications* 72 (Apr. 2017), pp. 151–159. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.12.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417416306819.

[16] Bradley P Carlin, Alan E Gelfand, and Sudipto Banerjee. *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC, 2014. ISBN: 9781439819173.

[17] Sangit Chatterjee, Matthew Laudato, and Lucy A. Lynch. "Genetic algorithms and their statistical applications: an introduction." In: *Computational Statistics & Data Analysis* 22 (6 Oct. 1996), pp. 633–651. ISSN: 01679473. DOI: 10.1016/0167-9473(96)00011-4. URL: https://linkinghub.elsevier.com/retrieve/pii/0167947396000114.

[18] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. "The rise of deep learning in drug discovery." In: *Drug Discovery Today* 23 (6 June 2018), pp. 1241–1250. ISSN: 13596446. DOI: 10.1016/j.drudis.2018.01.039.

URL: https://linkinghub.elsevier.com/retrieve/pii/S1359644617303598.

[19] Xin Chen and Charles H. Reynolds. "Performance of Similarity Measures in 2D Fragment-Based Similarity Searching: Comparison of Structural Descriptors and Similarity Coefficients." In: *Journal of Chemical Information and Computer Sciences* 42 (6 Nov. 2002), pp. 1407–1414. ISSN: 0095-2338. DOI: 10.1021/ci025531g. URL: https://pubs.acs.org/doi/10.1021/ci025531g.

[20] Artem Cherkasov et al. "QSAR Modeling: Where Have You Been? Where Are You Going To?" In: *Journal of Medicinal Chemistry* 57 (12 June 2014), pp. 4977–5010. ISSN: 0022-2623. DOI: 10.1021/jm4004285. URL: https://pubs.acs.org/doi/10.1021/jm4004285.

[21] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C Tappert. "A survey of binary similarity and distance measures." In: *Journal of Systemics, Cybernetics and Informatics* 8 (1 2010), pp. 43–48. ISSN: 1690-4524.

[22] Taeryon Choi and Mark J. Schervish. "On posterior consistency in nonparametric regression problems." In: *Journal of Multivariate Analysis* 98 (10 Nov. 2007), pp. 1969–1987. ISSN: 0047259X. DOI: 10.1016/j.jmva.2007.01.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0047259X07000048.

[23] Wei Chu, Zoubin Ghahramani, and Christopher K I Williams. "Gaussian processes for ordinal regression." In: *Journal of machine learning research* 6 (7 2005), pp. 1019–1041. ISSN: 1532-4435. URL: http://jmlr.org/papers/v6/chu05a.html.

[24] S. R. Colby. "Calculating Synergistic and Antagonistic Responses of Herbicide Combinations." In: *Weeds* 15 (1 Jan. 1967), p. 20. ISSN: 0096719X. DOI: 10.2307/4041058. URL: https://www.jstor.org/stable/4041058?origin=crossref.

[25] Lehel Csató and Manfred Opper. "Sparse On-Line Gaussian Processes." In: *Neural Computation* 14 (3 Mar. 2002), pp. 641–668. ISSN: 0899-7667. DOI: 10.1162/089976602317250933. URL: https://direct.mit.edu/neco/article/14/3/641-668/6594.

[26] Wojciech Marian Czarnecki. "Weighted Tanimoto Extreme Learning Machine with Case Study in Drug Discovery." In: *IEEE Computational Intelligence Magazine* 10 (3 Aug. 2015), pp. 19–29. ISSN: 1556-603X. DOI: 10.1109/MCI.2015.2437312. URL: http://ieeexplore.ieee.org/document/7160842/.

[27] Danishuddin and Asad U. Khan. "Descriptors and their selection methods in QSAR analysis: paradigm for drug design." In: *Drug Discovery Today* 21 (8 Aug. 2016), pp. 1291–1302. ISSN: 13596446. DOI: 10.1016/j.drudis.2016.06.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S1359644616302318.

[28] David Duvenaud. "Automatic model construction with Gaussian processes." University of Cambridge, 2014.

[29] Mark Ebden. "Gaussian Processes: A Quick Introduction." In: *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford* 91 (May 2015), pp. 424–436. URL: http://arxiv.org/abs/1505.02965.

[30] Sylvan Elhay and Jaroslav Kautsky. "Algorithm 655: Iqpack: Fortran subroutines for the weights of interpolatory quadratures." In: *ACM Transactions on Mathematical Software (TOMS)* 13.4 (1987), pp. 399–415.

[31] Peter Fenner and Edward Pyzer-Knapp. "Privacy-preserving Gaussian process regression–a modular approach to the application of homomorphic encryption." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 3866–3873.

[32] Andrew O Finley, Huiyan Sang, Sudipto Banerjee, and Alan E Gelfand. "Improving the performance of predictive process modeling for large datasets." In: *Computational statistics & data analysis* 53.8 (2009), pp. 2873–2884.

[33] Michael A Fligner, Joseph S Verducci, and Paul E Blower. "A Modification of the Jaccard-Tanimoto Similarity Index for Diverse Selection of Chemical Compounds Using Binary Strings." In: *Technometrics* 44 (2 May 2002), pp. 110–119. ISSN: 0040-1706. DOI: 10.1198/004017002317375064. URL: http://www.tandfonline.com/doi/abs/10.1198/004017002317375064.

[34] Jerry L. Flint, Paul L. Cornelius, and Michael Barrett. "Analyzing Herbicide Interactions: A Statistical Treatment of Colby's Method." In: *Weed Technology* 2 (3 July 1988), pp. 304–309. ISSN: 0890-037X. DOI: 10.1017/S0890037X00030645. URL: https://www.cambridge.org/core/product/identifier/S0890037X00030645/type/journal_article.

[35] Reinhard Furrer, Marc G Genton, and Douglas Nychka. "Covariance Tapering for Interpolation of Large Spatial Datasets." In: *Journal of Computational and Graphical Statistics* 15 (3 Sept. 2006), pp. 502–523. ISSN: 1061-8600. DOI: 10.1198/106186006X132178. URL: http://www.tandfonline.com/doi/abs/10.1198/106186006X132178.

[36] Kaifu Gao, Duc Duy Nguyen, Vishnu Sresht, Alan M. Mathiowetz, Meihua Tu, and Guo-Wei Wei. "Are 2D fingerprints still valuable for drug discovery?" In: *Physical Chemistry Chemical Physics* 22 (16 2020), pp. 8373–8390. ISSN: 1463-9076. DOI: 10.1039/D0CP00305K. URL: http://xlink.rsc.org/?DOI=D0CP00305K.

[37] Thomas Gärtner, Quoc Viet Le, and Alex J Smola. "A short tour of kernel methods for graphs." In: *Under Preparation* (2006).

[38] Amir Abadi Ghadim, Ross Kingwell, and David Pannell. "An economic evaluation of deep tillage to reduce soil compaction on crop-livestock farms in Western Australia." In: *Agricultural Systems* 37 (3 Jan. 1991), pp. 291–307. ISSN: 0308521X. DOI: 10.1016/0308-521X(91)90038-C. URL: https://linkinghub.elsevier.com/retrieve/pii/0308521X9190038C.

[39] Ali Ghaheri, Saeed Shoar, Mohammad Naderan, and Sayed Shahabuddin Hoseini. "The Applications of Genetic Algorithms in Medicine." In: *Oman Medical Journal* 30 (6 Nov. 2015), pp. 406–416. ISSN: 1999768X. DOI: 10.5001/omj.2015.82. URL: http://www.omjournal.org/fultext_PDF.aspx?DetailsID=704&type=fultext.

[40] Tilmann Gneiting and Adrian E Raftery. "Strictly Proper Scoring Rules, Prediction, and Estimation." In: *Journal of the American Statistical Association* 102 (477 Mar. 2007), pp. 359–378. ISSN: 0162-1459. DOI: 10.1198/016214506000001437. URL: http://www.tandfonline.com/doi/abs/10.1198/016214506000001437.

[41] J. C. Gower and P. Legendre. "Metric and Euclidean properties of dissimilarity coefficients." In: *Journal of Classification* 3 (1 Mar. 1986), pp. 5–48. ISSN: 0176-4268. DOI: 10.1007/BF01896809. URL: http://link.springer.com/10.1007/BF01896809.

[42] John C. Gower and Matthijs J. Warrens. *Similarity, Dissimilarity, and Distance, Measures of*. Wiley, May 2017, pp. 1–11. DOI: 10.1002/9781118445112.stat02470.pub2. URL: https://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat02470.pub2.

[43] John Clifford Gower. "Properties of Euclidean and non-Euclidean distance matrices." In: *Linear algebra and its applications* 67 (1985), pp. 81–97.

[44] J. Gressel and L.A. Segel. "The paucity of plants evolving genetic resistance to herbicides: Possible reasons and implications." In: *Journal of Theoretical Biology* 75 (3 Dec. 1978), pp. 349–371. ISSN: 00225193. DOI: 10.1016/0022-5193(78)90340-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0022519378903405.

[45] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Vol. 1. Springer New York, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7. URL: http://link.springer.com/10.1007/978-0-387-84858-7.

[46] Matthew J. Heaton et al. "A Case Study Competition Among Methods for Analyzing Large Spatial Data." In: *Journal of Agricultural, Biological and Environmental Statistics* 24 (3 Sept. 2019), pp. 398–425. ISSN: 1085-7117. DOI: 10.1007/s13253-018-00348-w. URL: http://link.springer.com/10.1007/s13253-018-00348-w.

[47] Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W. Coley. "Uncertainty Quantification Using Neural Networks for Molecular Property Prediction." In: (May 2020). URL: http://arxiv.org/abs/2005.10036.

[48] Jennifer A. Hoeting, Richard A. Davis, Andrew A. Merton, and Sandra E. Thompson. "Model Selection For Geostatistical Models." In: *Ecological Applications* 16 (1 Feb. 2006), pp. 87–98. ISSN: 1051-0761. DOI: 10.1890/04-0576. URL: http://doi.wiley.com/10.1890/04-0576.

[49] Ovidiu Ivanciuc. *Representing Two-Dimensional (2D) Chemical Structures with Molecular Graphs*. Apr. 2010, pp. 1–36. DOI: 10.1201/9781420082999-c1.

[50] P. K. Jensen and P. Kudsk. "Prediction of herbicide activity." In: *Weed Research* 28 (6 Dec. 1988), pp. 473–478. ISSN: 0043-1737. DOI: 10.1111/j.1365-3180.1988.tb00830.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1365-3180.1988.tb00830.x.

[51] I.M. Kapetanovic. "Computer-aided drug discovery and development (CADDD): In silico-chemico-biological approach." In: *Chemico-Biological Interactions* 171 (2 Jan. 2008), pp. 165–176. ISSN: 00092797. DOI: 10.1016/j.cbi.2006.12.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0009279706003541.

[52] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. "Molecular graph convolutions: moving beyond fingerprints." In: *Journal of Computer-Aided Molecular Design* 30 (8 Aug. 2016), pp. 595–608. ISSN: 0920-654X. DOI: 10.1007/s10822-016-9938-8. URL: http://link.springer.com/10.1007/s10822-016-9938-8.

[53] Peter Kirkpatrick and Clare Ellis. "Chemical space." In: *Nature* 432 (7019 Dec. 2004), pp. 823–823. ISSN: 0028-0836. DOI: 10.1038/432823a. URL: http://www.nature.com/articles/432823a.

[54] Andrew R. Kniss, Joseph D. Vassios, Scott J. Nissen, and Christian Ritz. "Nonlinear Regression Analysis of Herbicide Absorption Studies." In: *Weed Science* 59 (4 Dec. 2011), pp. 601–610. ISSN: 0043-1745. DOI: 10.1614/WS-D-11-00034.1. URL: https://www.cambridge.org/core/product/identifier/S0043174500020877/type/journal_article.

[55] Andrew R. Leach and Valerie J. Gillet. *An Introduction To Chemoinformatics*. Springer Netherlands, 2007. ISBN: 978-1-4020-6290-2. DOI: 10.1007/978-1-4020-6291-9. URL: http://link.springer.com/10.1007/978-1-4020-6291-9.

[56] Andy Liaw and Matthew Wiener. "Classification and regression by randomForest." In: *R news* 2 (3 2002), pp. 18–22. ISSN: 1609-3631.

[57] Alan H Lipkus. "A proof of the triangle inequality for the Tanimoto distance." In: *Journal of Mathematical Chemistry* 26 (1-3 1999), pp. 263–265. DOI: 10.1023/A:1019154432472.

[58] Yu-Chen Lo, Stefano E. Rensi, Wen Torng, and Russ B. Altman. "Machine learning in chemoinformatics and drug discovery." In: *Drug Discovery Today* 23 (8 Aug. 2018), pp. 1538–1546. ISSN: 13596446. DOI: 10.1016/j.drudis.2018.05.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S1359644617304695.

[59] Vinicius Gonçalves Maltarollo, Thales Kronenberger, Gabriel Zarzana Espinoza, Patricia Rufino Oliveira, and Kathia Maria Honorio. "Advances with support vector machines for novel drug discovery." In: *Expert Opinion on Drug Discovery* 14 (1 Jan. 2019), pp. 23–33. ISSN: 1746-0441. DOI: 10.1080/17460441.2019.1549033. URL: https://www.tandfonline.com/doi/full/10.1080/17460441.2019.1549033.

[60] K V Mardia, J T Kent, and J M Bibby. *Multivariate Analysis*. London NA1 7DX: Academic Press Ltd, 1979.

[61] Bertil Matérn. *Spatial Variation*. Vol. 36. Springer New York, 1986. ISBN: 978-0-387-96365-5. DOI: 10.1007/978-1-4615-7892-5. URL: http://link.springer.com/10.1007/978-1-4615-7892-5.

[62] Charles E McCulloch. "Maximum likelihood algorithms for generalized linear mixed models." In: *Journal of the American statistical Association* 92.437 (1997), pp. 162–170.

[63] José L. Medina-Franco, Norberto Sánchez-Cruz, Edgar López-López, and Bárbara I. Díaz-Eufracio. "Progress on open chemoinformatic tools for expanding and exploring the chemical space." In: *Journal of Computer-Aided Molecular Design* (June 2021). ISSN: 0920-654X. DOI: 10.1007/s10822-021-00399-1.

[64] John B. O. Mitchell. "Machine learning methods in chemoinformatics." In: *WIREs Computational Molecular Science* 4 (5 Sept. 2014), pp. 468–481. ISSN: 1759-0876. DOI: 10.1002/wcms.1183. URL: https://onlinelibrary.wiley.com/doi/10.1002/wcms.1183.

[65] Henry B. Moss and Ryan-Rhys Griffiths. "Gaussian Process Molecule Property Prediction with FlowMO." In: *arXiv preprint arXiv:2010.01118* (Oct. 2020). URL: http://arxiv.org/abs/2010.01118.

[66] Cristian R. Munteanu, Enrique Fernandez-Blanco, Jose A. Seoane, Pilar Izquierdo-Novo, Jose Angel Rodriguez-Fernandez, Jose Maria Prieto-Gonzalez, Juan R. Rabunal, and Alejandro Pazos. "Drug Discovery and Design for Complex Diseases through QSAR Computational Methods." In: *Current Pharmaceutical Design* 16 (24 Aug. 2010), pp. 2640–2655. ISSN: 13816128. DOI: 10.2174/138161210792389252. URL: http://www.eurekaselect.

com/openurl/content.php?genre=article&issn=1381-6128&volume=16&issue=24&spage=2640.

[67] Bruno J. Neves, Rodolpho C. Braga, Cleber C. Melo-Filho, José Teófilo Moreira-Filho, Eugene N. Muratov, and Carolina Horta Andrade. "QSAR-Based Virtual Screening: Advances and Applications in Drug Discovery." In: *Frontiers in Pharmacology* 9 (Nov. 2018), p. 1275. ISSN: 1663-9812. DOI: 10.3389/fphar.2018.01275. URL: https://www.frontiersin.org/article/10.3389/fphar.2018.01275/full.

[68] Duc Duy Nguyen, Zixuan Cang, Kedi Wu, Menglun Wang, Yin Cao, and Guo-Wei Wei. "Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges." In: *Journal of Computer-Aided Molecular Design* 33 (1 Jan. 2019), pp. 71–82. ISSN: 0920-654X. DOI: 10.1007/s10822-018-0146-6. URL: http://link.springer.com/10.1007/s10822-018-0146-6.

[69] Ole K. Nielsen, Christian Ritz, and Jens. C. Streibig. "Nonlinear Mixed-Model Regression to Analyze Herbicide Dose-Response Relationships." In: *Weed Technology* 18 (1 Jan. 2004), pp. 30–37. ISSN: 0890-037X. DOI: 10.1614/WT-03-070R1. URL: http://www.bioone.org/doi/abs/10.1614/WT-03-070R1.

[70] William S Noble. "What is a support vector machine?" In: *Nature Biotechnology* 24 (12 Dec. 2006), pp. 1565–1567. ISSN: 1087-0156. DOI: 10.1038/nbt1206-1565. URL: http://www.nature.com/articles/nbt1206-1565.

[71] Debleena Paul, Gaurav Sanap, Snehal Shenoy, Dnyaneshwar Kalyane, Kiran Kalia, and Rakesh K. Tekade. "Artificial intelligence in drug discovery and development." In: *Drug Discovery Today* 26 (1 Jan. 2021), pp. 80–93. ISSN: 13596446. DOI: 10.1016/j.drudis.2020.10.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S1359644620304256.

[72] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*. Springer, 2004, pp. 63–71. ISBN: 026218253X. DOI: 10.1007/978-3-540-28650-9_4. URL: http://link.springer.com/10.1007/978-3-540-28650-9_4.

[73] Stephen W. Raudenbush and Anthony S. Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage, 2002. ISBN: 9780761919049.

[74] Jean-Louis Reymond and Mahendra Awale. "Exploring Chemical Space for Drug Discovery Using the Chemical Universe Database." In: *ACS Chemical Neuroscience* 3 (9 Sept. 2012), pp. 649–657. ISSN: 1948-7193. DOI: 10.1021/cn3000422. URL: https://pubs.acs.org/doi/10.1021/cn3000422.

[75] Ahmet Sureyya Rifaioglu, Heval Atas, Maria Jesus Martin, Rengul Cetin-Atalay, Volkan Atalay, and Tunca Doğan. "Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases." In: *Briefings in Bioinformatics* 20 (5 Sept. 2019), pp. 1878–1912. ISSN: 1467-5463. DOI: 10.1093/bib/bby061. URL: https://academic.oup.com/bib/article/20/5/1878/5062947.

[76] Christian Ritz, Andrew R. Kniss, and Jens C. Streibig. "Research Methods in Weed Science: Statistics." In: *Weed Science* 63 (SP1 Feb. 2015), pp. 166–187. ISSN: 0043-1745. DOI: 10.1614/WS-D-13-00159.1. URL: https://www.cambridge.org/core/product/identifier/S0043174500015137/type/journal_article.

[77] Huiyan Sang and Jianhua Z. Huang. "A full scale approximation of covariance functions for large spatial data sets." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74 (1 Jan. 2012), pp. 111–132. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2011.01007.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2011.01007.x.

[78] Steven S. Seefeldt, Jens Erik Jensen, and E. Patrick Fuerst. "Log-Logistic Analysis of Herbicide Dose-Response Relationships." In: *Weed Technology* 9 (2 June 1995), pp. 218–227. ISSN: 0890-037X. DOI: 10.1017/S0890037X00023253. URL: https://www.cambridge.org/core/product/identifier/S0890037X00023253/type/journal_article.

[79] Matthias Seeger. "Gaussian processes for machine learning." In: *International Journal of Neural Systems* 14 (02 Apr. 2004), pp. 69–106. ISSN: 0129-0657. DOI: 10.1142/S0129065704001899. URL: https://www.worldscientific.com/doi/abs/10.1142/S0129065704001899.

[80] Myungwon Seo. In: *Journal of cheminformatics* (2020), pp. 1–17.

[81] Tiara Shanie, Jadi Suprijadi, and Zulhanif. "Text grouping in patent analysis using adaptive K-means clustering algorithm." In: vol. 1827. 2017, p. 020041. DOI: 10.1063/1.4979457. URL: http://aip.scitation.org/doi/abs/10.1063/1.4979457.

[82] Zhenming Shun and Peter McCullagh. "Laplace Approximation of High Dimensional Integrals." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57 (4 Nov. 1995), pp. 749–760. ISSN: 00359246. DOI: 10.1111/j.2517-6161.1995.tb02060.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1995.tb02060.x.

[83] Anatolij V. Skorochod. *Lectures on the Theory of Stochastic Processes*. De Gruyter, 2019. ISBN: 9783110618167. DOI: doi:10.1515/9783110618167. URL: https://doi.org/10.1515/9783110618167.

[84] Alex J Smola and Peter L Bartlett. "Sparse greedy Gaussian process regression." In: MIT Press, 2000, pp. 598–604.

[85] Edward Snelson and Zoubin Ghahramani. "Sparse Gaussian processes using pseudo-inputs." In: MIT Press, 2005, pp. 1257–1264.

[86] Vivek Srivastava, Chandrabose Selvaraj, and Sanjeev Kumar Singh. *Chemoinformatics and QSAR*. Springer Singapore, 2021, pp. 183–212. DOI: 10.1007/978-981-33-6191-1_10.

[87] S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. "Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity." In: *Bioinformatics* 21 (Suppl 1 June 2005), pp. i359–i368. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti1055. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti1055.

[88] R Core Team. *R: A Language and Environment for Statistical Computing*. 2017. URL: https://www.R-project.org/.

[89] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-15. 2019. URL: https://CRAN.R-project.org/package=rpart.

[90] Fidel Toldrá. *Advances in food and nutrition research*. Vol. 87. Academic Press, 2019.

[91] Eric-Jan Wagenmakers and Simon Farrell. "AIC model selection using Akaike weights." In: *Psychonomic Bulletin & Review* 11 (1 Feb. 2004), pp. 192–196. ISSN: 1069-9384. DOI: 10.3758/BF03206482. URL: http://link.springer.com/10.3758/BF03206482.

[92] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: http://ggplot2.org.

[93] Simon N Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2017. ISBN: 9781498728331.

[94] Yu Xue, Haokai Zhu, Jiayu Liang, and Adam Slowik. "Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification." In: *Knowledge-Based Systems* 227 (Sept. 2021), p. 107218. ISSN: 09507051. DOI: 10.1016/j.knosys.2021.107218. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705121004809.

[95] B Zagidullin, Z Wang, Y Guan, E Pitkänen, and J Tang. "Comparative analysis of molecular fingerprints in prediction of drug combination effects." In: *Briefings in Bioinformatics* 22 (6 Nov. 2021), pp. 1–15. ISSN: 1467-5463. DOI: 10.1093/bib/bbab291. URL: https://academic.oup.com/bib/article/doi/10.1093/bib/bbab291/6353238.

[96] Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. "From machine learning to deep learning: progress in machine intelligence for rational drug discovery." In: *Drug Discovery Today* 22 (11 Nov. 2017), pp. 1680–1685. ISSN: 13596446. DOI: 10.1016/j.drudis.2017.08.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S1359644616304366.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Final Version as of January 22, 2023 (classicthesis v4.6).