Parameter Inference Using Probabilistic Techniques

Rendani Mbuvha

Supervisor: Prof Tshilidzi Marwala

Co-supervisor: Dr Ilyes Boulkaibet

Department of Electrical and Electronic Engineering

University of Johannesburg

A thesis submitted at the Faculty of Engineering and Built Environment in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Electrical and Electronic Engineering

August 2021

To my family

"...as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns — the ones we don't know we don't know ...."

Donald Rumsfeld

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Rendani Mbuvha
August 2021

Acknowledgements

I firstly express great gratitude to my supervisor Prof. Tshilidzi Marwala for inspiring and encouraging me through this challenging yet fruitful journey. I am grateful for his constant motivation to perform globally relevant research that endeavours to address the challenges of our time.

I also express my gratitude to Dr Ilyes Boulkaibet for his encouragement and support at the initial stages of this work.

I am blessed to be supported by my loving wife Aluwani, a great researcher in her own right, who has personally proofread a majority of my papers with constructive critique. The numerous discussions and debates we have had on experimental design and analysis have contributed immensely to this manuscript.

I am thankful to my parents (Dr Tshifhiwa and Mr Rashaka Mbuvha), my siblings (Khuliso, Dakalo and Rabelani Mbuvha) and grandparents, who have always supported me and believed in my life endeavours.

My gratitude also goes to my collaborators Wilson Mongwe, Samuel Cohen and Prof Marc Deisenroth - who have walked parts of this journey with me.

I would like to thank Google for supporting three years of my PhD with the Google Africa PhD Fellowship in Machine Learning.

I also thank the Centre For High-Performance Computing at the Council for Scientific and Industrial Research South Africa for providing the computational resources that make this work possible.

Above all, I thank the Lord GOD Almighty, who has kept me and brought me this far.

Abstract

Complex non-linear prediction systems have become ubiquitous in numerous decision making and other socio-technical systems. In recent years, the increased adoption and use of these complex non-linear systems has been dominated by universal approximators such as neural networks and Gaussian Processes. These systems' applications span a large number of critical domains, including transportation, drug design, law enforcement, financial services, energy planning, and pandemic forecasting.

The aforementioned critical nature of the application domains necessitates the need to study the inference methods for training or calibration of these systems' parameters. Further to this, inference methods coupled with estimators of the uncertainty around the system's predictions and measures of the relative influence of its inputs aid in managing the very high societal risks associated with incorrect predictions. This thesis investigates probabilistic parameter inference methods that provide both the required uncertainty and relevance measures.

We first introduce Metropolis Hastings (MH) and Hybrid Monte Carlo (HMC) methods for parameter inference in Bayesian Neural Networks (BNNs) with applications in credit risk modelling and South African wind energy resource planning.

We further utilise a Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) method for the first time in the inference of BNN parameters. S2HMC addresses traditional MCMC methods' discretisation constraints by using a perturbed Hamiltonian, which is conserved at a higher-order by the numerical integration scheme. Experimental results on wind energy and credit datasets find that S2HMC yields higher effective sample sizes than the competing Hybrid Monte Carlo (HMC). The predictive performance of S2HMC and HMC based BNNs is found to be similar.

We thirdly perform hierarchical inference for BNN parameters by combining the S2HMC sampler with Gibbs sampling of hyperparameters for Automatic Relevance Determination (ARD). A generalisable ARD committee framework is introduced to synthesise various sampler's ARD outputs into robust feature selections. Experimental results show that this ARD committee approach selects features of high predictive information value. Further, the results show that dimensionality reduction performed through this approach improves

the sampling performance of samplers which suffer from random walk behaviour such as Metropolis-Hastings (MH).

The thesis also addresses predictive distribution calibration pathologies of the existing product of Gaussian Process expert models. We introduce a solution to the predictive dominance of uninformed experts through expert combination via the Wasserstein Barycenter and sparsity control through tempered softmax weightings. These proposals are empirically shown to outperform other product of experts (PoE) methods. The proposed PoE are also found to outperform BNNs on wind speed forecasting regression tasks.

Finally, the thesis provides a Bayesian inference approach to change point determination in the spreading rates of the novel Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) in South Africa. This approach is a first in literature, probabilistically principled method for quantifying the relative efficacy of the various South African government interventions to slow the spread of SARS-CoV-2.

Keywords:
Bayesian Neural Networks; Markov Chain Monte Carlo; Separable Hamiltonian; Shadow Hybrid Monte Carlo; Automatic Relevance Determination; Gaussian Process; Products of Experts; Wasserstein Barycenter; Compartmental Models; SIR; SEIR.

Contents

List of Figures

UNIVERSITY
OF
JOHANNESBURG

List of Tables

Nomenclature

Acronyms / Abbreviations

ANFIS Adaptive Neuro-Fuzzy Inference System

ANN Artificial Neural Network

AUC Area Under the Receiver Operating Characteristic Curve

BCM Bayesian Committee Machine

BNN Baysesian Neural Network

ESS Effective Sample Size

GA Genetic Algorithm

GA-PSO Genetic Algorithm with PSO crossover

GA-PSO-I Genetic Algorithm with Particle Swam Optimisation initialisation

GP Gaussian Process

gPoE Generalised Product of Experts

HMC Hybrid Monte Carlo

MCMC Makrov Chain Monte Carlo

MH Metropolis Hastings Algorithm

MLP Multilayer Perceptron

NN Neural Network

PoE Product of Experts

PSO Particle Swarm Optimisation

rBCM Robust Bayesian Committee Machine

RMSE Root Mean Square Error

S2HMC Separable Shadow Hamiltonian Hybrid Monte Carlo

SEIR Susceptible-Exposed-Infected-Recovered

SIR Susceptible-Infected-Recovered

SVGP Sparse Variational Gaussian Process

WASA Wind Atlas for South Africa

Chapter 1

Introduction

Complex non-linear systems, in particular machine learning models, have become highly prevalent in numerous aspects of modern-day life. Applications driven by machine learning models include *inter alia* health care, financial services, transportation and energy [80, 86, 85, 6, 73].

Applications that utilise machine learning models rely on training a learning machine that encodes the relationship between the inputs and outputs of the specific phenomenon being modelled. These input-output relationships are extracted from data generated by the phenomenon [70]. Such learning machines leverage non-linear relationships that were typically considered computationally prohibitive [81]. Recent advances in microprocessor, graphical processor, and data storage technology have increased computational efficiency that allows for both offline training and prediction as well as real-time-online training and prediction.

Learning machines can be parameterised to encode relationships exhibited within the data generated by a specific phenomena being modelled. Popular learning machines in literature and practice include artificial neural networks, decision trees, support vector machines and ensembles or weighted committees of each of the respective models such as random forests, bagged and boosted models[16, 56].

In the past, the predictive performance of such learning machines significantly relied on domain-specific expertise to design inputs or features that represent the signals emanating from the phenomenon that is being modelled [70]. This has significantly changed with the development of deep learning models that can learn representations, such as convolutional neural networks. A direct result of the advancements in representation learning is the increased application of learning machines in many critical domains such as X-ray analysis and autonomous driving without a domain expert's presence or supervision.

The process of training these learning machines has thus become a focal area due to the critical nature of the tasks where learning machines are deployed [45]. The training process is the process through which we obtain model parameters that are used for future predictions. In statistical literature, this process is referred to as inference. In essence, model parameters and the data used to train or infer them encapsulate both the model's predictive potential and the risk associated with incorrect predictions. The societal cost of incorrect predictions can be as high as resulting in automotive or aviation accidents, incorrect forecasts of pandemic trajectories, clinical misdiagnosis and unfair bias or prejudice [45]. Thus ideal parameter inference methods for critical applications should adequately reflect the level of uncertainty around such parameters and their associated predictions.

This thesis investigates probabilistic approaches to parameter inference that yield such required confidence estimates on both parameters and predictions. This aids in both analysis and decision making based on predictions from learning machines.

1.1 Neural Networks

Artificial Neural Networks (ANNs) are learning machines that are inspired by models of neurological processes in living organisms. Neural Networks (NN) have been widely used as universal approximators of many complex systems [86]. In this work we focus on the Multilayer Perceptron (MLP) [80]. In the MLP we aim to learn input - output mappings by propagating inputs through a sequence of hidden layers of weighted non-linear transformations [86]. We then search for an optimal set of weights that minimizes an objective function defined in terms of a distance metric between the network outputs and the true values in a training data set. Cybenko [32] showed that an MLP with a single hidden layer can approximate any function of arbitrary complexity defined in a compact domain provided it has sufficient hidden units. Figure 1.1 shows an example of an MLP similar to those considered in this work. The outputs of a regression network with a single output as depicted in Figure 1.1 are defined as follows:

$$f_k(x) = b_k + \sum_j v_{jk} h_j(x) \tag{1.1}$$

$$h_j(x) = tanh\left(a_j + \sum_i w_{ij} x_i\right) \tag{1.2}$$

Where w_{ij} is the weight connection for the i^{th} input to the j^{th} hidden unit and v_{jk} is the weight connection between the j^{th} hidden unit to the k^{th} output - in our case $k = 1$. The activation function $tanh$ provides the non-linearity required to approximate complex

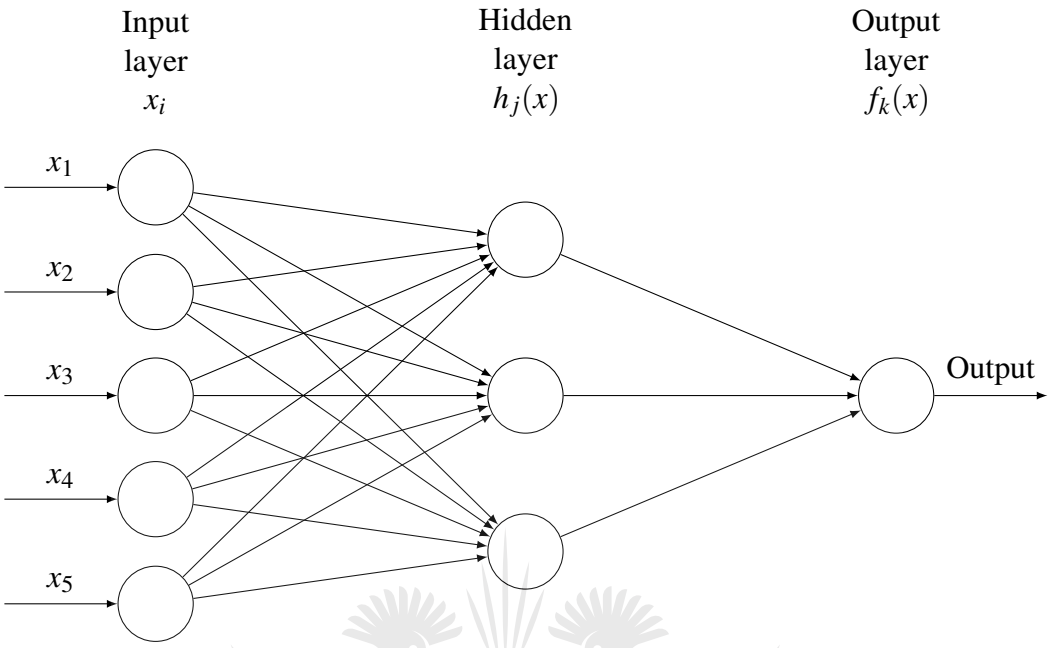


Figure 1.1 An example of a feed-forward MLP with five inputs, three hidden nodes and one output. The inputs are fed-forward such that values of the units in a particular layer are a non-linear transformation of a weighted sum of those in the preceding layer [86]

non-linear relationships. There are a variety of common activation functions in literature. These include functions such as the sigmoid and rectified linear unit (ReLU) [79].

In deriving the input-output approximation of the MLP, network weights and biases are tuned to minimise the errors in mapping outputs of a training dataset. The weights, therefore, can be considered as an encoding of such an input-output relationship. This error minimisation on a dataset $D = \{x^{(i)}, t^{(i)}\}$ is defined by the equation [49]:

$$E_D = \frac{1}{2}\sum_{i=1}^{N}\left(t^{(i)} - y(X^{(i)}; w)\right)^2 \tag{1.3}$$

In the classification case the error is defined by the cross-entropy between the output and the target vectors. On a data set $D = \{\mathbf{x_i}, \mathbf{t_i}\}$ with $i = 1, ..., N$, with K classes, the total network error E_D is defined by:

$$E_D = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_{ik} \log y_{ik} \tag{1.4}$$

The output layer activation function in classification networks is often in the form of a softmax function:

$$y_k = \frac{e^{y_k^{in}}}{\sum_k^{\text{nclass}} e^{y_k^{in}}} \tag{1.5}$$

1.2 Methods of Parameter Inference

There exists a myriad of methods in literature for performing inference of model parameters. We discuss the main ideas around these methods below.

1.2.1 Gradient Descent Methods

A natural approach to solve the inverse problem of parameter inference is the use of gradient-based optimisation. A gradient descent approach aims to find an optimal set of parameters that minimises a learning machine's error based on a specified distance metric between a learning machine's outputs and real-world outcomes or targets.

Parameters of learning machines are generally high dimensional with a limited allowance for closed-form solutions. Gradient descent methods minimise a learning machine's error by iteratively updating its parameters in the opposite direction of the gradient of its error function [106]. The general structure of gradient descent updates that sequentially step through the error surface is defined by equation 1.6.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial E_D(\mathbf{w}_t)}{\partial \mathbf{w}} \tag{1.6}$$

The size of the jumps in gradient descent is regulated by the learning rate η. This update process is repeated until a convergence criterion is met or a pre-specified maximum number of iterations is reached.

There are variations and extensions of gradient descent in literature [106, 16]. These include stochastic gradient descent that randomly shuffles the training data and adapts parameters one example at a time. The addition of momentum parameters is also common to allow for acceleration on rugged error surfaces [96]. Variations that employ adaptive learning rates such as Adam [68], Adagrad [39] and RMSProp are also commonly used in practice.

1.2.2 Meta-heuristic Methods

Meta-heuristic methods are generalised algorithmic structures that simulate metaphoric observations in optimisation or search procedures occurring in biological systems, physics,

and population dynamics [1]. Popular meta-heuristic methods include the genetic algorithm (GA) and Particle Swarm Optimisation (PSO).

Genetic Algorithm (GA)

The biological process of natural selection inspires the GA. In GA, candidate solutions are individuals within the population. Each individual's fitness is evaluated based on a cost function. At each iteration, three evolutionary procedures, selection, crossover and mutation, are executed to obtain a global minimum. First, certain individuals are sampled from the population (the selection step) for a crossover where parts of the selected individuals are randomly exchanged. Next, another set of randomly selected individuals are also mutated. In my earlier work [85], I use a continuous GA to optimise the parameters of an Adaptive Neuro-Fuzzy Inference System (ANFIS) model where the mutation is performed by adding Gaussian noise to randomly selected parts of the vector of the unknown parameters. The process continues until the algorithm converges or a specified maximum number of iterations is executed.

Particle Swarm Optimisation (PSO)

The PSO is one of the most recognised meta-heuristic optimisation algorithms that is inspired by the natural process of flocking birds searching for food. In the PSO algorithm, each particle in a swarm is considered a candidate solution for the optimisation problem. In this algorithm, each particle is updated in each iteration using the following equation [85]:

$$P(i+1) = P(i) + V(i+1) \tag{1.7}$$

Where $V(i+1)$ is the particle's velocity which is updated by [85]:

$$V(i+1) = w_0 V(i) + c_1 r_1 (P_{best} - P(i)) + c_2 r_2 (G_{best} - P(i)) \tag{1.8}$$

Where w_0 is the inertia weight which maintains previous velocity. c_1 is the particle's acceleration constant towards its personal best solution P_{Best}, while c_2 is the acceleration of the particle towards the best known position amongst all particles. r_1 and r_2 are randomly selected from a uniform distribution $U(0,1)$ where these two parameters are used to add randomness to the search space exploration. These updates continue until the algorithm converges or a specified maximum number of iterations is executed.

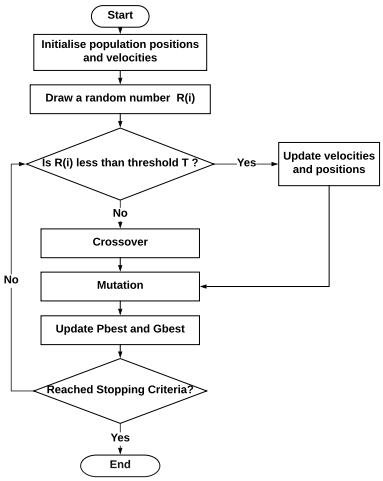


Figure 1.2 Flowchart showing the GAPSO algorithm.

Hybrid Meta-heuristic Approaches

Generally, the GA's main issue is the lack of memory since the information contained by the candidate solution that has not been selected for a crossover (or mutation) may be lost to future generations [48]. In my earlier work [85], the two-hybrid methods between GA and PSO below are proposed such that the GA can be further improved by the memory and social learning elements of the PSO algorithm.

1. **GA with PSO Crossover (GA-PSO)** Here we adapt the GA crossover by probabilistically alternating the standard GA crossover and the PSO velocity updates. This algorithm is shown in Figure 1.2.

2. **GA with PSO initialisation (GAPSO-I)** Here we run the PSO algorithm for a limited number of iterations. Simultaneously, the best particle obtained by the PSO is used as one of the individuals that initialise the GA population. This is most similar to the algorithm proposed by Yu et al. [136]. However, we use only one particle from the PSO rather than all the M best particles in the GA initialisation; using just the best particle from the PSO with other random initialisations increases the search space of the GA, while using M particles could localise the search at very early stages.

1.2.3 Bayesian Inference

The Bayesian inference framework provides a unifying paradigm between the observed data and the modeller's prior hypothesis on the model parameters [75]. This framework's result is

a posterior probability distribution on the model parameters [93]. This posterior distribution is governed by Bayes theorem as follows:

$$P(\mathbf{w}|D,H) = \frac{P(D|\mathbf{w},H)P(\mathbf{w})}{P(D)} \tag{1.9}$$

where $P(\mathbf{w}|D,H)$ is the posterior distribution of a vector of model parameters $(\mathbf{w})$ given the model (H) and observed data (D), $P(D|\mathbf{w},H)$ is the data likelihood and $P(D)$ is the evidence.

The Bayesian framework deviates from the other optimisation methods discussed above as they are based on maximum likelihood estimation of a single set of parameters. Machine learning models such as neural networks typically result in multi-modal loss surfaces, with multiple ridges and local minima [24]. Probabilistic exploration of such surfaces presented by the Bayesian framework thus aids the exploration of globally optimal parameters.

Prior distributions in the Bayesian framework introduce model regularisation in a principled way [74]. Regularisation is often critical for inverse problems such as parameter inference to reduce over-fitting. In practice, posterior inference is not tractable in closed form; numerous approximate inference techniques are employed to perform posterior inference, including Laplace approximation, variational inference and Markov Chain Monte Carlo.

Laplace Approximation

The Laplace approximation to the posterior utilises a localised multivariate Gaussian approximation around the mode of the posterior [74]. A second order Taylor expansion of the log posterior $\ln(P(\mathbf{w}|D,H))$ around the mode can be defined as :

$$\ln(P(\mathbf{w}|D,H) \approx \ln(P(\mathbf{w}_0|D,H) - \frac{\mathbf{A}}{2}(\mathbf{w}-\mathbf{w}_0)^2 \tag{1.10}$$

Where $\mathbf{w}_0$ is the local maximum and $\mathbf{A}$ is the Hessian evaluated at $\mathbf{w}_0$

$$A = -\frac{\partial^2}{\partial \mathbf{w}^2} \ln P(\mathbf{w}|D,H)\Big|_{\mathbf{w}=\mathbf{w}_0} \tag{1.11}$$

Since equation 1.10 is evaluated at $\mathbf{w}_0$ which is a saddle point the first term becomes zero leaving the approximate Gaussian distribution as $\mathcal{N}(\mathbf{w_0}, \mathbf{A}^{-1})$.

Variational Inference

Variational inference methods address the posterior inference problem as an optimisation problem [19]. Variational inference aims to minimise the Kullback-Leibler divergence between approximate densities $q(\mathbf{w})$ and the true posterior distribution $P(\mathbf{w}|D,M)$. The optimised approximate density is such that

$$q^*(\mathbf{w}) = \arg\min_{q(\mathbf{w})\in\mathcal{Q}} \mathrm{KL}\Big(q(\mathbf{w})||p(\mathbf{w}|D,m)\Big) \tag{1.12}$$

Where $\mathcal{Q}$ is a family of approximate densities. The Mean-field variational family of densities is often computationally convenient as it partitions elements of $\mathbf{w}$ into disjoint independent blocks with unique factors in the variational density [16, 19].

Markov Chain Monte Carlo Methods (MCMC)

Markov Chain Monte Carlo (MCMC) methods use a Markov Chain with a stationary distribution that converges to $P(\mathbf{w}|D,M)$ to draw samples from the posterior distribution, where a Markov Chain is a sequence of random variables $\mathbf{w}_t$ such that:

$$P(\mathbf{w}_{t+1}|\mathbf{w}_1,...,\mathbf{w}_t) = P(\mathbf{w}_{t+1}|\mathbf{w}_t)$$

An advantage of MCMC methods over other approximate inference methods is that they are asymptotically guaranteed to converge to the true posterior distribution [19]. These distinctions are illustrated in Figure 1.3

The Metropolis Hastings (MH) algorithm is one of the simplest algorithms for generating a Markov Chain which converges to the correct stationary distribution. The MH generates samples using a proposal distribution. A common proposal distribution is a symmetric random walk obtained by adding Gaussian noise to a previously accepted parameter state. The random walk behaviour of such a proposal typically results in low sample acceptance rates. In this thesis, we explore in detail dynamical MCMC methods that address the drawbacks of MH.

1.3 A Review of Machine Learning in Critical Tasks

This thesis considers the Bayesian model parameter inference problem in the context of three critical real-world tasks: short-term wind power forecasting, credit default modelling and later pandemic trajectory modelling.

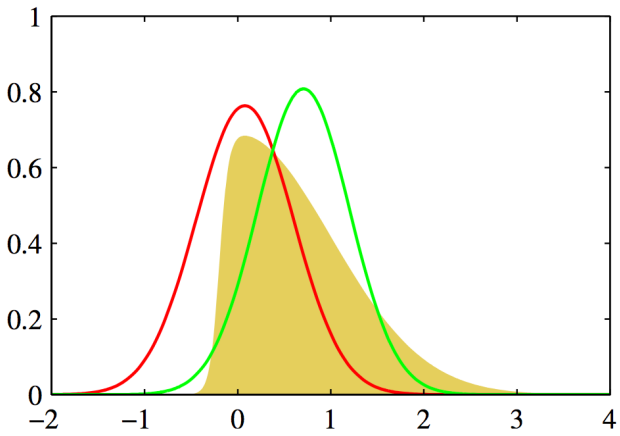


Figure 1.3 An illustration of the approximate distributions from Laplace (red), Variational (green) approximation and the true posterior distribution which is asymptotically equivalent to MCMC (mustard) [16].

1.3.1 Credit Default Modelling

Credit default risk modelling is critical to the loss management of credit portfolios for financial institutions. Effective credit risk estimation prevents and manages losses that arise when borrowers cannot make the necessary credit repayments on agreed terms. Accurate estimation of an individual's credit risk is of benefit both to the lending institution and the borrower in any credit agreement [55]. The lending institution benefits from increased profits or reduced losses while the borrower benefits by only being involved in transactions within their ability of fulfilment. Subjective expert judgement has historically been used to determine the credit risk presented by a borrower [129]. This clearly presents challenges of introducing cognitive biases and does not allow for streamlined operational efficiency in large financial institutions [129].

In recent times, increased demand for credit and the development of efficient computing systems has given prominence to sophisticated machine learning techniques in the credit risk determination process [55]. Tree-based models and artificial neural networks (ANNs) dominate literature in machine learning approaches to credit risk modelling [10, 135].

Xia et al. [134] proposed an Xtreme Gradient boosting (XGboost) tree model for credit scorecard creation. Their results show that XGboost after Bayesian parameter tuning outperforms random forests (RF) and support vector machines (SVMs) based on accuracy and the Area Under the Curve (AUC) measures on five benchmark credit datasets. Twala [124] shows that ensemble tree-based classifiers demonstrate superior predictive performance when noise is introduced to credit scoring attributes.

Sun and Vasarhelyi [118] compare deep neural networks (DNNs) and simple ANNs in predicting credit default on a Brazilian Banking Dataset. Their results show that simple ANNs and DNNs outperform logistic regression and tree-based methods on the AUC performance measure. The work of Angelini et al. [6] similarly finds that low classification errors can be obtained when using ANNs to predict credit default for Italian Small businesses. Yeh and Lien [135] use ANNs to predict credit card default for Taiwanese credit cardholders. Their results show that ANNs produced the lowest errors when predicting the probability of default than K-nearest neighbours, logistic regression and decision trees. The work of Hamori et al. [54] shows that the performance of ANNs in credit default modelling is significantly affected by choice of activation function and the dropout mechanism employed. Baesens et al. [10] benchmarks classification algorithms on eight benchmark data sets with several performance measures. Their overall results show that Hill Climbing Ensemble Selection with bootstrapping method outperformed other methods with ANNs performing second.

While ANN methods are well covered in the literature, there has been no attempt to address some of their shortcomings regarding their applications to credit risk modelling. These shortcomings include the fact that traditional ANNs do not give an indication of which attributes are relevant for the prediction of credit risk, and this does not aid in the transparency of the credit granting process, which might be required by regulations such as General Data Protection Regulation (GDPR)'s 'right to explanation' in the European Union [50]. The second drawback which applies to both ANNs and tree-based models is that they give the probability of default but do not address the level of uncertainty behind such a probability - this does not aid in the reliability analysis and risk appetite assessment of the lending institution. In this work, we develop probabilistic formulations of ANNs that will address the two shortcomings identified above.

1.3.2 Short Term Wind Power Forecasting

Climate change and the reduction of greenhouse gas emissions have become central items on the global sustainability agenda. This has culminated in the Paris climate accord of 2015 between over 192 state parties [125]. The agreements, amongst other things, commit these states to a just transition from fossil fuels to renewable energy sources such as wind and solar [86].

The main reservation around large-scale wind energy adoption is its intermittency as energy production is directly dependent on uncertain future atmospheric conditions [40]. Forecasting of short term wind energy production has thus become critical to operations management and planning for electricity suppliers [82]. Such forecasts can then be used

for proactive reserve management and energy market trading to ensure that electricity load demands are optimally met [82, 100].

Statistical and machine learning methods have become increasingly prominent in wind power forecasting. These are based on refining predictions from Numerical Weather Predictions (NWP) into localised predictions for the wind farms in question. Eseye et al. [42] propose a two-stage Adaptive Neuro-Fuzzy Inference System (ANFIS) with the first stage refining the NWP winds speeds. In contrast, the second stage uses the refined wind speed estimate for wind power prediction. ANNs are used by Eseye et al. [41] for day ahead predictions trained based on both observed wind speed data and NWP windspeed data for wind speed predictions. A combination of Radial Basis Function (RBF) NNs and fuzzy inference is employed by Sideratos and Hatziargyriou [111] with significant improvement over baselines. Fugon et al. [44] uses random forest models trained on NWP wind speed and direction forecasts to outperform NNs, linear regression and other ensemble decision trees on prediction horizons of 1 to 60 hours. Daniel et al. [33] compares ANNs, boosted decision trees and generalised additive models for wind speed forecasting in South Africa. They find significant outperformance by ANNs with additional improvements given by forecast combination methods.

Mbuvha et al. [86] uses a Laplace approximation Bayesian Neural Network (BNN) to forecast wind power in a Norwegian wind farm. Mbuvha et al. [84] further shows that BNNs significantly outperform MLPs trained by maximum likelihood and are capable of identifying relevant inputs for prediction. In this thesis, we further explore the idea of BNNs in wind power prediction with parameter inference using more theoretically accurate MCMC methods.

I will now illustrate the efficacy of using NNs over other popular models such as ANFIS for this task using my earlier work in Mbuvha et al. [85]. The details of the workings of ANFIS are outlined in Appendix A.

The relative performance of the proposed hybrid methods meta-heuristic methods discussed in Subsection 1.2.2 is investigated when training an ANFIS for predicting one-hour ahead wind power production. I use the Norwegian Wind Farm dataset described in Appendix A.3. The data is split date wise into 70% for training and 30% for testing. The model performance is evaluated based on Root Mean Square Error (RMSE) as defined in equation 2.26.

An ANFIS with 3 Gaussian membership functions (MFs) for each input was trained using a) Hybrid backpropagation least squares (BP-LS) method of Jang [65], b) The normal GA [28], c) The proposed GAPSO, d) The GAPSO-I, e) An MLP trained by backpropagation.

To allow for the stochastic effects of random initialisation, each algorithm's training is repeated 30 times. This also allows us to perform statistical significance tests on the results using a non-parametric Kruskal-Wallis (KW) [89] and post-hoc bonferorni [62] test for pairwise comparisons. An MLP with three hidden neurons is also trained for comparison purposes.

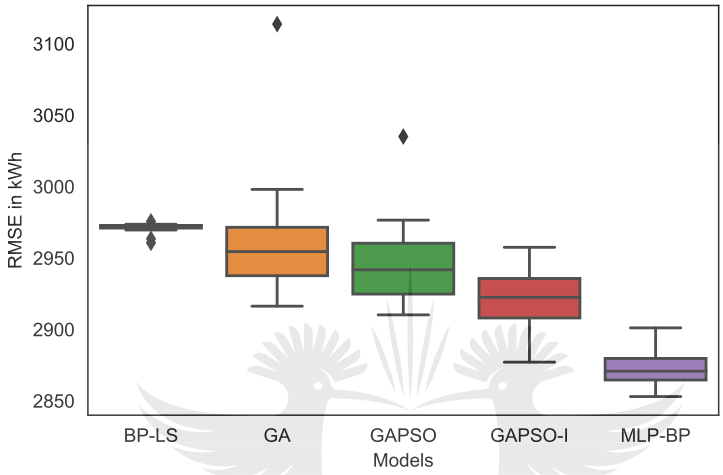


Figure 1.4 Boxplot showing the distribution of testing RMSE from the thirty trials of each method

The boxplot in Figure 1.4 shows the distribution of testing RMSE from the different models. It can be seen from the plot that an MLP trained by backpropagation shows the lowest mean RMSE of 2870.30 kWh. This is followed by an ANFIS trained using the proposed GAPSO-I method.

The results also show that the RMSE for the evolutionary techniques has greater variation than the BP-LS. This is because there are more stochastic elements in the algorithms than just in the initialisation of the backpropagation-based methods.

Table 1.1 Results showing the mean RMSE from 30 trails for one-hour ahead wind power prediction.

Model	**Mean Training RMSE (kWh)**	**Mean Testing RMSE (kWh)**
ANFIS BP-LS	2955.34	2971.57
ANFIS GA	3012.07	2959.54
ANFIS GAPSO	2998.24	2941.02
ANFIS GAPSO-I	2989.17	2919.60
MLP BP	2865.50	2870.30

A KW statistical test performed on the testing RMSE gives a p-value of $2.2298e{-}22$, indicating that the differences in model performance are statistically significant. A further Bonferroni test for pair-wise differences shows that the difference between the testing RMSE from the evolutionary techniques and the BP-LS is statistically significant in all cases at an acceptance level of $\alpha = 0.05$. The Bonferroni test also showed that the MLP-BP had a statistically significant lower RMSE than all the ANFIS methods.

We note in these illustrative results that the MLP outperformed ANFIS under numerous training regimes in a statistically significant manner. This result is consistent with Şahin and Erol [108], Sobhani et al. [115], Zamani et al. [138] and Armaghani and Asteris [8]. Other reasons why ANN-based models can be preferred relative to ANFIS include; the flexibility in architecture (the possibility of Deep Neural Networks), continuity of loss functions, and the active research community's general size.

1.4 Thesis Scope

This thesis is aimed at probabilistic parameter inference using MCMC methods in predictive models with a focus on neural networks. This thesis has the following limitations:

- Applications are limited to credit default modelling, wind power forecasting and COVID-19 forecasting.
- Only dynamical MCMC methods with separable Hamiltonians are considered.

1.5 Contributions of this Thesis

The issues of probabilistic parameter inference span numerous model types and many related applications. The contributions in this work transcend both algorithm theory and application spheres. More precisely, the contributions of this work are as follows;

- Firstly, this work provides a first implementation and evaluation of the Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) of Sweet et al. [119] in the inference of Bayesian Neural Network (BNN) parameters. S2HMC addresses the deterioration of acceptance rates and effective sample sizes in MCMC algorithms.
- Secondly, This work combines automatic relevance determination (ARD) using Gibbs sampling and S2HMC to further infer regularisation hyperparameters on neural network inputs. A generalisable ARD committee framework is also introduced to synthesise

various sampler's ARD outputs into robust feature selections. This framework can be easily adapted to any odd number of feature selectors and relevance metrics.

- Thirdly, this work proposes novel solutions to the predictive dominance of weak experts in products of Gaussian Process (GP) experts models (PoEs). Existing pathologies of PoEs are addressed via prediction aggregation using a Wasserstein barycenter and sparsity control through tempered softmax weightings.

- Fourth, this work utilises MCMC methods to infer change points in the spreading rates of the novel coronavirus (COVID-19) in South Africa. This contribution addresses essential societal questions on the relative efficacy of state-led non-pharmaceutical interventions (NPIs) in South Africa.

- It is also important to note that this work presents a first in literature application of BNNs and PoEs on the Wind Atlas for South Africa (WASA) datasets.

1.6 Publications

The following peer reviewed publications and preprints were published in the period corresponding to the duration of this study:

1.6.1 Conference Proceedings

- **Mbuvha, R.**, Boulkaibet, I., Marwala, T., & de Lima Neto, F. B. (2018, June). A hybrid GA-PSO adaptive neuro-fuzzy inference system for short-term wind power prediction. *In International Conference on Swarm Intelligence* (pp. 498-506). Springer, Cham.

- **Mbuvha, R.**, Boulkaibet, I., & Marwala, T. (2019, September). Bayesian automatic relevance determination for feature selection in credit default modelling. *In International Conference on Artificial Neural Networks* (pp. 420-425). Springer, Cham.

- Cohen, S., **Mbuvha, R.**, Marwala, T., & Deisenroth, M. (2020, November). Healing products of Gaussian process experts. *In International Conference on Machine Learning* (pp. 2068-2077). PMLR.

1.6.2 Journals

- **Mbuvha, R**, Marwala,T (2020). Bayesian inference of COVID-19 spreading rates in South Africa. *PloS one* 15(8): e0237126.
- Daniel, L. O., Sigauke, C., Chibaya, C., & **Mbuvha, R.** (2020). Short-Term Wind Speed Forecasting Using Statistical and Machine Learning Methods. *Algorithms*, 13(6), 132.
- Mutavhatsindi, T., Sigauke, C., & **Mbuvha, R.** (2020). Forecasting Hourly Global Horizontal Solar Irradiance in South Africa Using Machine Learning Models. *IEEE Access*, 8, 198872-198885.
- Ngwenduna, K. S., & **Mbuvha, R.** (2021). Alleviating Class Imbalance in Actuarial Applications Using Generative Adversarial Networks. *Risks*, 9(3), 49.
- Mongwe, W. T., **Mbuvha, R.**, & Marwala, T. (in press). Antithetic Magnetic And Shadow Hamiltonian Monte Carlo. *IEEE Access*.

1.6.3 Preprints

- **Mbuvha, R.**, Boulkaibet, I., & Marwala, T. (2019). Automatic Relevance Determination Bayesian Neural Networks for Credit Card Default Modelling. arXiv preprint arXiv:1906.06382.
- **Mbuvha, R.**, Boulkaibet, I., & Marwala, T. (2020). An Automatic Relevance Determination Prior Bayesian Neural Network for Controlled Variable Selection. arXiv preprint arXiv:2001.01765.
- **Mbuvha, R.**, & Marwala, T. (2020). On Data-Driven Management of the COVID-19 Outbreak in South Africa. medRxiv.

1.7 Outline of this Thesis

- **Chapter 2** introduces probabilistic inference in Bayesian neural networks using MCMC with applications in wind speed forecasting and credit default prediction.
- **Chapter 3** introduces Separable Shadow Hamiltonian (S2HMC) Hybrid Monte Carlo for sampling BNN posteriors.

- **Chapter 4** augments S2HMC with automatic relevance (ARD) to create a framework for robust feature selection.
- **Chapter 5** identifies current challenges with products of Gaussian Process expert models and proposes novel weighting schemes for improved prediction calibration.
- **Chapter 6** employs probabilistic inference methods in change point detection on COVID-19 spreading rates in South Africa.
- **Chapter 7** gives a conclusion and possible future improvements to this work.

Chapter 2

Markov Chain Monte Carlo in Neural Networks

This chapter provides the necessary background to the probabilistic inference of neural network parameters. Bayesian Neural Networks are first set out in Section 2.2, proceeding to classical MCMC methods for inference encompassing Metropolis Hastings MH and Hybrid Monte Carlo (HMC). Experiments in both classification and regression are then presented, followed by an analysis of results and conclusion. Parts of the work in this chapter also appears in Mbuvha et al. [84].

2.1 Introduction

The Bayesian formulation of ANNs was first proposed by MacKay [76]. MacKay [76] proposed a Laplace approximation to the posterior distribution, which uses a single multivariate Gaussian centred around the maximum posterior estimate (MAP) with co-variances defined by the Hessian of the log posterior in the vicinity of the MAP. This method has shown superior performance relative to gradient descent methods in numerous applications, including conflict analysis [69], energy consumption modelling [75] and wind power forecasting [86]. Some drawbacks of such an approximation include the inability to address multiple local minima and the need for enormous amounts of data if the approximation is to hold for highly complex models [93].

MCMC techniques are theoretically guaranteed to ergodically explore the posterior parameter space [95] thus implicitly addressing issues of multimodality. Naive MCMC techniques such as MH suffer from inefficiencies resulting from random walk behaviour.

Random walk behaviour results in highly correlated samples that make exploration of the posterior distribution extremely slow.

Dynamical MCMC methods are often used to suppress random walk behaviour exhibited by MH. Such methods use the gradient information of the negative log posterior distribution to make an efficient distant proposal with high acceptance probabilities. HMC, which was first proposed by Duane et al. [38], uses auxiliary variables to simulate the fictitious dynamics of a Hamiltonian system. Parameter distributions are obtained by marginalising over the auxiliary momentum variables. HMC was first applied as a sampling technique for Bayesian Neural Network models (BNNs) in the seminal work of Neal [93]. Since then, HMC has proved to be the most effective way to obtain such samples from the posterior distribution of BNNs, outperforming other approximate inference methods such as Gaussian approximation of MacKay [75] and variational inference of Hinton and Van Camp [61] ([131],[15]).

2.2 Bayesian Neural Networks

The NN models described in Section 1.1 can be parametrised to make probabilistic assumptions about the data [80]. This parametrisation is obtained via the Bayesian framework. MLP models trained by minimising error through backpropagation, as discussed in Section 1.1 can be seen in a frequentist light as yielding the most probable set of weights given the data, i.e. maximum likelihood estimation.

The Bayesian framework allows for the use of prior conditions on the distribution of network weights. These conditions can be defined individually or globally for all the weights in the network.

The Bayesian framework's foundation emanates from Bayes theorem, which defines the posterior distribution of parameters given the data and prior distributional assumptions. In the context of a neural network model with architecture H, weights $\mathbf{w}$, and training dataset D, Bayes theorem translates to [86, 75]:

$$P(\mathbf{w}|D,H) = \frac{P(D|\mathbf{w},H)P(\mathbf{w}|H)}{P(D|H)} \tag{2.1}$$

where $P(\mathbf{w}|D,H)$ is the posterior probability of the weights given the data and model architecture, $P(D|\mathbf{w},H)$ is the likelihood of the data given the model. $P(\mathbf{w}|H)$ is the prior probability of the weights. $P(D|H)$ is the probability of the data given the model - this is referred to as the evidence [86].

We now set out the components of a Bayesian formulation of the MLP proposed by MacKay [76].

2.2.1 The Likelihood

In the probabilistic formulation, we interpret the error function defined in equation 1.3 as the negative log-likelihood of the network noise model. On the assumption of a Gaussian noise model with a precision parameter β, the likelihood is then defined as [86]:

$$P(D|\mathbf{w},\beta,H) = \frac{1}{Z_D(\beta)}\exp(-\beta E_D) \tag{2.2}$$

$$= \frac{1}{Z_D(\beta)}\exp\left(-\frac{\beta}{2}\sum_{i=1}^{N}\left(t^{(i)} - y(X^{(i)};\mathbf{w})\right)^2\right) \tag{2.3}$$

where $Z_D(\beta)$ is a normalising constant.

2.2.2 The Prior

The prior imposes some subjective views on the distribution of the network weights. Often the prior over the weights are defined by a Gaussian with zero mean and a precision α as follows [86]:

$$P(\mathbf{w}|\alpha,H) = \frac{1}{Z_W(\alpha)}\exp(-\alpha E_W) \tag{2.4}$$

$$= \frac{1}{Z_W(\alpha)}\exp\left(-\frac{\alpha}{2}\sum_i \mathbf{w}_i^2\right) \tag{2.5}$$

Similarly $Z_W(\alpha)$ is the normalising constant and E_W can be considered as the log prior probability distribution over the weights. Since α regulates how far we would expect the weights to vary from the mean of zero, it is referred to as the regularisation or decay parameter.

2.2.3 The Posterior

The posterior distribution over the weights is then derived from equation 2.1 as [75, 86] :

$$P(\mathbf{w}|\alpha,\beta,H) = \frac{1}{Z(\alpha,\beta)}\exp(-(\alpha E_W + \beta E_D)) \tag{2.6}$$

2.2.4 Predictive Distribution

The predictive distribution is the distribution from which we make predictions of the target t^{N+1} given new inputs $X^{(N+1)}$. In the pure Bayesian sense this distribution is obtained by

integrating the output distribution over the posterior parameter distribution as follows [86]:

$$P(t^{(N+1)}|D,\alpha,\beta,\mathcal{H}) = \int P(t^{(N+1)}|\mathbf{w},\beta,\mathcal{H})P(\mathbf{w}|\alpha,\beta,\mathcal{H},D)d\mathbf{w} \tag{2.7}$$

The posterior distribution in equation 2.6 and its predictive distribution in equation 2.7 are intractable in closed form and require either approximate inference or sampling [16].

2.3 Monte Carlo Integration

Monte Carlo (MC) sampling techniques attempt to compute expectations over complex distributions. MC methods generate a large number of realisations from the posterior distribution that are representative of its distributional nature. This implies principled handling of multiple modes and their relative weightings. Theoretically, an MC prediction at a given data point $\mathbf{x}$ can be expressed by the integral over the parameters as [15]:

$$E[f_k(\mathbf{x},\mathbf{w})] = \int f_k(\mathbf{x},\mathbf{w})P(\mathbf{w}|\alpha,\beta,D)d\mathbf{w} \tag{2.8}$$

The corresponding MC estimate of this integral can be obtained using a mean over N distributional samples of parameters $\mathbf{w}^{(t)}$ as [15] :

$$E[f_k(\mathbf{x},\mathbf{w})] \approx \frac{1}{N}\sum_{t=1}^{N} f_k(\mathbf{x},\mathbf{w}^{(t)}) \tag{2.9}$$

The estimated standard error of such an estimate or the Monte Carlo standard error (MCSE) for non correlated samples can be expressed as [23]:

$$MCSE = \frac{\sigma}{\sqrt{N}} \quad \text{where } \sigma^2 = \text{var}\big(f_k(\mathbf{x},\mathbf{w})\big) \tag{2.10}$$

The challenge in arriving at these estimates becomes obtaining the samples that are accurately representative of the posterior distribution.

2.4 Markov Chain Monte Carlo

MCMC generates samples of the target posterior distribution using a Markov chain which is constructed such that its stationary distribution is the target posterior.

A Markov Chain is a sequence of random variables $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, ..., \mathbf{w}^{(n-1)}, \mathbf{w}^{(n)}$ such that

$$P(\mathbf{w}^{(n)}|\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, ..., \mathbf{w}^{(n-1)}) = P(\mathbf{w}^{(n)}|\mathbf{w}^{(n-1)}) \tag{2.11}$$

A Markov chain is fully defined by an initial distribution $P(\mathbf{w}^{(1)})$ and a transition density $P(\mathbf{w}^{(n+1)}|\mathbf{w}^{(n)})$ governing movements between states. A distribution $P^*(\mathbf{w})$ is said to be a stationary distribution of the chain if transitions $P(\mathbf{w}'|\mathbf{w})$ leave such a distribution invariant. This condition can be defined as [94]:

$$P^*(\mathbf{w}') = \int P(\mathbf{w}'|\mathbf{w})P^*(\mathbf{w})d\mathbf{w} \tag{2.12}$$

This condition can be simplified into a stronger criterion called detailed balance such that $\forall\ \mathbf{w}'$ and $\mathbf{w}$:

$$P(\mathbf{w}'|\mathbf{w})P^*(\mathbf{w}) = P(\mathbf{w}|\mathbf{w}')P^*(\mathbf{w}') \tag{2.13}$$

Detailed balance ensures reversibility as it equates the probability of moving from $\mathbf{w}$ to $\mathbf{w}'$ the probability of moving in the reverse direction. If such a Markov chain is ergodic, its stationary distribution is unique and can be attained from any initial distribution. Thus a Markov chain with a stationary distribution $P^*(\mathbf{w})$ can be used to estimate expectations with respect to $P^*(\mathbf{w})$ [93]. We now look at two methods for constructing such Markov chains, the Metropolis-Hastings algorithm (MH) and Hybrid Monte Carlo (HMC).

2.4.1 Metropolis Hastings Algorithm

The MH algorithm is one of the simplest algorithms for generating a Markov Chain which converges to the correct stationary distribution. The MH generates proposed samples using a proposal or transition distribution $P(\mathbf{w}^*|\mathbf{w})$. A new parameter state $\mathbf{w}^*$ is accepted or rejected probabilistically given the current state $\mathbf{w}$ based on the posterior likelihood ratio [23]:

$$P\big(accept\,(\mathbf{w}^*)\big) = min\left(1, \frac{P(\mathbf{w}^*)P(\mathbf{w}|\mathbf{w}^*)}{P(\mathbf{w})P(\mathbf{w}^*|\mathbf{w})}\right) \tag{2.14}$$

A common proposal distribution is a symmetric random walk obtained by adding Gaussian noise to a previously accepted parameter state which becomes known as random walk Metropolis (RwM). In RwM, the transition density is $\mathcal{N}(\mathbf{w}, \varepsilon\Sigma)$ where ε is the noise scaling constant.

When such a proposal or transition density is symmetric, it implies that $P(\mathbf{w}|\mathbf{w}^*) = P(\mathbf{w}^*|\mathbf{w})$ reducing equation 2.14 to a ratio of posterior likelihoods as follows [113]:

$$P\big(accept\,(\mathbf{w}^*)\big) = min\left(1, \frac{P(\mathbf{w}^*)}{P(\mathbf{w})}\right) \tag{2.15}$$

Random walk behaviour of such a proposal typically results in low sample acceptance rates, slow convergence to stationary distribution and highly correlated samples. The effect of random walk behaviour can be suppressed using a more intelligent proposal that leverages gradient information. Algorithm 2.1 provides a summary of MH when applied to BNNs.

Algorithm 2.1 Metropolis Hasting Algorithm for BNNs.

Data: Training dataset $\{\mathbf{X}^{(i)}, \mathbf{t}^{(i)}\}$
Result: N Samples of BNN weights $\mathbf{w}$
initialise the network weights $\mathbf{w}$
$\mathbf{w}_0 \leftarrow \mathbf{w}_{\text{init}}$
for $n \leftarrow 1$ **to** N **do**
  Generate Candidate weights $\mathbf{w}^*$ using the transition density $P(\mathbf{w}^*|\mathbf{w})$
  Metropolis Update step:
  $\mathbf{w}_n \leftarrow \mathbf{w}^*$ with probability:
  $min\left(1, \frac{P(\mathbf{w}^*)P(\mathbf{w}|\mathbf{w}^*)}{P(\mathbf{w})P(\mathbf{w}^*|\mathbf{w})}\right)$
end

2.4.2 Gibbs Sampling

Gibbs sampling, like MH, is one of the simpler MCMC techniques [94]. The Gibbs sampler partitions the multidimensional vector of parameters $\mathbb{R}^p$ into smaller blocks $\mathbb{R}^{p_1}, \mathbb{R}^{p_2}, ...\mathbb{R}^{p_m}$ such that $p_1 + p_2 + \cdots + p_m = p$ [113]. These m blocks are selected such that the conditional distributions are easier to compute in closed form. Samples are then generated from the conditional of a specific block of parameters given fixed values (immediate past samples) of the other blocks of parameters. Algorithm 2.2 shows the pseudo-code for the Gibbs sampler. Given the requirement for simplified conditionals [94], Gibbs sampling cannot be used for sampling BNN weights. However, Gibbs sampling can be used in conjunction with other MCMC samplers for hyperparameter sampling - we discus this setup in detail in Chapter 4.

Algorithm 2.2 Pseudo code for the Gibbs Sampler.

Data: a vector of parameters $\mathbf{w} = \{w_1, w_2, \ldots, w_p\}$ with p blocks
Result: N Samples of parameter vector $\mathbf{w}$
Initialise starting value $\mathbf{w}^{(0)}$
for $n \leftarrow 1$ **to** N **do**
  Draw $w_1^{(n)}$ from $P(w_1|w_2^{(n-1)}, \ldots, w_p^{(n-1)})$
  Draw $w_2^{(n)}$ from $P(w_2|w_1^{(n-1)}, w_3^{(n-1)}, w_4^{(n-1)}, \ldots, w_p^{(n-1)})$
  Draw $w_3^{(n)}$ from $P(w_3|w_1^{(n-1)}, w_2^{(n-1)}, w_4^{(n-1)}, \ldots, w_p^{(n-1)})$
  .
  .
  .
  Draw $w_p^{(n)}$ from $P(w_p|w_1^{(n-1)}, w_2^{(n-1)}, w_3^{(n-1)}, \ldots, w_{p-1}^{(n-1)})$
end

2.4.3 Hybrid Monte Carlo

HMC proposed by Duane et al. [38] reduces random walk behaviour by adding auxiliary momentum variables to the parameter space [83]. HMC creates a vector field around the current state using gradient information, which assigns the current state a trajectory towards a high probability next state [83]. The dynamical system formed by the model parameters $\mathbf{w}$ and the auxiliary momentum variables $\mathbf{p}$ is represented by the Hamiltonian $H(\mathbf{w}, \mathbf{p})$ written as follows [93, 83]:

$$H(\mathbf{w}, \mathbf{p}) = L(\mathbf{w}) + K(\mathbf{p}) \tag{2.16}$$

Where $L(\mathbf{w})$ is the negative log-likelihood of the posterior distribution in equation 1.9, also referred to as the potential energy. $K(\mathbf{p})$ is the kinetic energy defined by the kernel of a Gaussian with a covariance matrix $\mathbf{M}$ [94]:

$$K(\mathbf{p}) = \frac{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}}{2} \tag{2.17}$$

In this work we consider $\mathbf{M}$ as to the identity matrix.

The trajectory vector field is defined by considering the parameter space as a physical system that follows Hamiltonian dynamics [83]. The dynamical equations governing the trajectory of the chain are then defined by Hamiltonian equations at a fictitious time t and dimensions $i = 1, \ldots, d$ as follows [93]:

$$\frac{\partial w_i}{\partial t} = \frac{\partial H}{\partial p_i} \tag{2.18}$$

$$\frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial w_i} \tag{2.19}$$

In practical terms, the dynamical trajectory is discretised using the leapfrog integrator. In the leapfrog integrator to reach the next point in the path, we take half a step in the momentum direction, followed by a full step in the direction of the model parameters - then ending with another half step in the momentum direction.

$$p_i(t+\varepsilon/2) = p_i(t) + (\varepsilon/2)\frac{\partial H}{\partial w_i}\Big(w_i(t)\Big) \tag{2.20}$$

$$w_i(t+\varepsilon) = w_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i} \tag{2.21}$$

$$p_i(t+\varepsilon) = p_i(t+\varepsilon/2) + (\varepsilon/2)\frac{\partial H}{\partial w_i}\Big(w_i(t+\varepsilon)\Big) \tag{2.22}$$

Due to the discretising errors arising from leapfrog integration a Metropolis acceptance step is then performed in order to accept or reject the new sample proposed by the trajectory [95, 83]. In the Metropolis step the parameters proposed by the HMC trajectory $\mathbf{w}^*$ are accepted with the probability [93]:

$$P(accept) = min\left(1, \frac{P(\mathbf{w}^*|D,H)}{P(\mathbf{w}|D,H)}\right) \tag{2.23}$$

Algorithm 2.3 shows the pseudo-code for the HMC where ε is a discretisation step size. The leapfrog steps are repeated until the maximum trajectory length L is reached. The HMC algorithm has multiple parameters that require tuning for efficient sampling, such as the step size and the trajectory length [87]. In terms of trajectory length, a trajectory length that is too short leads to random walk behaviour similar to MH. While a trajectory length that is too long results in a trajectory that inefficiently traces back [63]. The step size is also a critical parameter for sampling, small step sizes are computationally inefficient leading to correlated samples and poor mixing while large step sizes compound discretisation errors leading to low acceptance rates.

2.4.4 Step Size Tuning by Dual Averaging

The issue of step size selection for HMC, MH and later Separable Hamiltonian Hybrid Monte Carlo (S2HMC) is addressed through dual averaging during multiple initial trail runs of each sampler [63]. We target a Metropolis acceptance rate δ using dual averaging updates as

Algorithm 2.3 Hybrid Monte Carlo Algorithm for BNNs.

Data: Training dataset $\{\mathbf{X}, \mathbf{t}\}$
Result: N Samples of BNN Weights $\mathbf{w}$
$\mathbf{w}_0 \leftarrow w_{\text{init}}$
for $n \leftarrow 1$ **to** N **do**
  sample the auxiliary momentum variables $\mathbf{p}$
  $\mathbf{p} \sim \mathcal{N}(0, \mathbf{M})$
  Use leapfrog steps to generate proposals for $\mathbf{w}$
  for $t \leftarrow 1$ **to** L **do**
    $\mathbf{p}(t+\varepsilon/2) \leftarrow \mathbf{p}(t) + (\varepsilon/2)\frac{\partial H}{\partial \mathbf{w}}\big(\mathbf{w}(t)\big)$
    $\mathbf{w}(t+\varepsilon) \leftarrow \mathbf{w}(t) + \varepsilon\frac{\mathbf{p}(t+\varepsilon/2)}{\mathbf{M}}$
    $\mathbf{p}(t+\varepsilon) \leftarrow \mathbf{p}(t+\varepsilon/2) + (\varepsilon/2)\frac{\partial H}{\partial w}\big(\mathbf{w}(t+\varepsilon)\big)$
  end
  Metropolis Update step:
  $(\mathbf{p}, \mathbf{w})_n \leftarrow (\mathbf{p}(L), \mathbf{w}(L))$ with probability:
  $min\big(1, \frac{P(\mathbf{w}(L)|D,H)}{P(\mathbf{w}_{(n-1)}|D,H)}\big)$
end

follows:

$$\varepsilon_{t+1} \leftarrow \mu - \frac{\sqrt{t}}{\gamma}\frac{1}{t+t_0}\sum_{i=1}^{t} H_i \tag{2.24}$$

$$\bar{\varepsilon}_{t+1} \leftarrow \eta_t \varepsilon_{t+1} + (1-\eta_t)\bar{\varepsilon}_t \tag{2.25}$$

where μ is a free parameter that ε_t gravitates to and γ controls the convergence towards μ. η_t is a decaying rate of adaptation in line with Andrieu and Thoms [5]. H_t is the difference between the target acceptance rate and the actual acceptance rate. Our dual averaging updates are such that:

$$\mathbb{E}[H_t] = \mathbb{E}[\delta - \alpha_t] = 0.$$

This has the effect of updating the step size towards the target acceptance rate δ.

2.5 Experiment Setup

MCMC methods for inference of BNN parameters are evaluated in both credit default modelling and wind speed forecasting tasks. On each dataset, HMC and MH are employed to draw 5000 samples for a BNN with a single hidden layer and 5 hidden neurons. This simple architecture was chosen based on an initial architecture search which showed reasonable

performance across datasets. As the experiments are to evaluate the relative efficacy of the samplers - the architecture can be considered as a control variable.

2.5.1 Credit Datasets

Credit datasets used in this thesis are from UCI machine learning repository [37]. These include the Taiwan credit dataset of Yeh and Lien [135], the German credit dataset and the Australian credit dataset. Table 2.1 gives a summary of each of these datasets. A description of features in the Taiwan credit dataset is given in Table 2.2. Feature descriptions for the German and Australian datasets are not shown as they are anonymised. The types of features in these datasets are client descriptive information as well as recent repayment patterns. All credit datasets are randomly split into 70% for training and 30% testing partitions.

Table 2.1 Summary information for the credit datasets.

Dataset	Features	N
Taiwan credit	24	30000
Australian credit	14	690
German credit	24	1 000

2.5.2 WASA Meteorological datasets

The wind speed datasets used in this thesis are based on meteorological observations collected from three weather stations participating in the Wind Atlas for South Africa (WASA) project [52]. The locations of the three weather stations considered in this work are indicated by the map in Figure 2.1. The stations are selected to represent wind speed patterns along the coast of South Africa. The training and testing data split for this data is based on calendar

Table 2.2 Features in the Taiwan credit dataset.

Attribute	**Attribute Name**
X1	Amount of the given credit
X2	Gender
X3	Education
X4	Marital status
X5	Age (years)
X6 - X11	History of past payment - each of the last six months
X12 - X17	Amount of bill statement - each of the last six months
X18 - X23	Amount of previous payment - each of the last six months

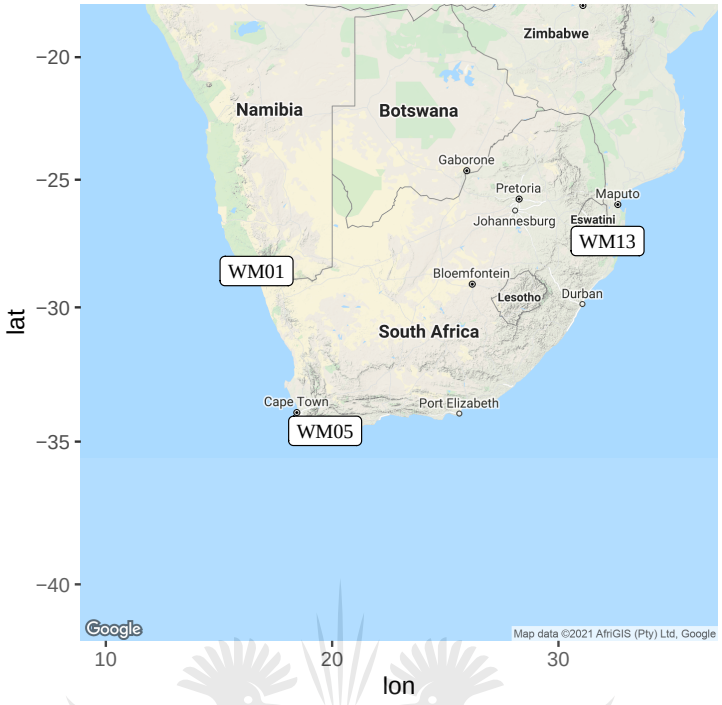


Figure 2.1 Map showing the locations of the weather stations included in the wind datasets.

dates. The training date range is 01/01/2016-31/12/2016, while the testing date range is 01/01/2017-30/06/2017 across all stations. Details of station-specific data characteristics are in Table 2.3. Observations at every station are recorded in 10-minute intervals. The date range across the stations is the same, minor differences in dataset sizes are purely because of short periods of equipment malfunction. The prediction task on this datasets is to forecast hour-ahead wind speed at the height of 62 meters given observations in the previous hour and some generalised features as described in Table 2.4. A height of 62 meters is selected as it is the typical hub height of wind turbines [33].

Table 2.3 Locations of weather stations considered for wind speed modelling.

Station Name	Features	N	Latitude	Longitude
WM01 Alexander Bay	19	78642	-28.60	16.66
WM05 Napier	19	78761	-34.61	19.69
WM13 Jozini	19	78658	-27.43	32.17

Normalisation of attributes has been shown to improve the performance of NN models [54]. Thus, each of the attributes of all the datasets above is pre-processed by projecting it onto the range $[0, 1]$ using min-max normalisation.

Table 2.4 Descriptions of features utilised in the WASA data sets. All summary statistics (mean, max, etc.) are over ten minutes intervals.

Feature	Description
WS_62_mean	Mean wind speed in m/s at 62 meters
WS_62_min	Minimum wind speed in m/s at 62 meters
WS_62_max	Maximum wind speed in m/s at 62 meters
WS_62_stdv	Standard deviation of wind speed in m/s at 62 meters
WS_60_mean	Mean wind speed in m/s at 60 meters
WS_40_mean	Mean wind speed in m/s at 40 meters
WS_20_mean	Mean wind speed in m/s at 20 meters
WS_10_mean	Mean wind speed in m/s at 10 meters
WD_60_mean	Mean wind direction (angle) in m/s at 60 meters
WD_20_mean	Mean wind direction (angle) in m/s at 20 meters
Tair_mean	Mean air temperature in degrees Celsius at 20 meters
Tgrad_mean	Mean air temperature difference between 60 meters and 10 meters
Pbaro_mean	Barometric pressure in hpa
RH_mean	Relative Humidity (%)
Hour	Previous Hour
Month	Calender Month
ws_mean_lag_1	1 hour lagged mean wind speed at 62 meters
ws_mean_lag_2	2 hour lagged mean wind speed at 62 meters
ws_mean_lag_1_day	1 day lagged mean wind speed at 62 meters

2.5.3 Performance Evaluation

The metrics detailed in this section are utilised to evaluate predictive performance and sampling performance throughout Chapters 2 to 4.

Area Under the Receiver Operating Characteristic Curve

The Area Under the Receiver Operating Characteristic Curve (AUC) is used to evaluate classification predictive performance. The AUC can be interpreted as the probability of the model correctly assigning a higher probability of default to defaulters relative to non-defaulters. The AUC is a more robust metric in imbalanced credit classification problems where datasets are often biased towards non-defaulters [6].

Root Mean Squared Error

Root Mean Square Error (RMSE) is used as the evaluation metric for regression models. RMSE is defined for observation series $\mathbf{T}$ and corresponding model prediction series $\mathbf{Y}$ as

follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_i - Y_i\right)^2} \tag{2.26}$$

Effective Sample Sizes

The main indicator of an MCMC sampler's efficiency is the auto-correlation between samples. Samplers that exhibit high correlation levels between samples result in poor mixing since multiple samples will effectively represent very similar points in the posterior distribution. Thus a larger number of samples will be required to obtain a sample set that adequately represents the entire distribution. The effective sample size (ESS) is a metric that measures the number of adequately independent samples. In this thesis, we make use of the multivariate ESS of Vats et al. [126] defined as

$$\mathbb{ESS} = N \times \left(\frac{|\Lambda|}{|\Sigma|}\right)^{\frac{1}{p}} \tag{2.27}$$

where N is the number of generated samples, p is the dimensionality of parameters, Λ is the sample covariance matrix, and Σ is the estimate of the MCSE. This metric takes into account the internal correlations between the dimensions of each sample, which would not otherwise be accounted for in univariate ESS.

2.5.4 Experimental Parameter Settings

The detailed parameter settings for the experiments are documented in table 2.5.

Table 2.5 Experimental Settings for the BNNs and Sampling runs.

Setting	Value
BNN Number of Hidden Layers	1
BNN Number of Hidden Neurons	5
BNN Activation Function	ReLu
Sampling Trajectory Length (HMC)	100
Number of Samples	5000

2.5.5 Preliminary Step Size Tuning Runs

The dual averaging method detailed in Section 2.4.4 is employed to inform the setting of step sizes for each dataset problem as step sizes are inherently dependent on the specific posterior

likelihood landscape. These preliminary runs entail 5000 samples with a target acceptance rate of 75%. Table 2.6 shows the resultant step size after these tuning runs for each dataset.

Table 2.6 Step size selections for each dataset after initial dual averaging runs.

Dataset	Step Size
Australian credit	0.038
German credit	0.026
Taiwan credit	0.004
WM01 Alexander Bay	0.161
WM05 Naiper	0.146
WM13 Jozini	0.176

More complex classification likelihoods tend to have resulted in smaller step sizes when compared to regression likelihoods. The regression datasets converged to relatively close step sizes. This is to be expected as they are of the same dimensionality and similar dataset sizes. These steps size are used for all BNNs across Chapters 2 to 4.

2.6 Results and Discussion

The results discussion and analysis in this section are based on 10 independent chains with 5000 samples for each inference method.

2.6.1 Sampling Performance

Figure 2.2 shows the negative log likelihood trace plots across all datasets. It can be seen that HMC converges at lower levels of the posterior negative log-likelihood across all datasets. This illustrates that samples drawn through HMC are more probable given the data and the prior distribution on parameters. Early convergence of HMC relative to MH can be seen from the early flattening of the trace plots.

Figure 2.3 shows the distributions of ESS across the different datasets. It can be seen from Table 2.7 the ESS values for MH are zero. This implies that random walk exploration of the posterior is highly ineffective for BNNs resulting in high auto-correlations between samples. This does not necessarily imply that ESS is zero across all dimensions as the dimensions with highest auto-correlations are significant contributors to the ESS value [126].

The relatively high sample sizes of HMC indicate the gradient information in HMC result in distant sample proposals that exhibit significantly lower auto-correlations compared to MH. The effective sample size also tends to take on low values for smaller datasets such as

Australian credit and German credit, showing that larger datasets aid in exploration of the posterior.

Table 2.7 Mean ESS statistics over ten independent chains each with 5000 samples using MH and HMC on all datasets.

Dataset	MH	HMC
Australian credit	0	2360
German credit	0	2067
Taiwan credit	0	631
WM01 Alexander Bay	0	4302
WM05 Naiper	0	3714
WM13 Jozini	0	4572
Overall Average	0	2848

The acceptance rate statistics are depicted in Table 2.8. HMC achieves significantly higher acceptance rates relative to MH across all datasets on the same dataset-specific step sizes. The high rejection rate of MH can also be seen in the relatively flat level of the negative log-likelihood from Figure 2.2, which implies little movement in the chains. The deterioration in acceptance is observed to further increase for the larger wind speed regression datasets.

Table 2.8 Mean acceptance rate (%) statistics over ten independent chains each with 5000 samples using MH and HMC on all datasets.

Dataset	MH	HMC
Australian credit	43.734	67.076
German credit	41.770	78.22
Taiwan credit	59.604	78.134
WM01 Alexander Bay	23.096	78.624
WM05 Naiper	26.05	79.516
WM13 Jozini	23.202	79.248
Overall Average	36.242	76.803

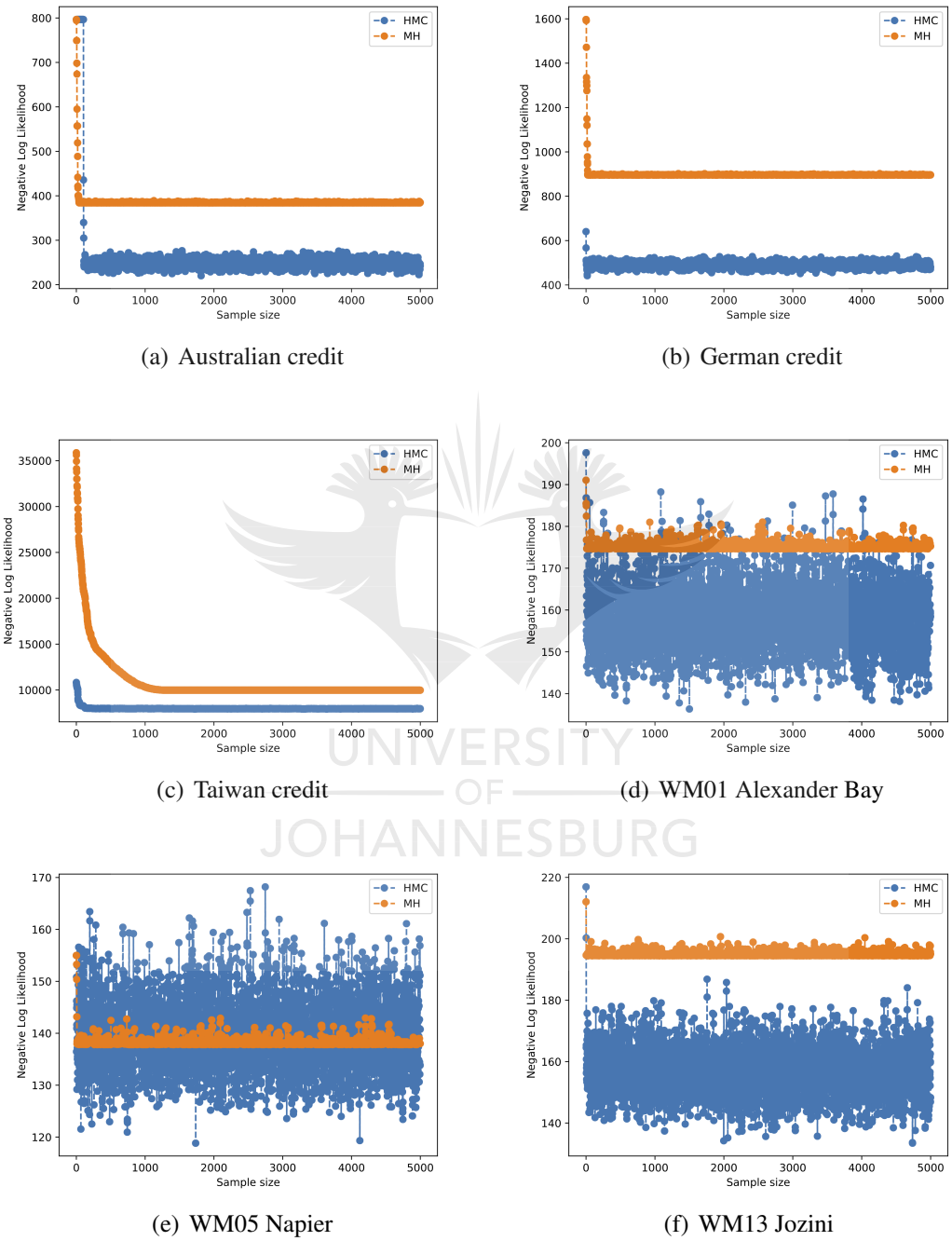
2.6.2 Predictive Performance

Predictive performance in terms of AUC for classification and RMSE for regression mirrors the sampling performance. The ROC curves in Figure 2.4 indicate superior performance of HMC over MH across all credit datasets with AUCs of 0.93 for Australian credit, 0.78 for German credit and 0.78 for Taiwan credit.

A similar phenomenon is also observed on wind speed datasets. These performance differences again point to the value of gradient information in the exploration of the posterior as the predictive performance metrics mirror the convergence levels of the negative log-likelihood in Figure 2.2.

Table 2.9 Mean testing RMSE resulting from BNNs trained using MH and HMC.

Dataset	MH	HMC
WM01 Alexander Bay	4.229	2.057
WM05 Napier	5.748	2.111
WM13 Jozini	2.934	1.856

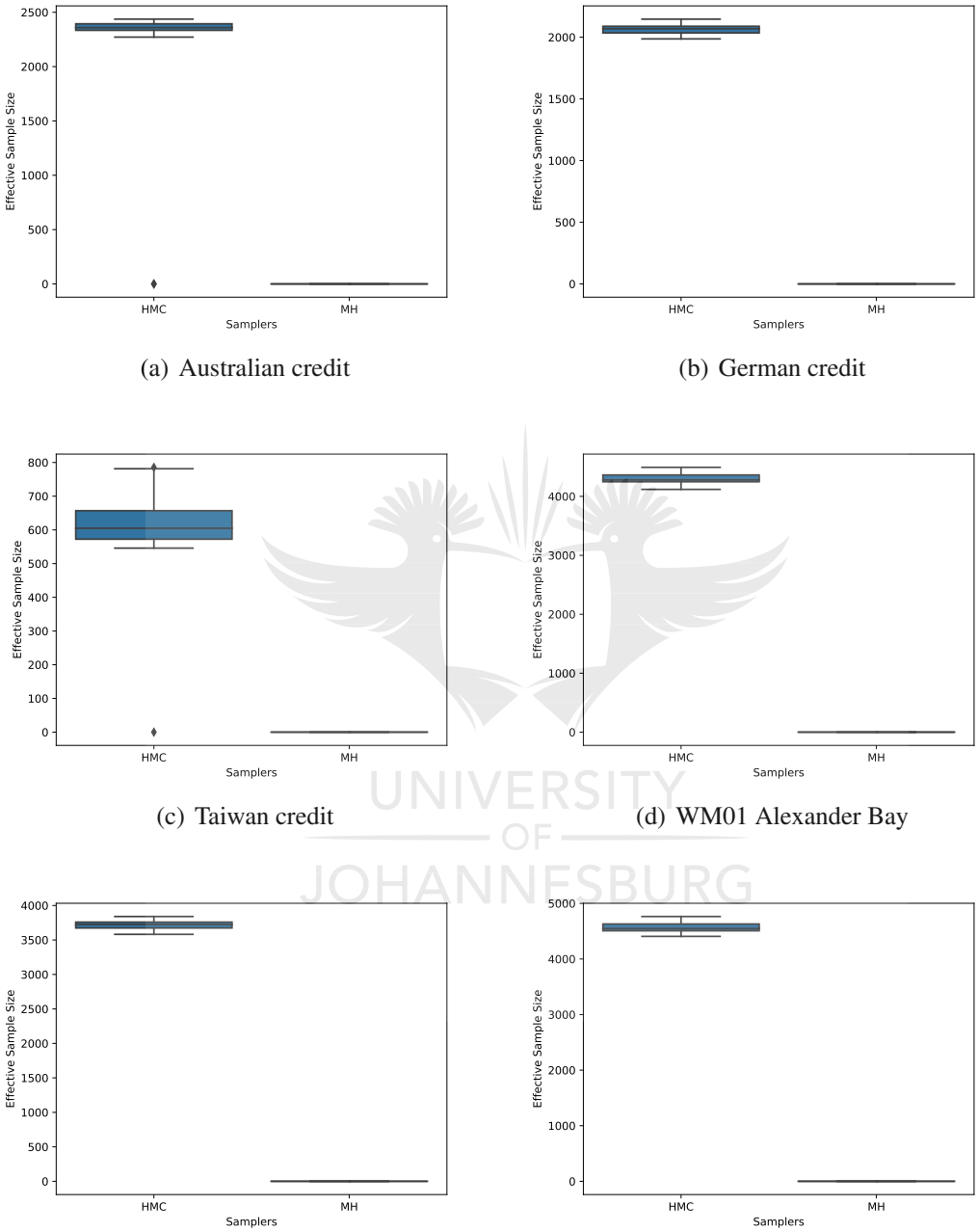


(a) Australian credit (b) German credit

(c) Taiwan credit (d) WM01 Alexander Bay

(e) WM05 Napier (f) WM13 Jozini

Figure 2.2 Negative log-likelihood trace plots for MH and HMC for all datasets.

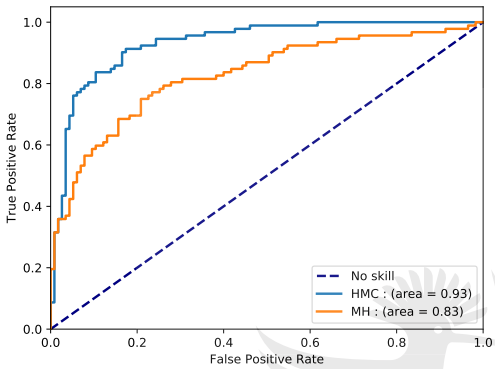


(a) Australian credit (b) German credit

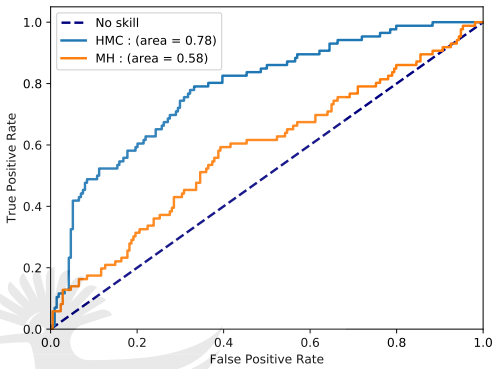
(c) Taiwan credit (d) WM01 Alexander Bay

(e) WM05 Napier (f) WM13 Jozini

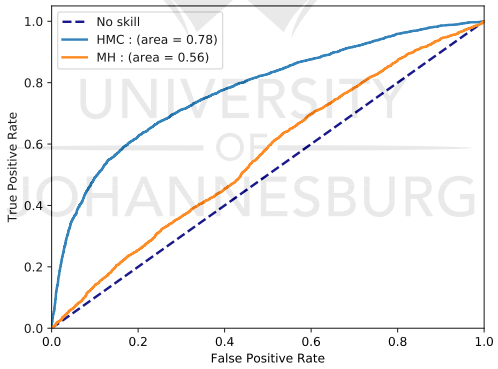
Figure 2.3 Boxplots showing the distribution of ESS over ten independent chains on each dataset.



(a) Australian credit



(b) German credit



(c) Taiwan credit

Figure 2.4 ROC curves based on mean class probabilities for the credit default datasets.

2.7 Conclusion

This chapter introduced MCMC methods with a focus on performing parameter inference for Bayesian Neural Networks. HMC and MH are introduced as algorithms for constructing Markov Chains that converge to the required posterior distribution. A dual averaging scheme for tuning the step size for HMC and variance of the random walk in MH is also introduced.

These samples are then applied in the inference of BNN parameters for credit default and wind speed modelling. The results show that HMC outperforms MH in both sampling efficiency and predictive performance. This finding demonstrates that gradient information added by the Hamiltonian dynamics is essential in the efficient exploration of the target distribution.

In the next chapter, we explore an enhancement of HMC to reduce discretisation error by sampling from an importance distribution obtained from a shadow Hamiltonian. This increases the acceptance of distant proposals, thus further improving ESS statistics.

Chapter 3

Separable Shadow Hamiltonian Monte Carlo

HMC has been widely applied to numerous posterior inference problems. A significant limitation to the increased adoption of HMC in inference for large scale machine learning systems, such as deep neural networks, is the exponential degradation of the acceptance rates and the corresponding effective sample sizes with increasing system size D due to numerical integration errors. In this chapter, a solution to this problem is provided by sampling from a modified or shadow Hamiltonian that is conserved to a higher-order by the leapfrog integrator more inline with the principle of conservation of energy. Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) allows for the feasibility of larger step and system sizes while maintaining high effective sample sizes.

3.1 Introduction

Predictive models of high parameter dimensionality have become the mainstay across a multitude of critical tasks such as medicine, law enforcement and self-driving automobiles [66]. The inherent importance of such applications suggests a greater emphasis on the understanding of predictive uncertainty arising from the models [84]. Bayesian Methods such as BNNs allow for principled inference of posterior distributions of model parameters. The Bayesian framework provides a principled approach to predictive uncertainty, theoretically justified interpretations of regularisation and generalisation through prior distributions.

MCMC methods are fundamental in performing inference of large scale probabilistic machine learning methods. HMC [95] and its variants have been a popular choice of inference due to its ability to suppress random walk behaviour through the use of first-

order gradient information [63]. While HMC has shown significant sampling efficiencies relative to its random walk counterparts, it still exhibits numerous pathologies that hinder its wide adoption for large scale practical applications. These include the need to tune highly sensitive parameters such as the discretisation step size ϵ and integration path length L as well as the requirement for computation of posterior likelihood gradients. Another practical issue with HMC, particularly in relation to deep neural network applications is the exponentially decreasing acceptance rates and the related effective sample sizes as the number of sampling parameters D increases due to the compounding discretisation errors of the leapfrog integration.

Recent advances in HMC literature have primarily focused on setting adaptive step sizes, path lengths [63, 131] and numerical or stochastic gradient approximations [29, 13]. However, the issue of degeneration of HMC with increases in model dimensionality or system size has received relatively little attention in machine learning literature.

Numerical integration errors which cause this degeneration in acceptance rates are analysed through modified equations that are exactly conserved by the discrete integrator [112, 53]. These modified equations can be defined by suitably truncated asymptotic expansions in the powers of the discretisation step size parameter [64, 112]. In Hamiltonian systems, such modified equations result in modified or shadow Hamiltonians which are conserved with higher-order accuracy relative to true Hamiltonian.

Since the discrete integrator produces a more accurate flow of the shadow Hamiltonian, sampling from the shadow Hamiltonian thus facilitates efficient sampling with higher acceptance rates [21, 64]. The bias in canonical averages of parameters introduced by sampling from the shadow Hamiltonian is then addressed by an importance sampling scheme based on the true Hamiltonian as target distribution.

Numerous approaches to shadow Hybrid Monte Carlo (SHMC) have been put forward such as Shadow Hybrid Monte Carlo (SHMC) [64], Targeted Shadow Hybrid Monte Carlo [3] and Mix & Match Hamiltonian Monte Carlo [102]. The computational performance of such approaches is limited by the need to either generate or partially refresh momenta to increase the probability of acceptance from a nonseparable shadow Hamiltonian [119].

In this chapter, separable shadow Hamiltonian hybrid Monte Carlo (S2HMC) [119] which employs a processed leapfrog integrator to generate momenta through a separable shadow Hamiltonian, is introduced for sampling the Hamiltonian efficiently. This work is the first such presentation of S2HMC in sampling parameters of BNNs.

3.2 Shadow Hamiltonians

Shadow or modified Hamiltonians are perturbations of the Hamiltonian that are by design exactly conserved by the numerical integrator. In the case of shadow Hamiltonian Hybrid Monte Carlo, we sample from the importance distribution defined by the shadow Hamiltonian as [53]:

$$\hat{\pi} \propto \exp\left(-\tilde{H}^{[k]}(\mathbf{w},\mathbf{p})\right) \tag{3.1}$$

Where $\tilde{H}^{[k]}$ is the shadow Hamiltonian defined using backward error analysis of the numerical integrator.

In backward error analysis the shadow Hamiltonian can then be defined by an asymptotic expansion in the powers of the discretisation step size around the Hamiltonian [53]:

$$\tilde{H} = H + \varepsilon H_2 + \varepsilon^2 H_3 + \varepsilon^3 H_4 + \dots \tag{3.2}$$

This asymptotic expansion diverges in practice, however a k^{th} order truncation of the expansion is used [53]:

$$\begin{aligned} \tilde{H}^{[k]} &= H + \varepsilon H_2 + \varepsilon^2 H_3 + \varepsilon^3 H_4 + \dots \\ &= \tilde{H} + \mathcal{O}(\varepsilon^k) \end{aligned} \tag{3.3}$$

The terms H_k can be determined by matching the corresponding components of the Taylor series in terms of ε and the expanded exact flow of the modified differential equation of the Hamiltonian. These modified equations can be proved to be Hamiltonian for symplectic integrators such as the leapfrog [53].

In this work, we focus on a fourth-order truncation of the shadow Hamiltonian under the leapfrog integrator. Since the leapfrog is second-order accurate ($\mathcal{O}^2$), the fourth-order truncation is conserved with higher accuracy ($\mathcal{O}^4$) than the true Hamiltonian.

The fourth-order shadow Hamiltonian for the leapfrog after matching coefficients from the flow and the asymptotic expansion becomes [119]:

$$\tilde{H}^{[4]} = U(\mathbf{w}) + K(\mathbf{p}) + \frac{\varepsilon^2}{12} K_{\mathbf{p}}^T U_{\mathbf{ww}} K_{\mathbf{p}} - \frac{\varepsilon^2}{24} U_{\mathbf{w}}^T K_{\mathbf{pp}} U_{\mathbf{w}} + \mathcal{O}(\varepsilon^4) \tag{3.4}$$

where $U_{\mathbf{w}}, U_{\mathbf{ww}}, K_{\mathbf{p}}$ and $K_{\mathbf{pp}}$ are Jacobians and Hessians of the potential and kinetic energies respectively.

The shadow Hamiltonian in equation 3.4 is non-separable in terms of $\mathbf{w}$ and $\mathbf{p}$, which necessitates computational expensive momenta acceptance criteria for momenta and potential tuning of additional parameters [64, 119]. This additional computational overhead

is relatively reduced by pre and post processing positions and momenta before and after propagating through the integrator [119].

3.2.1 Separable Shadow Hamiltonian Hybrid Monte Carlo

Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) [119] utilises a processed leapfrog integrator to create a separable Hamiltonian. The separable Hamiltonian in S2HMC is [119]:

$$\tilde{H}(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + K(\mathbf{p}) + \frac{\varepsilon^2}{24} U_{\mathbf{w}}^T M^{-1} U_{\mathbf{w}} + \mathcal{O}(\varepsilon^4) \tag{3.5}$$

Propagation of positions and momenta on this shadow Hamiltonian is performed after performing the reversible mapping $(\hat{\mathbf{w}},\hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w},\mathbf{p})$ where $(\hat{\mathbf{w}},\hat{\mathbf{p}})$ are obtained through the following fixed point iterations [119]:

$$\hat{\mathbf{p}} = \mathbf{p} - \frac{\varepsilon}{24}\left(U_{\mathbf{w}}(\mathbf{w} + \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}}) - U_{\mathbf{w}}(\mathbf{w} - \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}})\right) \tag{3.6}$$

$$\hat{\mathbf{w}} = \mathbf{w} + \frac{\varepsilon^2 \mathbf{M}^{-1}}{24}\left(U_{\mathbf{w}}(\mathbf{w} + \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}}) + U_{\mathbf{w}}(\mathbf{w} - \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}})\right). \tag{3.7}$$

After the leapfrog is performed this mapping is reversed using post-processing the following fixed point iterations [119]:

$$\mathbf{w} = \hat{\mathbf{w}} - \frac{\varepsilon^2 \mathbf{M}^{-1}}{24}\left(U_{\mathbf{w}}(\mathbf{w} + \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}}) + U_{\mathbf{w}}(\mathbf{w} - \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}})\right) \tag{3.8}$$

$$\mathbf{p} = \hat{\mathbf{p}} + \frac{\varepsilon}{24}\left(U_{\mathbf{w}}(\mathbf{w} + \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}}) - U_{\mathbf{w}}(\mathbf{w} - \varepsilon \mathbf{M}^{-1}\hat{\mathbf{p}})\right) \tag{3.9}$$

Once the samples are obtained from S2HMC as depicted in algorithm 3.1, importance weights are calculated to allow for the use of the shadow canonical density rather than the true density. These weights are based on the differences between the true and shadow Hamiltonians as follows [64, 119]:

$$w_n = \exp(-(H(\mathbf{w},\mathbf{p}) - \hat{H}(\mathbf{w}',\mathbf{p}')) \tag{3.10}$$

Mean estimates of observables $f(\mathbf{w})$ which are functions of the parameters $\mathbf{w}$ can be computed as a weighted average.

Algorithm 3.1 Separable Shadow Hamiltonian Hybrid Monte Carlo

Data: Dataset $\{\mathbf{X}, \mathbf{y}\}$
Result: N samples of model parameters $\mathbf{w}$
initialise the network weights w

$w_0 \leftarrow w_{\text{init}}$
for $n \leftarrow 1$ **to** N **do**
  sample the auxiliary momentum variables $\mathbf{p}$
  $\mathbf{p} \sim \mathcal{N}(0, \mathbf{M})$
  calculate $\tilde{H}(\mathbf{w}, \mathbf{p})$
  Apply the pre-processing mapping $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$
  Use leapfrog steps on $(\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  for $t \leftarrow 1$ **to** L **do**
    $\hat{\mathbf{p}}(t + \varepsilon/2) \leftarrow \hat{\mathbf{p}}(t) + (\varepsilon/2) \frac{\partial \hat{H}}{\partial \hat{\mathbf{w}}} \left(\hat{\mathbf{w}}(t) \right)$
    $\hat{\mathbf{w}}(t + \varepsilon) \leftarrow \hat{\mathbf{w}}(t) + \varepsilon \frac{\hat{\mathbf{p}}(t+\varepsilon/2)}{M}$
    $\hat{\mathbf{p}}(t + \varepsilon) \leftarrow \hat{\mathbf{p}}(t + \varepsilon/2) + (\varepsilon/2) \frac{\partial \hat{H}}{\partial \hat{\mathbf{w}}} \left(\hat{\mathbf{w}}(t + \varepsilon) \right)$
  end
  Apply the post-processing mapping $(\mathbf{w}_*, \mathbf{p}_*) = \mathcal{X}^{-1}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  Calculate $\tilde{H}(\mathbf{w}_*, \mathbf{p}_*)$
  Metropolis update step: $(\mathbf{w}, \mathbf{p})_n \leftarrow (\mathbf{w}_*, \mathbf{p}_*)$ with probability:
  $min \left(1, \frac{\hat{H}(\mathbf{w}_*, \mathbf{p}_*)}{\hat{H}(\mathbf{w}, \mathbf{p})} \right)$
end

3.3 Experiment Setup

The experiment setup in terms of datasets, BNN specification, step sizes and performance measures followed in this chapter are similar to that used in Section 2.5. The main exception to the previous experiment setup is the ESS calculation for S2HMC, as it is an importance sampler. In order to account for non-uniform importance of samples, calculation of ESS for S2HMC follows the $\text{ESS}_{\text{MCMC-IS}}$ suggested by Radivojević and Akhmatskaya [102]. An initial ESS (M) is calculated using the expression for multidimensional ESS in equation 2.27. M out of the total number of samples N are randomly selected to calculate $\text{ESS}_{\text{MCMC-IS}}$ as:

$$\text{ESS}_{\text{MCMC-IS}} = \frac{\left(\sum_{m=1}^{M} w_n \right)^2}{\left(\sum_{m=1}^{M} w_n^2 \right)} \tag{3.11}$$

where w_n are the importance weights defined in equation 3.10.

3.4 Results and Discussion

3.4.1 Step Size and Dimensionality Sensitivity

As an illustrative experiment, Figure 3.1 is based on simulated linear regression data and shows the effect of increasing step size on the acceptance rates when using HMC and S2HMC. The phenomena of degradation in acceptance rates in HMC becomes increasingly pronounced at larger step sizes relative to S2HMC. Increased rejections result in repetition of samples, which in turn drives down effective samples sizes due to higher auto-correlation. Figure 3.2 is also based on simulated linear regression datasets and shows similar degeneration of acceptance rates with the number of parameters for HMC. At the same time, S2HMC maintains high acceptance rates, which lead to better mixing and ESSs.

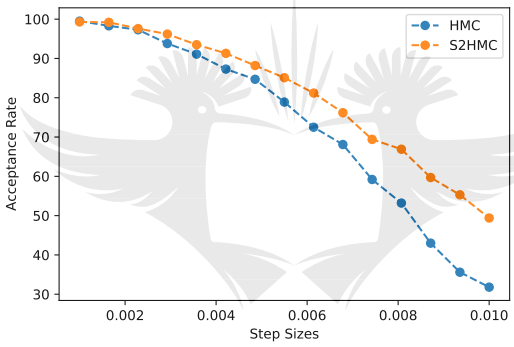


Figure 3.1 Acceptance rate degradation with step size. Simulated dataset with $N = 5000$ and $D = 100$.

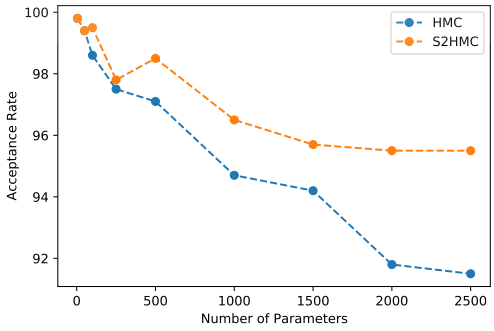


Figure 3.2 Acceptance rate degradation with number of model parameters at a constant step size of 0.01 and $N = 10000$.

3.4.2 Sampling Performance

Figure 3.3 shows the trace plots across datasets for MH, HMC and S2HMC. The rates and levels of convergence of HMC and S2HMC are almost identical as they both inherently use similar Hamiltonian dynamics in their proposals. If one looks at the negative log-likelihood as a goodness-of-fit measure, a consequence of this similarity is that these methods will have similar predictive performance in terms of RMSE and AUC for the two samplers.

The boxplots in Figure 3.4 show the distribution of ESS across 10 independent chains with the corresponding mean statistics represented in Table 3.1. It can be seen that across the datasets, S2HMC yields higher ESSs relative to HMC. This experimental result supports the modified equation theory which postulates that the shadow Hamiltonian is conserved at a higher order than the Hamiltonian itself as detailed in Section 3.2 and suggested by Izaguirre and Hampton [64], Skeel and Hardy [112] and Sweet et al. [119].

Table 3.1 Mean ESS statistics over ten independent chains each with 5000 samples using MH, HMC and S2HMC on all datasets.

Dataset	MH	HMC	S2HMC
Australian credit	0	2360	2354
German credit	0	2067	2093
Taiwan credit	0	631	642
WM01 Alexander Bay	0	4302	4352
WM05 Naiper	0	3714	3741
WM13 Jozini	0	4572	4616
Overall Average	0	2848	2914

The higher ESS statistics of S2HMC are further reinforced by the acceptance rate results in Table 3.2. This again suggests marginally higher acceptance rates for S2HMC relative to HMC at the same step size for each dataset.

3.4.3 Predictive Performance

Figure 3.5 shows the ROC curves for each sample based on the mean prediction of over 5000 samples and ten independent chains. As can be seen from the Figure, the predictive performance of HMC and S2HMC based BNNs mirrors the observations from the trace plots in Figure 3.3. The corresponding AUCs for both HMC and S2HMC are 0.93, 0,78 and 0.78 for the Australian, German and Taiwan credit datasets, respectively.

The regression results in terms of mean RMSEs in Table 3.3 indicate similar outcomes to the credit classification experiments. The performance difference between HMC and S2HMC

Table 3.2 Mean acceptance rates (%) over ten chains of 5000 samples for each sampler on all datasets.

Dataset	MH	HMC	S2HMC
Australian credit	43.734	67.076	70.290
German credit	41.770	78.22	79.260
Taiwan credit	59.604	78.134	80.690
WM01 Alexander Bay	23.096	78.624	80.942
WM05 Naiper	26.05	79.516	81.980
WM13 Jozini	23.202	79.248	81.322
Overall Average	36.242	76.803	79.087

is marginal, suggesting, as reflected in the trace plots, that their samples are relatively close geometrically. This phenomenon results in almost equal predictive performance.

Table 3.3 Mean testing RMSE resulting from BNNs trained using ten independent chains of MH, HMC and S2HMC at each of the weather stations.

Dataset	MH	HMC	S2HMC
WM01 Alexander Bay	4.22	2.057	2.056
WM05 Napier	5.748	2.111	2.108
WM13 Jozini	2.934	1.856	1.858

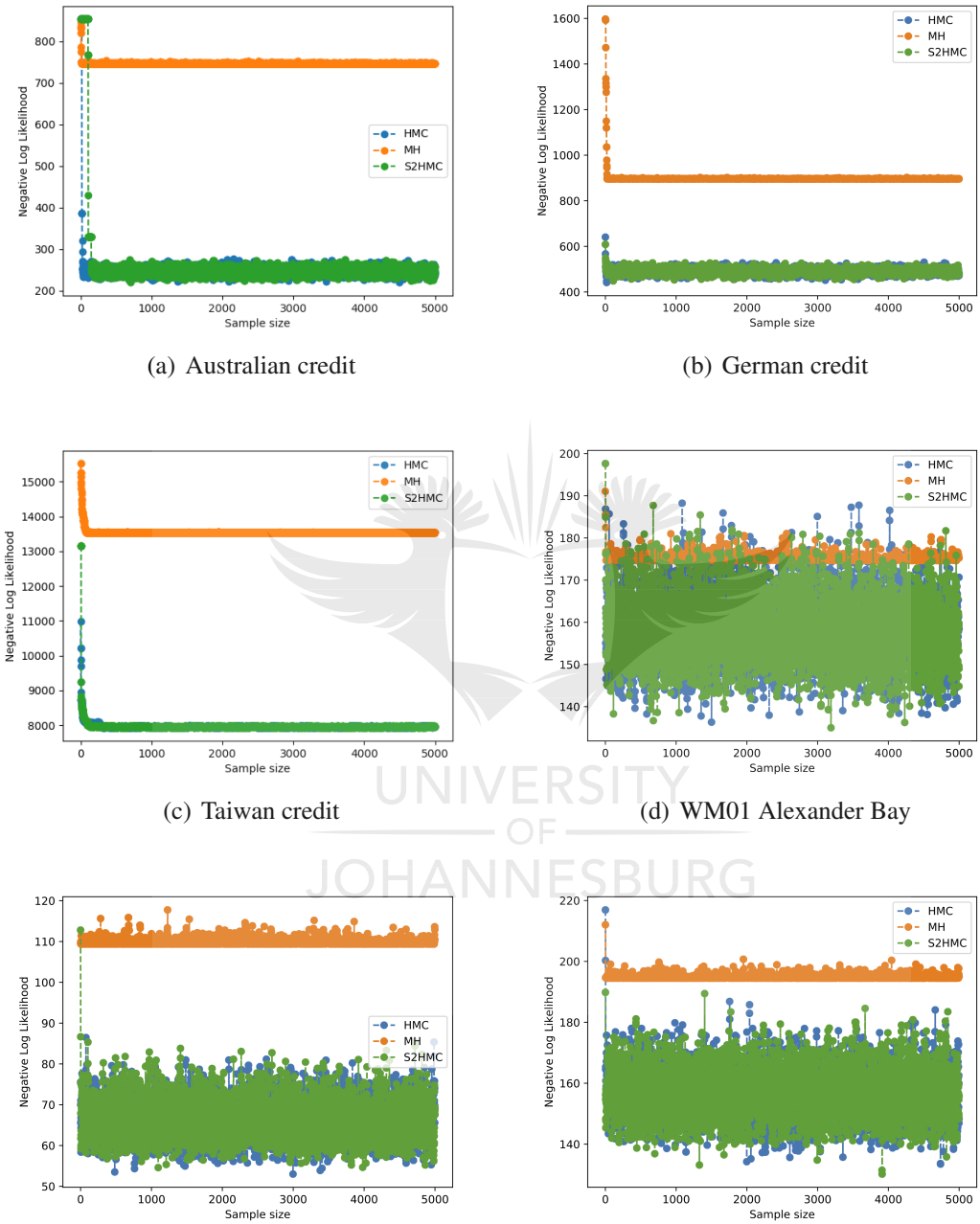
In summing up the experimental results - an important finding is that both S2HMC and HMC can sample sufficiently representative parameters of the BNNs across datasets. When looking at sampling performance, the results suggest that the shadow Hamiltonian in S2HMC marginally improves the acceptance rates and consequently the ESS.

3.4.4 Computation Time

Table 3.4 shows the mean computation time in minutes and the mean time normalised ESSs when generating 5000 samples across all the datasets. S2HMC generates 197.760 effective samples per minute relative to 459.809 effective samples per minute of HMC. This amounts to an effective speedup of 2.329. The increase in computational burden of S2HMC emanates from the additional steps that include pre-processing, calculation of the shadow Hamiltonian and post-processing.

Table 3.4 Mean computation time in minutes and mean time normalised ESSs for HMC and S2HMC when generating 5000 samples across all six datasets.

Dataset	Mean time (Minutes)	Mean ESS/t
HMC	7.884	459.809
S2HMC	16.413	197.760



(a) Australian credit

(b) German credit

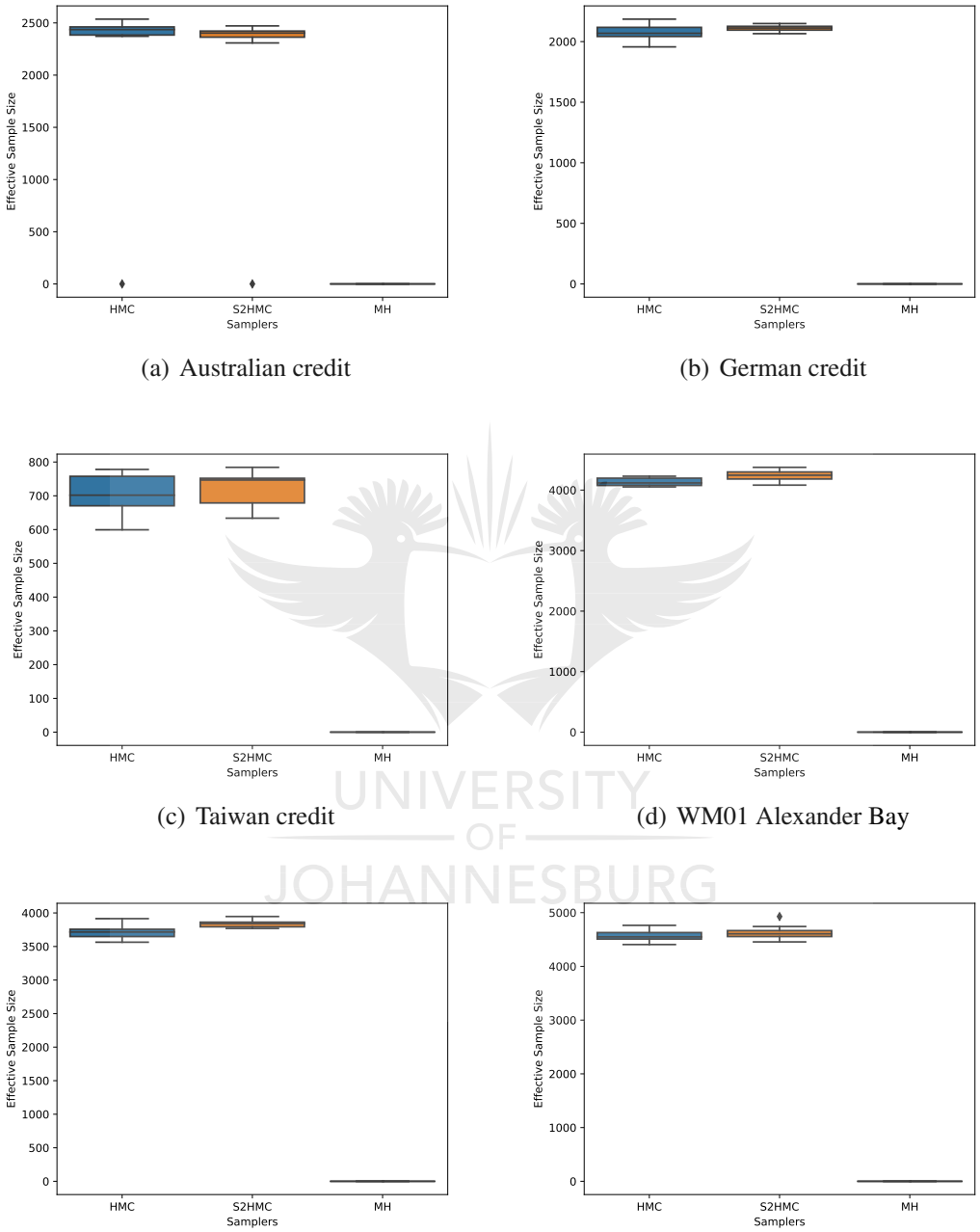
(c) Taiwan credit

(d) WM01 Alexander Bay

(e) WM05 Napier

(f) WM13 Jozini

Figure 3.3 Negative log-likelihood trace plots for MH (orange), HMC (blue), S2HMC (green) for all datasets.

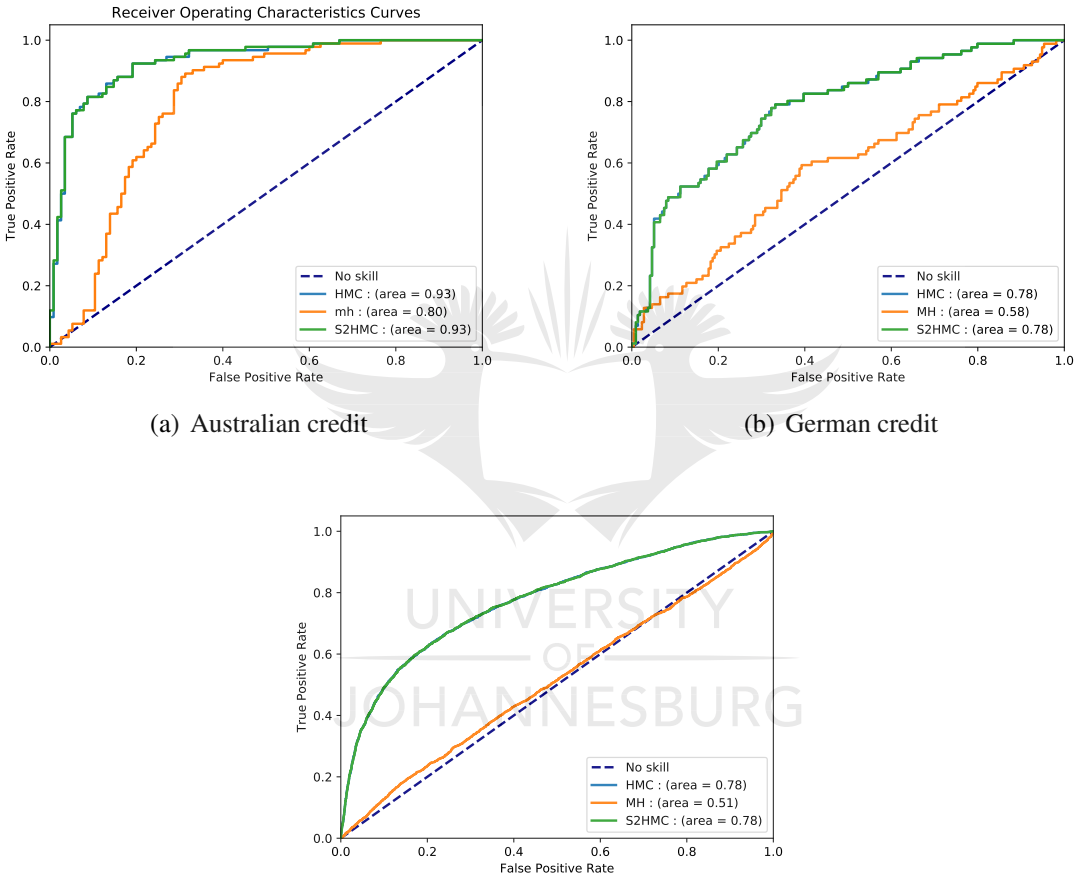


(a) Australian credit (b) German credit

(c) Taiwan credit (d) WM01 Alexander Bay

(e) WM05 Napier (f) WM13 Jozini

Figure 3.4 Boxplots showing the distribution of ESS over ten independent chains on each dataset.



(a) Australian credit

(b) German credit

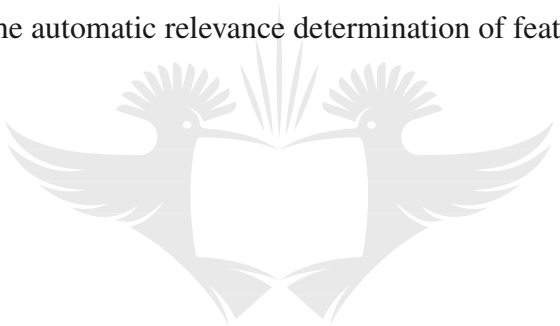
(c) Taiwan credit

Figure 3.5 ROC curves based on mean class probabilities for the credit default datasets.

3.5 Conclusion

This chapter presented the first implementation in literature of S2HMC in BNN inference. Initial experiments on simulated data show the robustness of S2HMC as both step sizes and problem dimensions increase. Experiments on real-world data for credit classification and wind speed forecasting show marginally higher ESSs and acceptance rate statistics for S2HMC relative to HMC. The predictive performance of S2HMC and HMC based BNNs is found to be similar, suggesting that both samplers are effective in posterior exploration. When considering these findings, it is important to take into account the additional computations required in S2HMC (relative to HMC) for variable pre and post processing as well as computation of the shadow Hamiltonians. S2HMC is found to increase the average computation time per effective sample by a factor of 2.329.

In the next chapter, we explore alternating S2HMC and HMC with Gibbs sampling of hyperparameters for the automatic relevance determination of features.



Chapter 4

Bayesian Variable Importance Using Automatic Relevance Determination Priors

The sampling methods discussed in Chapters 2 and 3 can be augmented with a hierarchical structure such that the relevance of each input can be inferred. In this chapter, we investigate automatic relevance determination priors in a hierarchical structure with MH, HMC and S2HMC samplers.

4.1 Introduction

Traditional criticisms of NN models is their "black-box" nature, hindering the ability to explain the influences of inputs on predictions. This becomes an impediment in applications of high societal importance such as the ones we consider in this work (credit and energy) [86, 84]. A natural question a banker will be asked when applying ML algorithms is "why customer X's credit was declined while customer Y's credit was granted". An electrical utility's management will typically ask themselves; "which data inputs are useful in predicting the amount of wind energy output in the next hour?"

Various attempts have been made to endow NNs with interpreterbity to provide insights on questions such as the above have primarily been based on perturbation or sensitivity methods [9, 105, 7].

The Bayesian framework presents a principled mechanism for inferring the relevance of various inputs using a stratified prior for each input called an automatic relevance determi-

nation (ARD) prior [75]. This approach is theoretically justified and intuitive in terms of probability theory.

4.2 Automatic Relevance Determination

The BNN formulation presented in Chapter 2 can be parameterised such that different groups of weights can come from unique prior distributions and thus have unique regularisation parameters α_c for each group. An ARD prior is one where weights associated with each network input have a distinct prior. Weights belonging to the same group (input) share the same regularisation parameter. The loss function in ARD is as follows [75]:

$$P(w|D,\alpha,\beta,H) = \frac{1}{Z(\alpha,\beta)} \exp\left(\beta E_D + \sum_c \alpha_c E_{W_c}\right) \tag{4.1}$$

The regularisation hyperparameters for each group of weights can be estimated online during the inference stage. The resulting regularisation parameter α_c for each input can be inferred as denoting the relevance of each input. Irrelevant inputs will have high values of the regularisation parameter, meaning that their weights will be forced to decay to values close to zero. Conversely on the basis of the posterior variances $\frac{1}{\alpha_c}$, important inputs will have weights with high variances while less important inputs will have low variances. Since the mean for the weight priors is fixed at zero, it therefore follows that weights with high variances are allowed to take values far from zero (thus higher influence), while those with small variances are forced to take values close to zero (thus lower influence). Figure 4.1 shows this rationale graphically where larger variances allow for more density far from zero, while low variances concentrate the density around zero.

4.2.1 Inference of ARD Hyperparameters

Inference of ARD hyperparameters is a fundamentally complex inverse problem due to the multimodal and non-convex nature of the ARD loss function [93, 75, 110]. MacKay [75] obtains estimates of the ARD hyperparameters by maximising the evidence. This procedure only becomes possible because the simplifying assumptions of the Laplace approximation result in a tractable closed form calculation of the evidence. This approach however, still suffers from the drawbacks of Laplace approximation discussed in Chapter 2. A detailed exposition of MacKay [75]'s approach is provided in Appendix B.

Neal [94] employs an alternating framework between Gibbs sampling for ARD hyperparameters from Cauchy, Gamma and Dirichlet posteriors while using HMC for network weights. This approach, as it is based on MCMC converges to an asymptotically exact

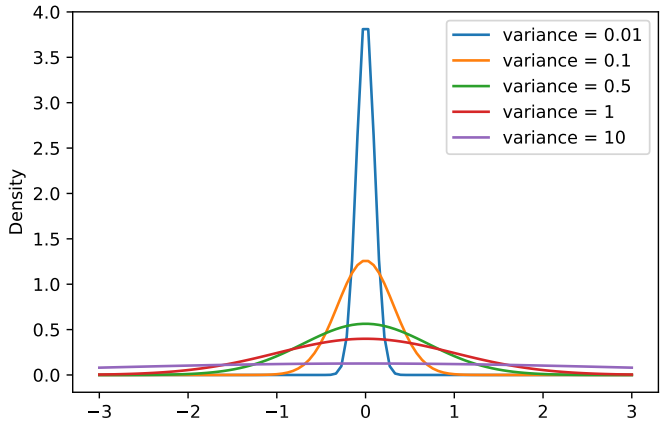


Figure 4.1 Prior distribution for weights under various variance settings.

posterior. However numerous design choices are still required such as the choice of prior distribution.

In this work we employ a similar approach to Neal [94] by allowing the α_c parameters to follow a Gamma prior with a fixed shape parameter τ and a fixed rate parameter θ such that:

$$P(\alpha_c) = \frac{\theta^{\tau}}{\Gamma(\tau)} \alpha_c^{\tau-1} \exp(-\theta\tau) \tag{4.2}$$

After taking into account the data likelihood on a given set of weights, the posterior for the parameter α_c corresponding to a particular group of parameters C becomes

$$P(\alpha_c | w_i \in C) \propto \alpha_c^{\tau+N_c-1} \exp\big(-\alpha_c(\theta + E_{W_c})\big) \tag{4.3}$$

Where N_c is the number of weights in group C. Equation 4.3 is still in the canonical form for a Gamma$(\tau + N_c, \theta + E_{W_c})$ distribution. This allows for Gibbs sampling of α_c parameters from the joint posterior given a fixed set of weights. Algorithm 4.1 shows a Gibbs step added to S2HMC at the beginning of each iteration, the approach of Neal [94] is shown in Appendix B Section B.3. In practice alternating between Gibbs sampling for hyper-parameters and sampling for weights allows for several uninterrupted weight sampling iterations before the update of hyper-parameters. This creates stability in the potential energy, which facilitates the convergence of both chains.

Algorithm 4.1 Separable Shadow Hamiltonian Hybrid Monte Carlo with ARD

Data: Dataset $\{\mathbf{X}, \mathbf{y}\}$
Result: N samples of model parameters $\mathbf{w}$ and $\frac{N}{n_{\text{Gibbs}}}$ samples of parameters α_c.
initialise the network weights $\mathbf{w}$ **for** $n \leftarrow 1$ **to** N **do**
  if $mod(n, n_{Gibbs}) = 0$ **then**
    Sample hyper-parameters α_c from Gamma$(\tau + N_c, \theta + E_{W_c})$
  end
  sample the auxiliary momentum variables $\mathbf{p}$
  p$\sim \mathcal{N}(0, \mathbf{M})$
  Initialise $\tilde{H}(\mathbf{w}, \mathbf{p})$
  Apply the pre-processing mapping $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$
  Use leapfrog steps on $(\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  for $t \leftarrow 1$ **to** L **do**
    $\hat{\mathbf{p}}(t + \varepsilon/2) \leftarrow \hat{\mathbf{p}}(t) + (\varepsilon/2)\frac{\partial \hat{H}}{\partial \hat{\mathbf{w}}}\left(\hat{\mathbf{w}}(t)\right)$
    $\hat{\mathbf{w}}(t + \varepsilon) \leftarrow \hat{\mathbf{w}}(t) + \varepsilon \frac{\hat{\mathbf{p}}(t+\varepsilon/2)}{M}$
    $\hat{\mathbf{p}}(t + \varepsilon) \leftarrow \hat{\mathbf{p}}(t + \varepsilon/2) + (\varepsilon/2)\frac{\partial \hat{H}}{\partial \hat{\mathbf{w}}}\left(\hat{\mathbf{w}}(t + \varepsilon)\right)$
  end
  Apply the post-processing mapping $(\mathbf{w}_*, \mathbf{p}_*) = \mathcal{X}^{-1}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  Calculate $\tilde{H}(\mathbf{w}_*, \mathbf{p}_*)$
  Metropolis update step:
  $(\mathbf{w}, \mathbf{p})_n \leftarrow (\mathbf{w}_*, \mathbf{p}_*)$ with probability:
  $min\left(1, \frac{\hat{H}(\mathbf{w}_*, \mathbf{p}_*)}{\hat{H}(\mathbf{w}, \mathbf{p})}\right)$
end

4.2.2 ARD Committees

In order to rationalise the feature relevance measures emanating from each method, we propose a simple majority vote committee approach for feature selection. This approach minimises reliance on one inference approach and thus adds some robustness to the feature selection task. Table 4.1 gives an illustration of this committee framework with p features and n samplers. In the ARD case each sampler gives a vote v_{ik} on the basis of some function of the posterior variance. In this work, a vote is attributed to a sampler if the posterior variance estimate $\frac{1}{\alpha_{ik}}$, corresponding to feature i is within the top five features on the basis of the ranked posterior variances from a particular sampler k. This can be defined as:

$$v_{ik} = f(\alpha_{ik}) = \begin{cases} 1, & \text{if } \frac{1}{\alpha_{ik}} \geq \frac{1}{\alpha_{\mathbb{I}_5}} \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

where $\frac{1}{\alpha_{\mathbb{I}_5}}$ is the fifth highest ranked posterior variance value amongst the p, $\frac{1}{\alpha_{ik}}$ values for sampler k. Thus this voting mechanism looks for concordance between samplers with some allowance for differences in feature rankings. In general $f(\alpha_{ik})$ can be any well defined function of a feature importance statistic of choice not limited to posterior variances (e.g. coefficients , gini impurity). The framework also allows for $f(\alpha_{ik})$ to include weighting of votes using metrics such as the predictive accuracy of the method that gives a specific feature importance statistic. This framework is consistent with Garcia-Chimeno et al. [47] who use of RF, Boosted Trees, LASSO and uni-variate variable selection to arrive at committee outputs. Pehlivanlı et al. [97] also employs a similar approach using t-statistics, Fisher scores, the ReliefF algorithm and effective range based gene selection.

Table 4.1 An illustration of the ARD committee framework with p features and n samplers.

Feature	Sampler 1	Sampler 2	...	Sampler k	...	Sampler n	Total Votes
Feature 1	υ_{11}	υ_{11}	...	υ_{1k}	...	υ_{1n}	$\sum_{k=1}^{n} \upsilon_{1k}$
Feature 2	υ_{21}	υ_{21}	...	υ_{2k}	...	υ_{2n}	$\sum_{k=1}^{n} \upsilon_{2k}$
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
Feature i	υ_{i1}	υ_{i2}	...	υ_{ik}	...	υ_{in}	$\sum_{k=1}^{n} \upsilon_{ik}$
...	...	...	...	...	...	...	...
...	...	...	...	...	...		...
Feature p	υ_{p1}	υ_{p2}	...	υ_{kp}	...	υ_{1n}	$\sum_{k=1}^{n} \upsilon_{pk}$

4.3 Experiment Setup

We demonstrate ARD using the samplers and datasets already discussed in this work. Significant attention will be given to the Taiwan credit and the WASA datasets due to their publicly available feature labels. The experimental settings are described in Table 4.2. After the committee selections, the BNNs are then retrained on only the selected features to investigate the selection's efficacy based on the retrained models' predictive performance.

4.4 Results and Discussions

Figure 4.2 shows ROC curves generated from mean predictions of NNs with ARD trained using HMC, MH and S2HMC, respectively. It can be seen that models trained with Hamiltonian dynamics based samplers significantly outperform MH. In the ARD case with additional hyperparameters to sample, the performance of MH deteriorates to below random guessing.

Table 4.2 Experimental Settings for ARD Sampling Runs.

Setting	Value
BNN Number of Hidden Layers	1
BNN Number of Hidden Neurons	5
BNN Activation Function	ReLU
Sampling Trajectory Length (HMC/S2HMC)	100
Number of samples between hyper-parameter Samples (n_{Gibbs})	50

The reasoning for such outperformance is similar to the discussions in Chapters 2 and 3. In this case, a slight decline in performance overall can be explained by the effect of additional regularisation in the input space. This effect typically occurs when the number of noisy features is low or non-existent [94].

4.4.1 Predictive Performance

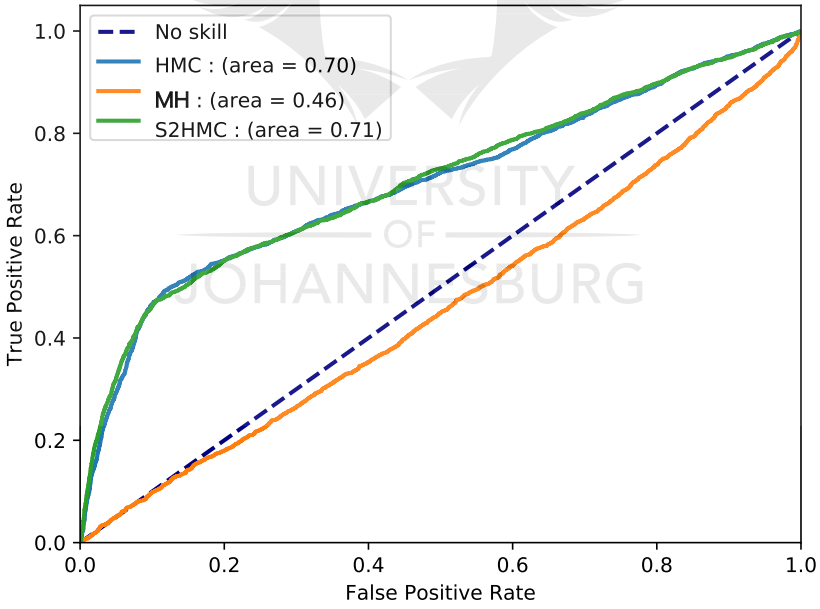


Figure 4.2 ROC curves for MH-ARD, HMC-ARD and S2HMC-ARD on the Taiwan credit dataset.

Table 4.3 shows the mean testing RMSE of the ARD BNNs on the windspeed datasets. As it follows from previous findings, MH-ARD is outperformed by HMC-ARD and S2HMC-ARD, respectively. It is important to note that with the introduction of the ARD prior on the

wind speed datasets there was an improvement in RMSE compared to the results without ARD reported in Section 3.4. Thus ARD in the context of these regression datasets has the desired effect of introducing sparseness in their input space. This sparseness reduces the influence of noisy inputs. This finding is also similar to those of MacKay [75], Neal [94] and Mbuvha et al. [86] using both Laplace approximation and MCMC inference.

Table 4.3 Mean testing RMSE resulting from BNNs trained using ten independent chains of MH, HMC and S2HMC at each of the weather stations.

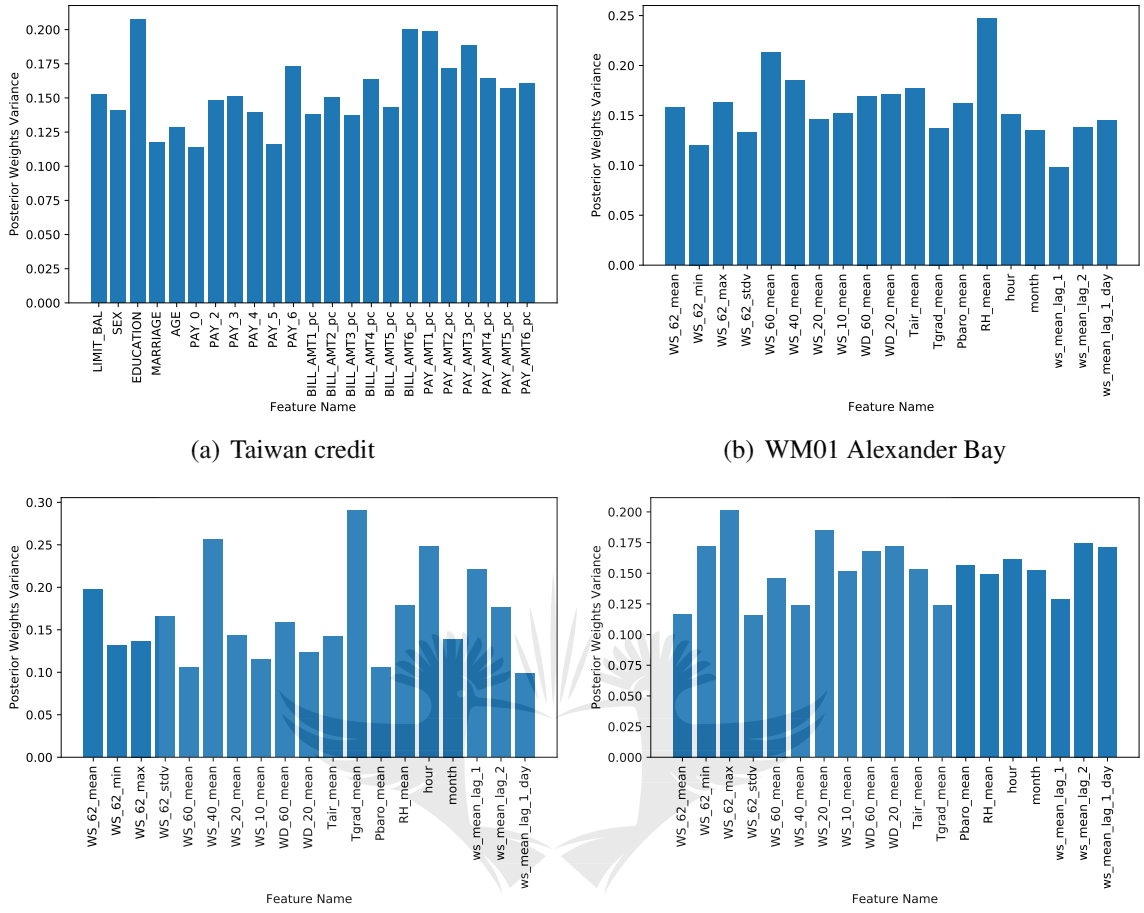
Dataset	MH	HMC	S2HMC
WM01 Alexander Bay	8.939	1.642	1.651
WM05 Napier	8.466	1.496	1.527
WM13 Jozini	7.423	1.425	1.420

4.4.2 ARD Committees and Feature Importance

The plots in Figures 4.3 - 4.5 and the ARD committees in Tables 4.4 - 4.7 are utilised to decode the ARD relevance statistics. The ARD committee considers "votes" from each method based on the top 5 highest posterior variances. Features with more than 2 votes are then considered highly relevant. On an overall level, it can be seen from the posterior variance plots that HMC-ARD and S2HMC-ARD result in clearer sparsity when compared to regularisation by MH-ARD. Confidence in the relative feature importance can implicitly be assigned based on the predictive performance of each inference method.

In the Taiwan credit dataset, education is found to be highly important across all inference methods. This finding aligns well with known findings on the relationships between income, education, and creditworthiness [91, 11]. Bill amounts six months ago, payments in the first and third month are also found to be important with concurrence between inference methods. There is significant congruence between the feature relevance found in this dataset and similar studies by Mbuvha et al. [83] and Sariannidis et al. [109] using Gaussian approximation and ensemble tree-based methods, respectively

In the WM01 Alexander Bay dataset, calendar month emerges as highly important, signalling a strong seasonal effect on wind speed patterns. Atmospheric pressure and relative humidity also have high posterior variance. This relationship between atmospheric pressure, relative humidity and wind speed is well documented in atmospheric physics literature [77]. One day lagged wind speeds for the same hour also are highly important, suggesting within-day cycles. Strong lagged dependencies were similarly found in a study of the Norwegian



(a) Taiwan credit

(b) WM01 Alexander Bay

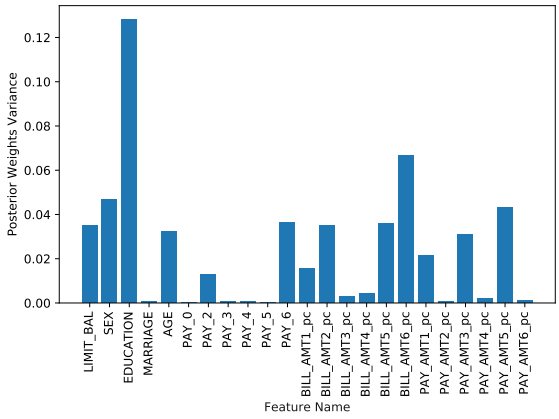
(c) WM05 Napier

(d) WM13 Jozini

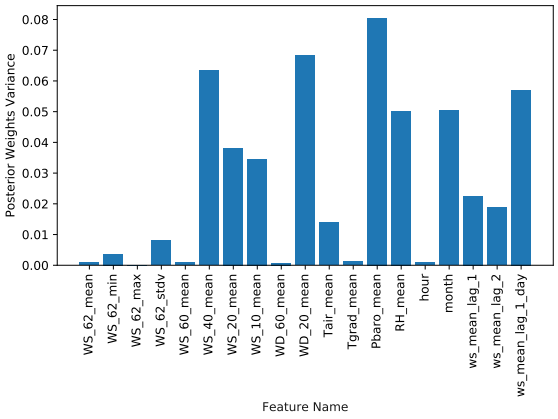
Figure 4.3 Mean posterior variances from the MH-ARD model which indicate the relevance of each attribute.

wind farm dataset by Mbuvha et al. [86]. Wind speeds at lower altitudes, i.e. 40 meters and 20 meters, are also found to be significant.

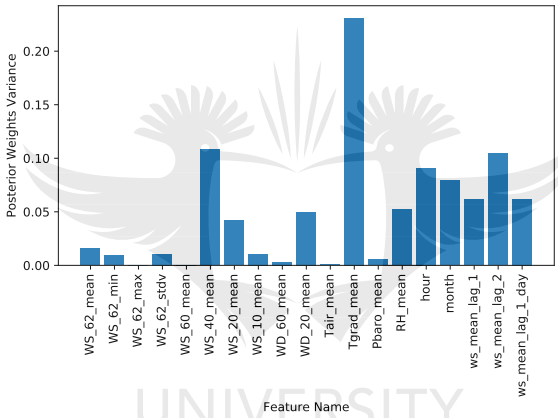
Posterior variances at WM05 Napier and WM13 Jozini sites show some overlap with the first site with a few site-specific variations. Air temperature and temperature gradients start to emerge as relevant features. This finding again reconciles well with atmospheric physics theory concerning causal links between atmospheric temperature, barometric pressure and wind speed [2]. Further evidence of intra-day and monthly cycles in wind speed patterns is also observed.



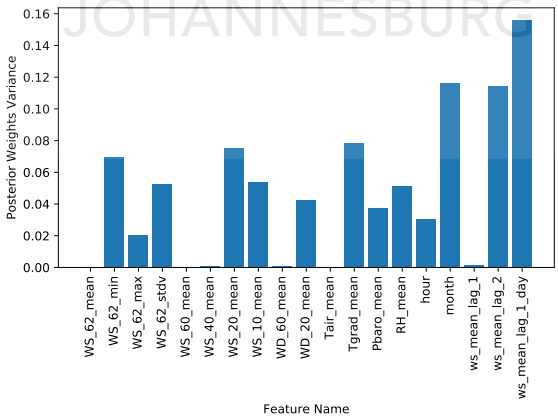
(a) Taiwan credit



(b) WM01 Alexander Bay

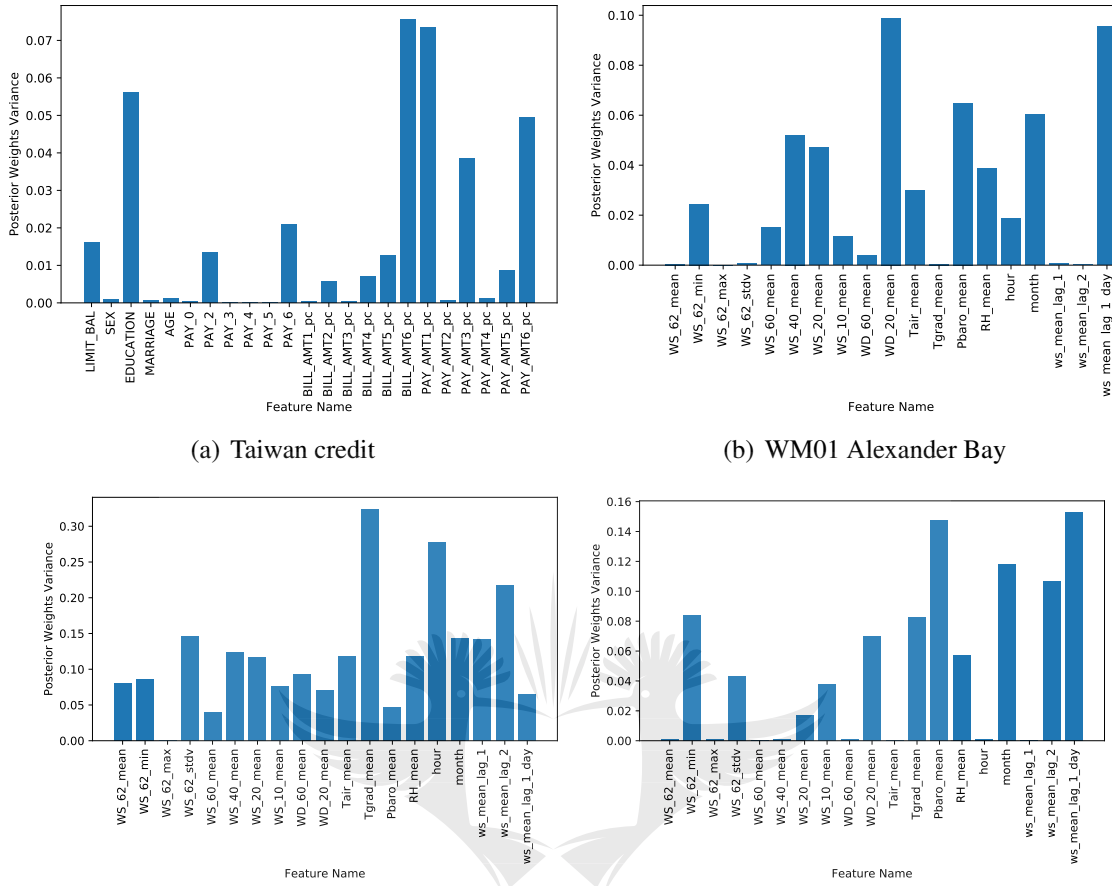


(c) WM05 Napier



(d) WM13 Jozini

Figure 4.4 Mean posterior variances from the HMC-ARD model which indicate the relevance of each attribute.



(a) Taiwan credit

(b) WM01 Alexander Bay

(c) WM05 Napier

(d) WM13 Jozini

Figure 4.5 Mean posterior variances from the S2HMC-ARD model which indicate the relevance of each attribute.

4.4.3 Re-training BNNs on Relevant Features

The robustness of the feature selections in the previous sections is investigated by retraining the BNNs on a reduced set of features corresponding to the ARD committee results.

Figure 4.6 depicts the ROC curves from the mean predictions of the retrained BNNs. A reduction in AUC from 0.78 to 0.70 for HHC/S2HMC is observed. It is important to note that the retrained models use only 4 of the 23 features (17%). This suggests that the 4 identified features, which is only (17%) of predictors, are responsible for about 90% of the predictive performance. Thus on this evidence, the ARD committees were able to identify highly relevant predictors. Notably, the predictive performance MH based BNNs improves from 0.56 to 0.58. Thus, in addition to inferences on the quality of the selected features, it is also implied that the MH sampler benefits from the significant reduction in the dimensionality of the problem.

Table 4.4 Committee table of ARD feature selections based on the top 5 features from each inference method on the Taiwan credit dataset.

Feature	MH	HMC	S2HMC	Total Votes
LIMIT_BAL		1		1
SEX		1		1
EDUCATION	1	1	1	3
MARRIAGE				0
AGE				0
PAY_0				0
PAY_2				0
PAY_3				0
PAY_4				0
PAY_5				0
PAY_6	1			1
BILL_AMT1_pc				0
BILL_AMT2_pc				0
BILL_AMT3_pc				0
BILL_AMT4_pc				0
BILL_AMT5_pc				0
BILL_AMT6_pc	1	1	1	3
PAY_AMT1_pc	1		1	2
PAY_AMT2_pc				0
PAY_AMT3_pc	1		1	2
PAY_AMT4_pc				0
PAY_AMT5_pc		1		1
PAY_AMT6_pc			1	1

Retraining results on the WASA datasets are similar to those on the classification dataset. While there is some deterioration in the HMC and the S2HMC based BNNs, MH based BNNs improve on all sites. A point worth noting is that the deterioration in the predictive performance of HMC/S2HMC based BNNs is not commensurate with the significant 59% reduction in the number of features. Again, the implication thereof is that the MH sampler gains additional efficiencies from the selection of high information quality features and the reduction in the dimensionality of the problem. Predictive performance robustness after dimensionality reduction through ARD is similarly found in the work of Mbuvha et al. [86] using the Gaussian Approximation approach.

Table 4.5 Committee table of ARD feature selections based on the top 5 features from each inference method on the WM01 Alexander Bay dataset.

Feature	MH	HMC	S2HMC	Total Votes
WS_62_mean	1			1
WS_62_min				
WS_62_max				
WS_62_stdv				
WS_60_mean	1			1
WS_40_mean	1		1	2
WS_20_mean		1	1	2
WS_10_mean				
WD_60_mean				
WD_20_mean				
Tair_mean				
Tgrad_mean				
Pbaro_mean		1	1	2
RH_mean	1	1		2
hour				
month		1	1	2
ws_mean_lag_1				
ws_mean_lag_2				
ws_mean_lag_1_day		1	1	2

Table 4.6 Committee table of ARD feature selections based on the top 5 features from each inference method on the WM05 Napier dataset.

Feature	MH	HMC	S2HMC	Total Votes
WS_62_mean	1			1
WS_62_min	1			1
WS_62_max				
WS_62_stdv				
WS_60_mean				
WS_40_mean				
WS_20_mean	1	1	1	3
WS_10_mean				
WD_60_mean				
WD_20_mean				
Tair_mean	1	1	1	3
Tgrad_mean			1	1
Pbaro_mean				
RH_mean	1	1	1	3
hour				
month	1	1	1	3
ws_mean_lag_1	1	1		2
ws_mean_lag_2				
ws_mean_lag_1_day				1

Table 4.7 Committee table of ARD feature selections based on the top 5 features from each inference method on the WM13 Jozini dataset.

Feature	MH	HMC	S2HMC	Total Votes
WS_62_mean				
WS_62_min	1	1	1	3
WS_62_max	1			
WS_62_stdv				
WS_60_mean				
WS_40_mean				
WS_20_mean	1	1	1	3
WS_10_mean				
WD_60_mean	1			1
WD_20_mean	1			1
Tair_mean				
Tgrad_mean		1	1	2
Pbaro_mean			1	1
RH_mean				
hour				
month		1	1	2
ws_mean_lag_1				
ws_mean_lag_2	1	1	1	3
ws_mean_lag_1_day	1	1	1	3

Table 4.8 Mean Testing RMSE resulting from BNNs re-trained using the relevant features identified in tables 4.5 to 4.7 for each weather station.

Dataset	MH	HMC	S2HMC
WM01 Alexander Bay	4.245	2.515	2.517
WM05 Napier	3.85	2.750	1.527
WM13 Jozini	2.842	2.164	1.420

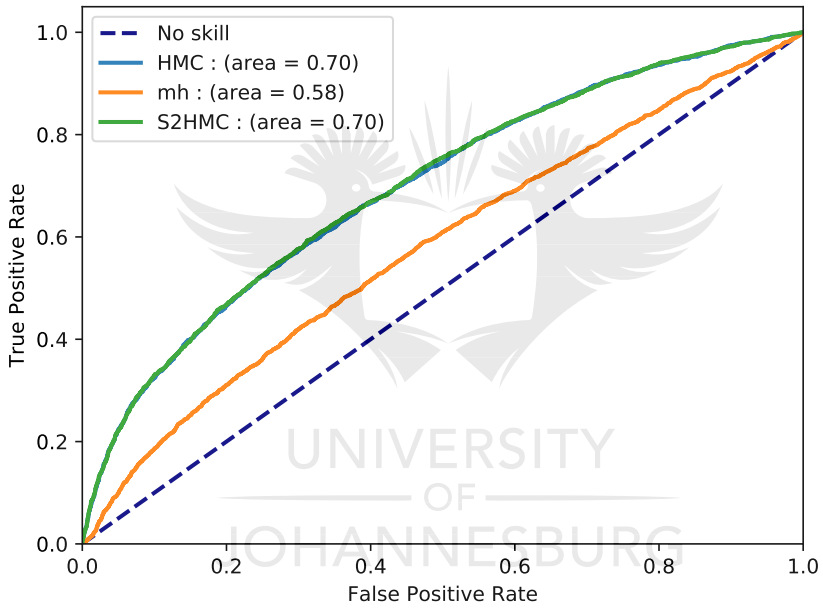


Figure 4.6 ROC curve for MH-ARD, HMC-ARD and S2HMC-ARD on the Taiwan credit dataset after fitting on relevant features identified by sampler committee in Table 4.4.

4.5 Conclusion

In this chapter, ARD through a hybrid of Gibbs sampling with MH, HMC and S2HMC is presented. This setup, to the knowledge of the author - is the first implementation of S2HMC-ARD in literature. On the regression wind speed datasets, it is observed that ARD improves predictive performance. On the classification dataset, predictive performance sightly declines signalling that the original feature set contains little noise.

An ARD committee framework is presented with the aim of adding robustness to the feature selection procedure. This committee framework can easily be generalised to any odd number of feature importance statistics from any learning machine (e.g. Random Forests, Gradient Boosting, Lasso).

The features identified from the posterior variance estimates indicate that education, credit limits and payments early and later in the term are of strong influence in predicting credit outcomes. On the WASA datasets, calendar month, day lagged wind spends, and relative humidity emerged as influential features. These findings suggest highly cyclical patterns both within the day and the year.

An inefficient sampler such as MH was shown to benefit from the ARD committees' dimensionality reduction. ARD provided a probabilistically principled structure for inference on the relative influences of various features on BNN predictions. Coupled with predictive uncertainty from BNNs, ARD facilitates troubleshooting and root-cause analysis of NN predictions. The relative computational costs of the methods remain similar to those discussed in Chapter 3.

Chapter 5

Healing Products of Gaussian Process Experts

Gaussian processes (GPs) are nonparametric Bayesian models that have been applied to regression and classification problems. One of the approaches to alleviate their cubic training cost is the use of local GP experts trained on subsets of the data. In particular, product-of-expert models combine the predictive distributions of local experts through a tractable product operation. While these expert models allow for massively distributed computation, their predictions typically suffer from erratic behaviour of the mean or uncalibrated uncertainty quantification. In this chapter, new schemes for calibrating predictions via a tempered softmax weighting and a Wasserstein barycenter are presented to provide a solution to these problems for multiple product-of-expert models, including the generalised product of experts and the robust Bayesian committee machine. The work presented in this chapter also appears in Cohen et al. [30] (where I was a joint first author).

5.1 Introduction

Gaussian processes (GPs) [104] are nonparametric stochastic processes that have been applied extensively to regression and classification problems. However, their cubic training and quadratic prediction cost hinders their application in large-scale problems. Different approaches alleviate this issue, including sparse approximations [114, 31, 101, 120], the exploitation of structural assumptions [133] and local-expert models [123, 103, 26, 35, 107, 121].

Sparse approximations effectively reduce the rank of the covariance matrix through inducing inputs, reducing the training cost from $O(n^3)$ to $O(nm^2)$, where m is the number

of inducing points and n is the size of the training dataset. Optimisation consists of jointly learning kernel hyperparameters and inducing locations. In particular, Titsias [120] treats inducing locations as variational parameters and optimises them and the kernel hyperparameters by maximising a lower bound on the marginal likelihood. Hensman et al. [59] scale this approach by introducing mini-batching, reducing the complexity to $O(m^3)$, while Gal et al. [46] reparametrise the problem to allow for distributed inference.

An alternative to sparse GP approximations is to use local experts. Here, the training dataset is partitioned into J subsets of size m where $m \ll n$. Then, J local GP experts are trained on each of these subsets, thereby reducing the training complexity to $O(Jm^3)$. Importantly, this approach scales to large datasets because training and prediction with each expert can be distributed across computing units [35]. For instance, Rasmussen and Ghahramani [103], Tresp [123], Trapp et al. [121] consider mixture-of-expert models (MoEs). In particular, Trapp et al. [121] propose a sum-product network with local-expert GP leaves allowing for tractable and exact posterior inference. Other approaches leverage product-of-experts models (PoEs) [122, 26], whereby a global prediction can be obtained by means of averaging the predictions of local experts. Generalisations of these models can control the relevance of different experts when making predictions [27, 35, 71].

In this work, we focus on PoEs because closed-form inference and training are tractable, which is not the case with typical MoEs. However, previous PoE approaches to combining predictions at test time suffer from unrealistic over- or under-estimation of the variance and erratic mean behaviours. This holds especially when the number of points m assigned to each expert is low, in which case a significant number of experts are weak [35]. These approaches are thus not overly robust to variations in m, which is a significant shortcoming. Unfortunately, scalability requires the number of points per expert to be reasonably small due to the $O(m^3)$ scaling of individual experts. We propose a solution to these problems by controlling the sparsity of expert weights through a tempered softmax at test time, leveraging tools from the the extensive uncertainty calibration literature [99, 17, 51]. We also propose a novel principled PoE approach arising from the optimal transport literature, which we name the barycenter of GPs, and demonstrate that its performance is competitive to the best PoE models on small and large-scale datasets. We demonstrate empirically that calibrating expert weights lead to substantial performance gains in both mean prediction and uncertainty quantification. We also discuss common failures of PoE models extensively and propose guidelines to remediating these.

The contributions of this chapter include introducing a new method for averaging GP experts based on optimal transport theory that performs competitively with the best-performing

PoE models and propose a solution to the shortcomings of previously proposed PoEs, based on controlling the weight sparsity.

5.2 Gaussian Processes

Gaussian processes are powerful nonparametric Bayesian models, often used for regression. A GP is defined as a collection of random variables, every finite subset of which is jointly Gaussian distributed [104]. GPs are fully defined by a mean $m(\cdot)$ and a kernel $k(\cdot,\cdot)$.

Consider a regression problem with a training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$ of n noisy observations $y_i = f(\mathbf{x}_i) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_y^2)$. With a GP prior on f, it follows that $f(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_x, \mathbf{K}_x + \sigma_y^2\mathbf{I})$ where $(\mathbf{m}_x)_i = m(\mathbf{x}_i)$ and $(K_x)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The mean and variance of the Gaussian posterior predictive distribution of the function value $f(\mathbf{x}_*)$ at a test point $\mathbf{x}_*$, are given by [104]:

$$\mathbb{E}[f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mathbf{m}_{\mathbf{x}_*} + \mathbf{K}_*(\mathbf{K}_x + \sigma_y^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}_\mathbf{x}), \tag{5.1}$$

$$\text{var}[f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mathbf{K}_{**} - \mathbf{k}_*^T(\mathbf{K}_x + \sigma_y^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{5.2}$$

respectively, where $\mathbf{K}_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$ and $\mathbf{K}_* = k(\mathbf{X}, \mathbf{x}_*)$. Here $\mathbf{X}, \mathbf{y}$ contain the training inputs and targets, respectively. Kernel hyperparameters and the noise parameter σ_y are learned by maximising the log-marginal likelihood [104]:

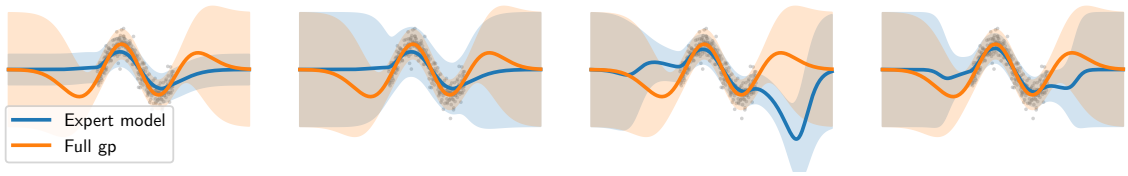
$$\log p(\mathbf{y}|\mathbf{X}, \theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{m}_x, \mathbf{K}_{xx} + \sigma_y^2\mathbf{I}). \tag{5.3}$$

Computing equation 5.3 requires the inversion of the matrix $\mathbf{K}_{xx} + \sigma_y^2\mathbf{I} \in \mathbb{R}^{n\times n}$, so that GP training scales in $O(n^3)$, where n is the size of the training dataset. Optimizing the log-marginal likelihood in equation 5.3 and the computation of the posterior predictive distribution at a test input $\mathbf{x}_*$ become computationally intractable for large training sets.

Several approaches have been explored to avoid the cubic training cost of GPs. These are mostly based on either sparse approximations and structure-exploiting assumptions to the covariance matrix [101, 120, 59, 133] or training distributed (weak) experts on subsets of the full dataset [122, 26, 35, 121, 71]. An alternative is to use large-scale computing infrastructure and incomplete Cholesky decompositions [130].

5.2.1 Sparse Gaussian Processes

Sparse GPs [101, 114] leverage inducing inputs to reduce the rank of the matrix to be inverted. Sparse variational GPs extend this by introducing a variational approximation to the



(a) PoE (b) gPoE (c) BCM (d) rBCM

Figure 5.1 Different expert models trained on synthetic data with three points per GP expert on a dataset of 300 observations. (a) PoE; (b) gPoE; (c) BCM; (d) rBCM. All models display some shortcomings in their vanilla forms. For instance (a): over-confidence, (b) under-confidence within data region, and (c)-(d) erratic mean in the transitioning region [30]

posterior [120], treating inducing inputs as variational parameters, and mini-batching [59] to scale. Wilson and Nickisch [133] exploit structural assumptions and combine inducing-point approaches with Kronecker and Toeplitz methods to perform kernel approximations leading to increased scalability. The approximation quality of sparse GPs relies on the number of inducing points, and a large number of these can be required to represent the local structures of fast varying functions.

5.2.2 Gaussian Process Experts

Another approach to scaling GPs to large datasets is to use expert models. Here, multiple GPs are trained on subsets of the data, and predictions are recombined using either a product-of-expert (log-opinion pool) approach [60, 122, 26, 35, 107, 14], or a mixture-of-expert (linear-opinion pool) approach [123, 103, 121]. MoEs are useful in heteroskedastic and nonstationary settings, but do not typically allow tractable posterior inference, by contrast with PoEs.

In this work, we thus focus on product-of-expert models with M experts, which all share hyperparameters. We first describe the training of such models. Assuming a full GP is the model we seek to approximate, sharing kernel hyperparameters automatically regularises the population of experts: individual experts can not overfit to the local subset of the data they are fed with due to this shared set of hyperparameters. Assuming independence across experts (given the training data), the log-marginal likelihood is [35]:

$$\log p(\mathbf{y}|\mathbf{X},\theta) = \sum_{j=1}^{J} \log p_j(\mathbf{y}^{(j)}|\mathbf{x}^{(j)},\theta), \tag{5.4}$$

where $\{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ is the data assigned to the j^{th} expert. To train the model, we maximise the log-marginal likelihood in equation 5.4 with respect to the (shared) kernel hyperparameters [35]. Training can be distributed across diverse compute clusters, enabling scaling with total time complexity $O(Jm^3)$ where J is the number of experts, and $m \ll n$ is the size of the training set of each expert. With J compute nodes, the complexity per node reduces to $O(m^3)$. This is in stark contrast to the $O(n^3)$ scaling of full GPs.

In the following, we describe the process of predicting with product-of-GP-experts models. In particular, we introduce several approaches to recombining predictions from trained experts. We note that an important particularity of these models is that all predictive distributions $p(f_*|\mathbf{x}_*)$ of function values are Gaussians, which is not the case with MoEs. Also, throughout the chapter, aggregation is performed in function space, and the likelihood is subsequently applied.

Product of experts (PoE) The product of GP experts aggregates predictions of M experts at test point $\mathbf{x}_*$ via [60]:

$$p(f_*|\mathbf{x}_*) \propto \prod_{j=1}^{M} p_j(f_*|\mathbf{x}_*, D^{(j)}), \tag{5.5}$$

where the predictive mean and precision of the Gaussian predictive distribution are given by

$$m_{poe}(\mathbf{x}_*) = \sigma^2_{poe}(\mathbf{x}_*) \sum_{j=1}^{M} \sigma_k^{-2}(\mathbf{x}_*) m_j(\mathbf{x}_*) \tag{5.6}$$

$$\sigma^{-2}_{poe}(\mathbf{x}_*) = \sum_{j=1}^{M} \sigma_j^{-2}(\mathbf{x}_*), \tag{5.7}$$

respectively. Here, $D^{(j)}$ is the dataset associated with the jth expert. The PoE has the advantage that predictions are easy to compute. However, as the number M of experts increases, the resulting aggregated variances vanish, which leads to overconfident predictions [35, 71], such that one of the main purposes of using a Gaussian process (reasonable uncertainty quantification) is defeated. We indeed observe such over-confident behaviour in Figure 5.1(a).

(Generalised) product of experts – (g)PoE The (g)PoE aggregates predictions of M experts at test point $\mathbf{x}_*$ via [26]:

$$p(f_*|\mathbf{x}_*) \propto \prod_{j=1}^{J} p_j^{\beta_j(\mathbf{x}_*)}(f_*|\mathbf{x}_*, \mathcal{D}^{(j)}), \tag{5.8}$$

where the predictive mean and precision are

$$m_{(g)poe}(\mathbf{x}_*) = \sigma^2_{(g)poe}(\mathbf{x}_*) \sum_{j=1}^{J} \beta_j(\mathbf{x}_*)\sigma_j^{-2}(\mathbf{x}_*)m_j(\mathbf{x}_*),$$
$$\sigma^{-2}_{(g)poe}(\mathbf{x}_*) = \sum_{j=1}^{J} \beta_j(\mathbf{x}_*)\sigma_j^{-2}(\mathbf{x}_*),$$

respectively. Here, $\mathcal{D}^{(j)} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ is the data assigned to expert j, $\beta_j(\mathbf{x}_*)$ controls the contribution of expert j at $\mathbf{x}_*$ (typically a measure of its confidence at $\mathbf{x}_*$), and the PoE model is recovered when setting $\beta_j(\mathbf{x}_*) = 1$ for all j. As the number of experts J increases, the PoE's aggregated variance vanishes, which leads to overconfident predictions [35, 71]. An illustration of such behaviour is shown in Figure 5.1(a).

The gPoE with uniform weights $\sum_j \beta_j(\mathbf{x}_*) = 1$ falls back to the prior far from training points, which is a desirable property. However, a drawback is that it over-estimates the variance close to training points [35] when setting the weights uniformly ($\beta_j(\mathbf{x}_*) = \frac{1}{J}$). We also observe such behaviour in Figure 5.1(b).

Bayesian committee machine (BCM) The Bayesian committee machine (BCM) [122] assumes conditional independence $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f_*$, and by repeated application of Bayes' theorem, we obtain the predictive distribution at test point $\mathbf{x}_*$ as [122]:

$$p(f_*|\vec{x}_*) = \frac{\prod_{j=1}^{M} p_j(f_*|\mathbf{x}_*, D^{(j)})}{p^{M-1}(f_*|\mathbf{x}_*)}, \tag{5.9}$$

such that the predictive mean and precision are:

$$m_{bcm}(\mathbf{x}_*) = \sigma^2_{bcm}(\mathbf{x}_*) \sum_{j=1}^{M} \sigma_j^{-2}(\mathbf{x}_*)m_j(\mathbf{x}_*) \tag{5.10}$$

$$\sigma^{-2}_{bcm}(\mathbf{x}_*) = \sum_{j=1}^{M} \sigma_j^{-2}(\mathbf{x}_*) + (1-M)\sigma_*^{-2}, \tag{5.11}$$

where the normalisation $p^{M-1}(f_*|\vec{x}_*)$ is the prior prediction at $\vec{x}_*$ taken to the $(M-1)^{th}$ power and σ_*^{-2} is the prior precision. This predictive distribution guarantees that the model falls back to the prior far from training data. However, it is not effective when very few data points are assigned to each expert. The BCM does exhibit uncharacteristic behaviour in regions of the state space, where we transition from high-density data to low-density data [35]. We observe such behaviour in Figure 5.1(c)

(Robust) Bayesian committee machine – (r)BCM The (robust) Bayesian committee machine (r)BCM [122, 35] assumes conditional independence $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f_*$. By repeated

application of Bayes' theorem, we obtain the predictive distribution [35]:

$$p(f_*|\mathbf{x}_*) = \frac{\prod_{j=1}^{J} p_j^{\beta_j(\mathbf{x}_*)}(f_*|\mathbf{x}_*, \mathcal{D}^{(j)})}{p^{-1+\sum_j \beta_j(\mathbf{x}_*)}(f_*|\mathbf{x}_*)} \tag{5.12}$$

at test point $\mathbf{x}_*$. Then the predictive mean and precision are

$$m_{(r)bcm}(\mathbf{x}_*) = \sigma^2_{(r)bcm}(\mathbf{x}_*) \sum_{j=1}^{J} \beta_j(\mathbf{x}_*)\sigma_j^{-2}(\mathbf{x}_*)m_j(\mathbf{x}_*),$$

$$\sigma^{-2}_{(r)bcm}(\mathbf{x}_*) = \sum_{j=1}^{J} \beta_j(\mathbf{x}_*)(\sigma_j^{-2}(\mathbf{x}_*) - \sigma_*^{-2}) + \sigma_*^{-2},$$

respectively. BCM is recovered when $\beta_j(\mathbf{x}_*) = 1$ for all j. This predictive distribution guarantees that the model falls back to the prior far from training data. However, the BCM exhibits uncharacteristic behaviour in regions transitioning from high to low-density data [35]; see Figure 5.1(c). The rBCM mitigates some of the issues of the BCM and allows for flexible weighting of GP experts, via $\beta_j(\mathbf{x}_*)$, but it still exhibits problematic behaviour in regions with density transitioning; see Figure 5.1(d).

Likelihoods As expert averaging is performed in function space throughout the chapter, we will need to map the aggregated predictive GP distribution $p(f_*)$ through a likelihood function to predict labels y_*. In the conjugate regression case with a Gaussian likelihood, this can be done in closed form [104]. For classification, we consider non-conjugate likelihoods, such as the Bernoulli or Poisson likelihoods. Since the aggregated predictive distribution $p(f_*)$ in PoEs is Gaussian, we obtain the expected predicted label by averaging under the posterior predictive latent distribution

$$\mathbb{E}[y_*|\mathbf{x}_*] = \int \phi(f(\mathbf{x}_*))\mathcal{N}(f_*|m(\mathbf{x}_*), \sigma^2(\mathbf{x}_*))df_*, \tag{5.13}$$

where ϕ is a classification likelihood (e.g., Bernoulli, Probit). The integral in equation 5.13 is intractable, but we can resort to standard approximate inference techniques for GP classification, such as MAP estimation, Laplace approximation, expectation propagation, variational inference, or numerical integration [104, 58]. Similarly, the marginal likelihood, which we use for training the experts, becomes intractable. Therefore, we use stochastic variational inference to train models in this setting [58], and apply the same strategies for training and prediction with other GP expert models.

5.3 Barycenters of Predictive Distributions

Now, we propose a new way of combining experts' predictions leveraging optimal transport theory. We begin by introducing two important tools, namely the Wasserstein distance and barycenter between 1D Gaussians, noting that both can be computed using simple closed-form formulas.

Given two Gaussians $\mu = \mathcal{N}(\mathbf{m}_1, \mathbf{K}_1)$ and $\nu = \mathcal{N}(\mathbf{m}_2, \mathbf{K}_2)$, we define the 2-Wasserstein distance between them as [128]

$$\mathcal{W}_2^2(\mu, \nu) = \|\mathbf{m}_1 - \mathbf{m}_2\|_2^2 + Tr\Big(\mathbf{K}_1 + \mathbf{K}_2 - 2(\mathbf{K}_1^{\frac{1}{2}}\mathbf{K}_2\mathbf{K}_1^{\frac{1}{2}})^{\frac{1}{2}}\Big). \tag{5.14}$$

Equation 5.14 can be interpreted as the minimal expected cost of transporting mass from the Gaussian μ to the Gaussian ν.

Given that distance, the barycenter between Gaussian-distributed $\mu_1, ..., \mu_J$ with weights β is:

$$\bar{\mu} = \arg\min_{\mu} \sum_{j=1}^{J} \beta_j \mathcal{W}_2^2(\mu_j, \mu), \tag{5.15}$$

where $\sum_j \beta_j = 1, 0 \leq \beta_j \leq 1$. Álvarez Esteban et al. [141] show that if $\mu_j = \mathcal{N}(\mathbf{m}_j, \mathbf{K}_j)$ for all j, the Wasserstein barycenter with weights β is itself a Gaussian measure $\bar{\mu} = \mathcal{N}(\bar{\mathbf{m}}, \bar{\mathbf{K}})$, where:

$$\bar{\mathbf{m}} = \sum_{j=1}^{J} \beta_j \mathbf{m}_j, \quad \bar{\mathbf{K}} = \sum_{j=1}^{J} \beta_j (\bar{\mathbf{K}}^{\frac{1}{2}}\mathbf{K}_j\bar{\mathbf{K}}^{\frac{1}{2}})^{\frac{1}{2}}. \tag{5.16}$$

We also propose a fixed-point iteration algorithm to efficiently compute $\bar{\mathbf{K}}$ in equation 5.16.

In the following, we discuss our approach to aggregating GP experts' predictions for regression and classification. In all product-of-experts models we discussed, each expert computes a predictive distribution of the form $p_j(f(\mathbf{x}_*)|\mathcal{D}^{(j)}) = \mathcal{N}(m_j(\mathbf{x}_*), \sigma_j^2(\mathbf{x}_*))$, where m_j and σ_j^2 are the posterior predictive mean and variance of the jth GP expert at test point $\mathbf{x}_*$. Since these distributions (in latent space of f) are all Gaussian (by definition of the GP), we propose combining these into their weighted 2-Wasserstein barycenter using equation 5.16, which can be computed in closed form in the one-dimensional case [20]. We obtain the

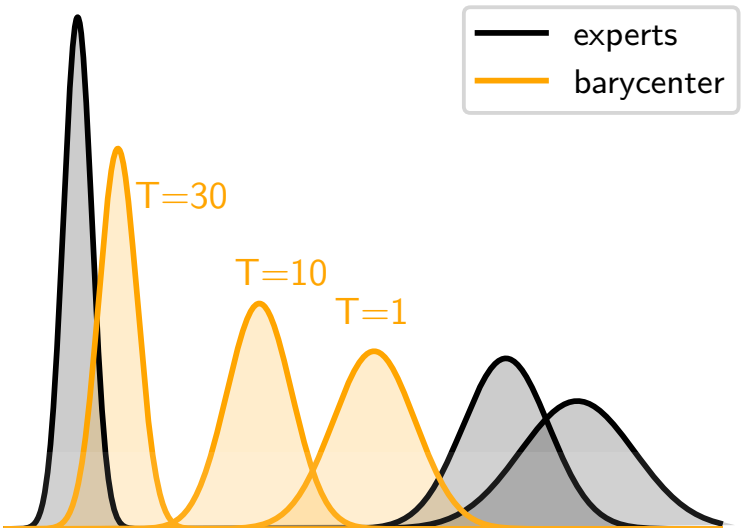


Figure 5.2 Illustration of the barycenter of GPs with tempered softmax weighting. At x_*, one expert (left) is highly confident about its prediction, and two are highly unconfident (right). As temperature increases, only confident experts get weight (sparsity increases), thus the barycenter is pulled towards the confident expert [30].

closed-form Gaussian predictive distribution

$$p(f_*|\mathbf{x}_*) = \mathcal{N}(m_{bar}(\mathbf{x}_*), \sigma^2_{bar}(\mathbf{x}_*)) \tag{5.17}$$

$$m_{bar}(\mathbf{x}_*) = \sum_{j=1}^{J} \beta_j(\mathbf{x}_*) m_j(\mathbf{x}_*), \tag{5.18}$$

$$\sigma^2_{bar}(\mathbf{x}_*) = \sum_{j=1}^{J} \beta_j(\mathbf{x}_*) \sigma^2_j(\mathbf{x}_*). \tag{5.19}$$

The barycenter of GPs is a product-of-experts variant, and the mean and variance of the predictive distribution consist of the weighted average of predictive means and variances of the experts. Importantly, such weights can be a function of test points, analogously to the gPoE and the rBCM.

We train the barycenter of GPs following the training procedure of other PoEs discussed in Section 5.2.2, namely by optimising the marginal likelihood in equation 5.4, and we share expert hyperparameters for regularising the expert pool.

The barycenter of GP's predictive distribution is deeply connected to that of previously proposed PoEs. In particular, the aggregated mean is a weighted mean of the experts' predictive means, which is also the case for other expert models. The aggregated variance is a weighted mean of experts' variances, which has a similar interpretation to the predictive

precision of other PoEs, itself a weighted mean of the experts' precisions. The barycenter of GPs falls back to the prior outside the data regime which is a highly desirable property, and is also true for gPoE with uniform weights, and rBCM. Further connections are discussed in Section 5.4.

WASP [117] leverages a related idea, which consists in averaging subset posteriors using Wasserstein barycenters. However, they average discrete measures consisting of samples from the different posteriors at a discretizsed set of points, and then have to solve a large linear problem to compute the barycenter. By contrast, we average marginal posterior predictive distributions, which is done in closed-form leveraging the known closed-form of barycenters of Gaussians in 1D.

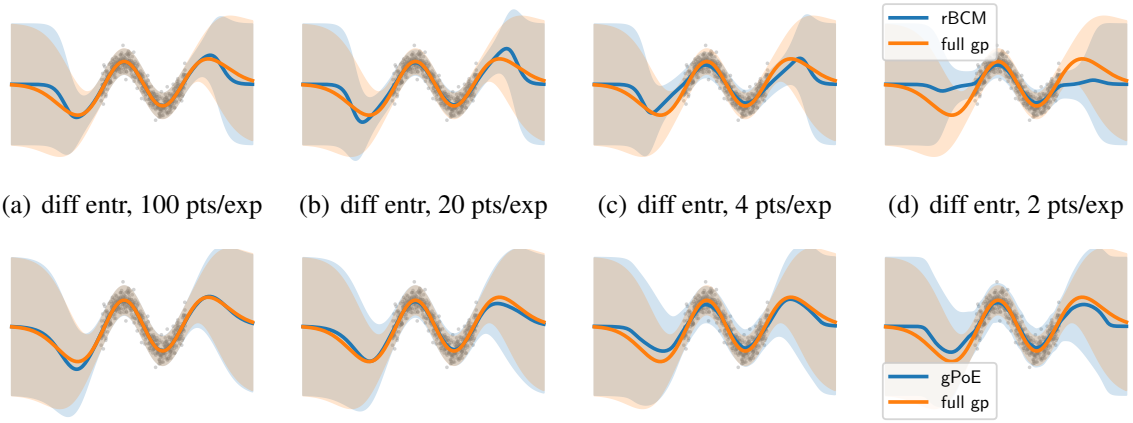
5.4 Calibrating Product-of-Experts

In the previous sections, we introduced several approaches to combining predictions of local GP experts, including our proposal, the barycenter of GPs. We also discussed shortcomings of previous PoE approaches in low-data regimes, including under- (Figure 5.1(a)) and over-estimation of the variance (Figure 5.1(b)), but also erratic and uncharacteristic behaviours of the mean and variance predictions (Figures 5.1(c)–5.1(d)). These behaviours are exacerbated when the number of points assigned per expert is low, which leads to a significant number of weak experts[1].

Whilst exact Gaussian processes are well-known for well-calibrated uncertainty estimates, approximate Bayesian methods fall prey to inferior calibration. These issues in the context of sparse GP approximations are discussed in depth by Bauer et al. [12]. Our aim in this section is to remediate such calibration issues for PoE models. There has been a significant recent emphasis on uncertainty calibration in the deep learning community [51], and we will extend tools from this literature to the problem of training product-of-experts-based GP approximations.

The prevalence of weak experts is significantly affected by the data assignment strategy. For example, when using stationary kernels, clustering-based partition approaches tend to create localised experts which leads to greater weak expert prevalence. The latter approach is intuitively sensible if we choose stationary kernels, as expert approaches can be interpreted as divide-and-conquer strategies. However, this strategy can have disastrous consequences if expert weights are not properly regulated. Indeed, the lower the number of points per expert, the weaker the experts are overall if the training data associated with these experts

[1] We refer to weak experts as experts that provide calibrated predictions only on local subsets of the data manifold.



(a) diff entr, 100 pts/exp (b) diff entr, 20 pts/exp (c) diff entr, 4 pts/exp (d) diff entr, 2 pts/exp

(e) soft-var, 100pts/exp (f) soft-var, 20 pts/exp (g) soft-var, 4 pts/exp (h) soft-var, 2 pts/exp

Figure 5.3 Full GP baseline (orange) and expert models (blue) trained on synthetic data with a decreasing number of points per experts (Left to Right), and for different weighting methods: rBCM with differential entropy in Figures (a)–(d) and the gPoE with proposed softmax-variance in Figures (e)–(h). Our method is significantly more robust to variations in the number of points per experts [30].

is not dense in the vicinity of test inputs. This can be observed in Figure 5.3 (Top), where pathologies arise as the number of points per expert decreases significantly. This is mainly caused by the poorly regulated expert weighting.

In this scenario, weight sparsity has to increase to alleviate the weakness of most experts by relying only on locally-calibrated predictions. In the following section, we propose a solution to such shortcomings that can be applied to gPoE, rBCM and the barycenter of GPs.

The softmax function provides a natural mechanism for controlling the sparsity of experts' importance weights. In particular, an (inverse) temperature parameters T can directly control the degree of smoothness and sparsity in the resulting weights. Using a temperature-endowed softmax to combat miscalibrated predictions has seen widespread use, ranging from hierarchical mixtures of experts [17] to support vector machines [99] and deep learning [51].

We adapt these ideas to weighted ensembles of GP experts, such as the gPoE, the rBCM and the barycenter. We therefore propose a general expression for expert weights as

$$\beta_j(\mathbf{x}_*) \propto \exp(-T\,\psi_j(\mathbf{x}_*)), \quad \sum_{j=1}^{M} \beta_j(\mathbf{x}_*) = 1, \tag{5.20}$$

where T is an (inverse) temperature parameter that controls the sparsity between experts by multiplicatively compounding the weights of stronger experts. The functional $\psi_j(\mathbf{x}_*)$

describes the level of confidence of the jth expert at test point $\mathbf{x}_*$. We provide an illustration of such framework in Figure 5.2. In particular, we plot the barycenter of GP experts' predictive distribution at $\mathbf{x}_*$ under several temperature values, highlighting that as temperature increases, the barycenter gets pulled towards the most confident expert, i.e., uncertain experts are not given weight in the prediction.

We now discuss the choice of confidence functional ψ. We set $\psi_j(\mathbf{x}_*)$ to the posterior predictive variance at $\mathbf{x}_*$, i.e.,

$$\psi_j(\mathbf{x}_*) = \sigma_j^2(\mathbf{x}_*). \tag{5.21}$$

Intuitively, this will give high weight to experts with low posterior predictive variance (high confidence) in their prediction. Such experts have training data close to test points (all experts share the same hyperparameters), and should thus have a high contribution in the final prediction. Our proposal can also be combined with the previously proposed differential entropy weighting [26]

$$\psi_j(\mathbf{x}_*) = \tfrac{1}{2}(\log \sigma_*^2 - \log \sigma_j^2(\mathbf{x}_*)) \tag{5.22}$$

or with the Wasserstein distance in equation 5.14, leveraging its closed-form computation in the 1D case [141], which has the same complexity as differential entropy. In the infinite temperature limit, weight sparsity is maximised.

Proposition 1. *In the infinite-temperature limit $T \to \infty$, and if $\psi_j = \sigma_j^2(\mathbf{x}_*)$, the gPoE, the rBCM and the barycenter of GPs have equivalent predictive distributions.*

Intuitively, in such a regime, only the most confident experts have (equal) weight, and as a result the inverse of the weighted sum of precisions of the two former equals the weighted sum of the variances, and thus predictive distributions are equal. Under weaker assumptions, the rBCM and the gPoE are equivalent:

Proposition 2. *If $\sum_j \beta_j(\mathbf{x}_*) = 1$ for all $\mathbf{x}_*$, then $m_{rbcm}(\mathbf{x}_*) = m_{gpoe}(\mathbf{x}_*)$ and $\sigma^2_{rbcm}(\mathbf{x}_*) = \sigma^2_{gpoe}(\mathbf{x}_*)$.*

Proposition 2 highlights that under normalised weights, gPoE and rBCM are equivalent. Therefore, under our weighting proposal, which consists of using normalised tempered softmax functionals, gPoE and rBCM's predictive distributions are equal. Proofs of propositions 1 and 2 can be found in Cohen et al. [30].

5.5 Experiments

Throughout this section, we evaluate the performance of our approaches to calibrating GP experts when applied to regression and classification, while comparing with sparse variational methods and previous approaches to local-expert weighting and averaging. We consider performance metrics including the negative log-predictive density (NLPD), and the root mean squared error (RMSE). [2]

Baselines: We consider the PoE, BCM, gPoE, rBCM and barGP with random and K-means partitioning to assess the effect of the data assignment strategy. For the rBCM, gPoE and barGP, we evaluate the proposed softmax weighting strategy (BAR_var, rBCM_var, gPoE_var) with different temperature choices as proposed in Section 5.4. We also evaluate differential entropy (_entr) weighting [26] and uniform weighting (_unif).

5.5.1 Regression

We evaluate the performance of our approach to setting local experts' weights and compare it to previous weighting methods. In particular, we evaluate the robustness of the rBCM using differential entropic weighting as motivated by Deisenroth and Ng [35], and the gPoE and barycenter with softmax-variance weighting (proposed in this work), when reducing the number of points per experts. As motivated in Section 5.4, the softmax weighting should encourage expert sparsity, and as such be effective when the number of points per experts decreases (causing the number of strong experts to decrease). In this case, we set the temperature T to 15 (for $T \geq 15$, sparsity is well-controlled; see Figure 5.4).

Figure 5.3 shows that the gPoE with softmax-variance weighting provides sensible and calibrated predictions even with only two points per experts, while the rBCM with differential entropic weights leads to erratic mean and variance behaviours in the transitioning region even with 20 points per experts. Thus, encouraging sparsity in the expert weights through the variance-softmax weighting enables expert models to be robust to the reduction in the number of points per experts, thereby addressing a shortcoming of local-expert models. Also, the erratic behaviour in the transitioning region appears remediated. With very weak experts, it is unrealistic to expect uncertainties that are identical to the full GP's uncertainty. Importantly, the predictions are (moderately) on the conservative side for the softmax-variance weighting, which is preferable to overconfidence. We report similar behaviours for the barycenter combination (Section 5.3) in Appendix C (figure C.1).

We now perform an evaluation of the different expert models with different choices of weighting, including our approach (softmax-variance) and previous approaches (uniform for

[2] Code available at https://github.com/samcohen16/Healing-POEs-ICML

Dataset	BAR_var	gPoE_var	rBCM_diff_entr	rBCM_var	BCM	gPoE_unif	PoE
WM01 Alexander Bay	-0.053 (1.294)	-0.0514(1.296)	0.009 (1.357)	0.718 (2.066)	1.336 (2.684)	1.331 (2.679)	19.497 (20.845)
WM05 Napier	-0.324 (1.084)	-0.323 (1.083)	-0.298 (1.109)	0.103 (1.510)	0.246 (1.653)	1.278 (2.685)	24.940 (26.347)
WM13 Jozini	0.321 (1.336)	0.322 (1.337)	0.366 (1.381)	1.161 (2.176)	1.696 (2.711)	1.335 (2.350)	9.864 (10.879)

Table 5.1 Mean NLPD (RMSE) on the three weather stations for the regression datasets using clustering partitioning.

gPoE and differential entropy for rBCM) on the 3 WASA datasets already considered in this work. For softmax weightings, we use a temperature of 100, which performs well across all the datasets (i.e., it induces enough weight sparsity). Clustering partitioning is used for the regression dataset, additional results with random partitioning are provided in Appendix C Table C.1.

Table 5.1 shows that the gPoE and the barGP with softmax-variance weighting outperform all other models on all datasets. They significantly outperform the rBCM with differential entropy weighting across datasets. Moreover, the gPoE with softmax-variance weighting outperforms the gPoE with uniform weighting by a large margin. It is also notable, that in terms of RMSE that across the regression datasets product of experts models outperforms the BNNs reported in Chapters 2, 3 and 4.

These demonstrate that controlling the sparsity of expert weights heals issues of the product-of-expert models and leads to more calibrated uncertainty quantification and mean estimation, while having the same running cost. Finally, Liu et al. [71] and Zhang and Williamson [139] found that the rBCM and the gPoE under-perform when averaging in y-space, which is the reason we average in f-space.

5.5.2 Sensitivity and Robustness Analysis

We now consider the sensitivity of the gPoE, rBCM and barGP with softmax-variance weighting to the temperature hyperparameter T. For the gPoE and barGP, we use the normalised version of the softmax (in which case the gPoE is equivalent to the rBCM with such weights). We also evaluate the rBCM's robustness, when using unnormalised softmax-variance weights. To that end, we consider the Kin40k benchmark dataset and plot the NLPD as a function of the temperature (Figure 5.4). We observe that the NLPD decreases monotonically until stabilising for both the gPoE and the barGP, demonstrating the robustness of these models with respect to the choice of the temperature parameter. Hence, the NLPD is stable across temperatures (for $T > 15$) when using normalised weights. We also produce such an analysis for unnormalised softmax-variance weights (in which case the rBCM is not equivalent to the gPoE). In this case, the model is more sensitive to the change in temperature, and it is difficult to find a single softmax scaling that performs well across

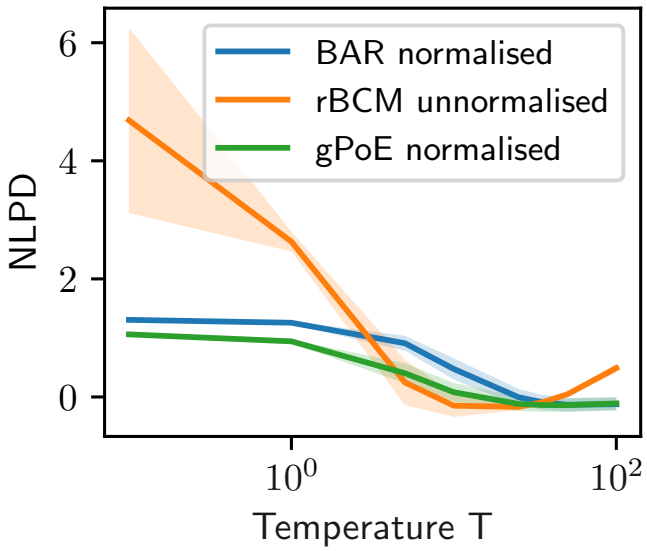


Figure 5.4 NLPD against temperature for different expert models with softmax-variance weighting on a benchmark dataset (Kin40K) [30].

	BAR_var	BCM	gPoE_unif	gPoE_var	PoE	rBCM_diff_entr	rBCM_var	$SVGP_{500}$
Top-1-accur.	0.911	0.895	0.894	0.910	0.895	0.895	0.395	0.862
Top-2-accur.	0.964	0.955	0.954	0.964	0.955	0.956	0.411	0.939
Top-3-accur.	0.981	0.976	0.975	0.982	0.976	0.976	0.418	0.967
NLPD	0.311	0.852	0.383	0.312	0.850	0.878	2.475	0.497

Table 5.2 Top-*n* accuracy and NLPDs on the MNIST dataset (PCA features).

small- and large-scale benchmarks. Hence, using normalised softmax weights is important to obtaining models that are robust to the choice of temperature.

5.5.3 Classification Benchmarks

We now assess the classification performance of expert models in a non-conjugate classification setting on the MNIST dataset and the Taiwan credit dataset. We opt for the much larger MNIST in lieu of the smaller Australian and German credit datasets to demonstrate the capabilities of PoE on large scale datasets. The MNIST dataset comprises of 10 classes with a training/test split of $60,000/10,000$ images. We reduce the dimensionality of images with

	BAR_var	BCM	gPoE_unif	gPoE_var	PoE	rBCM_diff_entr	rBCM_var	$SVGP_{500}$
Top-1-accur.	0.820	0.822	0.822	0.820	0.822	0.822	0.820	0.818
NLPD	0.803	1.182	0.841	0.767	1.177	1.173	0.952	0.709

Table 5.3 Top-1 accuracy and NLPDs on the Taiwan credit dataset.

PCA (20 principal components). Note that the overall accuracy resulting from PCA features will not outperform the state of the art. However, PCA features provide a deterministically reproducible basis for relative comparison of various methods. We assign 500 training points to each SVGP expert, and provide them with 100 trainable inducing inputs each. We use a multiclass likelihood with a robust-max link function. Note that in the setting of Liu et al. [71], classification is not directly applicable because averaging is happening in y-space, which is more challenging in non-conjugate settings.

Table 5.2 shows the classification results for MNIST. We report top-n accuracy and NLPD. Consistently, we observe that the BAR_var and gPoE_var outperform all products of experts and SVGP baseline models. The difference in performance between the rBCM_entr and gPoE_var shows that introducing weight sparsity via a tempered softmax improves the performance as it only allows confident experts to contribute to the aggregated predictions. We observe similar performance gaps between gPoE_unif and our proposals which suggests that using tempered softmax-variance weighting results in more informed posterior predictive means and variances.

The improvement of the SVGP_{100} expert models over a single (full) SVGP_{500} is not surprising since every single SVGP expert has the modelling capacity of the global SVGP, so that the distributed models effectively work with M times as many inducing inputs as the SVGP. This suggests that the combination of sparse GPs and expert models can be useful in settings, where a large number of inducing inputs for a full SVGP is required for good modelling.

Table 5.3 shows the results on the Taiwan credit dataset. It can also be seen that the BAR_var and gPoE_var also emerge as strong performers in terms of NLPD and accuracy. This performance can be similarly attributed to increase sparsity amongst predictors as well as normalisation of the softmax weights (when comparing to the rBCM). The competitive performance of the SVGP_{500} on this dataset can suggest that 500 inducing points can possibly be sufficiently representative of this smaller dataset when compared to the larger MNIST.

5.6 Conclusion

We identified significant shortcomings of previous approaches, notably the PoE, BCM, gPoE and rBCM, to scaling GP regression and classification via local-expert averaging. These models struggle in settings, where the number of strong experts is small, but the experts' weights are not sparse enough. Weight sparsity should thus be set to account for the overall strength of experts. To address these shortcomings, we control weight sparsity via the use of (normalised) softmax weights, along with a temperature to enforce this trade-off. Note

that our approach can be combined with SVGPs [58] (as was done in Section 5.5.3) but also with other methods, such as KISS-GP [133]. We provide strong empirical evidence that shortcomings of previous expert models can be addressed through this approach, which leads to substantial performance gains across datasets. We further propose a novel scalable and distributable approach to averaging GP experts' predictions by means of Wasserstein barycenters, which can be used for regression and classification problems. When combined with our weighting proposal, it obtains state-of-the art performance across most datasets. On the WASA regression datasets the new proposals outperform the MCMC based BNNs discussed in previous chapters. The computational complexity of our method is $O(Jm^3)$ compared to $O(n^3)$ and $O(nm^2)$ for full GP and sparse variational GPs, respectively, where the number of data points per expert m is less than the total number of data points n.

Chapter 6

Bayesian Parameter Inference in Infectious Disease Modelling

The Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) pandemic has highlighted the need for performing accurate inference with limited data. Fundamental to the design of rapid state responses is the ability to perform epidemiological model parameter inference for localised trajectory predictions. In this work, we perform Bayesian parameter inference using Markov Chain Monte Carlo (MCMC) methods on the Susceptible-Infected-Recovered (SIR) and Susceptible-Exposed-Infected-Recovered (SEIR) epidemiological models with time-varying spreading rates for South Africa. Some of the work presented in this chapter also appears in Mbuvha and Marwala [87].

6.1 Introduction

The first reported case of the novel coronavirus (SARS-CoV-2) in South Africa was announced on 5 March 2020, following the initial manifestation of the virus in Wuhan China in December 2019 [88, 132, 36]. Due to its further spread and the severity of its associated clinical outcomes, the disease was subsequently declared a pandemic by the World Health Organisation (WHO) on 11 March 2020 [132, 88]. In South Africa, by 26 April 2020, 4546 people had been confirmed to have been infected by the coronavirus with 87 fatalities [78].

Numerous states have attempted to minimise the growth in number of COVID-19 infections [43, 34, 88]. These attempts are largely based on non-pharmaceutical interventions (NPIs) aimed at separating "the infectious population from the susceptible population" [88].

These initiatives aim to strategically reduce the increase in infections to a level where their healthcare systems stand a chance of minimising the number of fatalities [88]. Some

of the critical indicators for policymaker response planning include projections of the infected population, estimates of health care service demand and whether current containment measures are effective [88].

As the pandemic develops in a rapid and varied manner in most countries, calibration of epidemiological models based on available data can prove to be difficult [116]. This difficulty is further escalated by the high number of asymptomatic cases and the limited testing capacity [132, 88].

A fundamental issue when calibrating localised models is inferring parameters of compartmental models such as susceptible-infectious-recovered (SIR) and the susceptible-exposed-infectious-recovered (SEIR) that are widely used in infectious disease projections. In the view of public health policymakers, a critical aspect of projecting infections is the inference of parameters that align with the underlying trajectories in their jurisdictions. The spreading rate is a parameter of particular interest which is subject to changes due to voluntary social distancing measures and government-imposed contact bans.

The uncertainty in utilising these models is compounded by the limited data in the initial phases and the rapidly changing dynamics due to rapid public policy changes.

To address these complexities, we utilise the Bayesian Framework for the inference of epidemiological model parameters in South Africa. The Bayesian framework allows for both incorporation of prior knowledge and principled embedding of uncertainty in parameter estimation.

This chapter combines Bayesian inference with the compartmental SEIR and SIR models to infer time-varying spreading rates that allow for quantification of the impact of government SARS-CoV-2 related interventions in South Africa.

6.2 Methods

6.2.1 Epidemiological Modelling

Compartmental models are a class of models that is widely used in epidemiology to model transitions between various stages of disease [22, 18, 88]. We now introduce the Susceptible-Exposed-Infectious-Recovered (SEIR) and the related Susceptible-Infectious-Recovered (SIR) compartmental models that have been dominant in COVID-19 modelling literature [43, 34, 72, 88].

The Susceptible-Exposed-Infectious-Recovered Model

The SEIR is an established epidemiological model for the projection of infectious diseases. The SEIR models the transition of individuals between four stages of a condition, namely:

- being susceptible to the condition,
- being infected and in incubation
- having the condition and being infectious to others and
- having recovered and built immunity for the disease.

The SEIR can be interpreted as a four-state Markov chain which is illustrated diagrammatically in Figure 6.1. The SEIR relies on solving the system of ordinary differential equations below representing the analytic trajectory of the infectious disease [88].

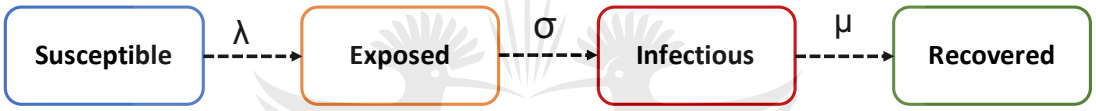


Figure 6.1 An Illustration of the underlying states of the Susceptible-Exposed-Infectious-Recovered Model(SEIR)

$$\frac{dS}{dt} = -\frac{\lambda SI}{N} \tag{6.1}$$

$$\frac{dE}{dt} = \frac{\lambda SI}{N} - \sigma E \tag{6.2}$$

$$\frac{dI}{dt} = \sigma E - \mu I \tag{6.3}$$

$$\frac{dR}{dt} = \mu I \tag{6.4}$$

Where S is the susceptible population, I is the infected population, R is the recovered population and N is the total population where $N = S + E + I + R$. λ is the transmission rate, σ is the rate at which individuals in incubation become infectious, and μ is the recovery rate. $1/\sigma$ and $1/\mu$ therefore, become the incubation period and contagious period respectively.

We also consider the Susceptible-Infectious-Recovered (SIR) model which is a subclass of the SEIR model that assumes direct transition from the susceptible compartment to the infected (and infectious) compartment. The SIR is represented by three coupled ordinary differential equations rather than the four in the SEIR. Figure 6.2 depicts the three states of the SIR model.

Figure 6.2 An Illustration of the underlying states of the Susceptible-Infectious-Recovered Model(SIR).

The Basic Reproductive Number R_0

The basic reproductive number (R_0) represents the mean number of additional infections created by one infectious individual in a susceptible population. According to the latest available literature, without accounting for any social distancing policies the R_0 for COVID-19 is between 2 and 3.5 [57, 132, 72, 34]. R_0 can be expressed in terms of λ and μ as:

$$R_0 = \frac{\lambda}{\mu} \tag{6.5}$$

Extensions to the SEIR and SIR models

We use an extended version of the SEIR and SIR models of Dehning et al. [34] that incorporates some of the observed phenomena relating to COVID-19. First we include a delay D in becoming infected (I^{new}) and being reported in the confirmed case statistics, such that the confirmed reported cases CR_t at some time t are in the form [34] :

$$\text{CR}_t = I^{\text{new}}_{t-D} \tag{6.6}$$

We further assume that the spreading rate λ is time-varying rather than constant with change points that are affected by government interventions and voluntary social distancing measures.

6.2.2 Bayesian Parameter Inference

We follow the framework of Dehning et al. [34] to perform Bayesian inference for model parameters on the South African COVID-19 data. As previously discussed in this thesis, Bayesian framework allows for the posterior inference of parameters which updates prior beliefs based on a data-driven likelihood.

The Likelihood

The Likelihood indicates the probability of observing the reported case data given the assumed model. In our study, we adopt the Student-T distribution as the Likelihood as

suggested by Dehning et al. [34]. Similar to a Gaussian likelihood, the Student-T likelihood allows for parameter updates that minimise discrepancies between the predicted and observed reported cases.

Priors

Parameter prior distributions encode some prior subject matter knowledge into parameter estimation. In the case of epidemiological model parameters, priors incorporate literature based expected values of parameters such as recovery rate (μ), spreading rate (λ), change points based on policy interventions etc.

The prior settings for the model parameters are listed in Table 6.1. We follow Dehning et al. [34] by selecting LogNormal distributions for λ and σ such that the initial mean basic reproductive number is 3.2 which is consistent with literature [132, 72, 34, 43, 140]. We set a LogNormal prior for the σ such that the mean incubation period is five days. We use the history of government interventions to set priors on change points in the spreading rate. The priors on change-points include 19/03/2020 when a travel ban and school closures were announced, and 28/03/2020 when a national lockdown was enforced. We keep the priors for the Lognormal distributions of the spreading rates after the change points weakly-informative by setting the same mean as λ_0 and higher variances across all change points. This has the effect of placing greater weight on the data driven likelihood. Similar to Dehning et al. [34] we adopt "weakly-informative" Half-Cauchy priors for the initial conditions for the infected and exposed populations.

Table 6.1 Prior distribution settings for SEIR and SIR model parameters.

Parameter	Prior Distribution
Spreading rate λ_0	LogNormal(log(0.4),0.5)
Spreading rate λ_1	LogNormal(log(0.4),0.7)
Spreading rate λ_2	LogNormal(log(0.4),0.7)
Incubation to infectious rate σ	LogNormal(log(1/5),0.5)
Recovery rate μ	LogNormal(log(1/8),0.2)
Reporting Delay D	LogNormal(log(8),0.2)
Initial Infectious I_0	Half-Cauchy(20)
Initial Exposed E_0	Half-Cauchy(20)
Change Point t_1	Normal(2020/03/18,1)
Change Point t_2	Normal(2020/03/28,1)

We use the samplers described in Chapter 2 to calibrate the SEIR and SIR models on daily new cases and cumulative cases data for South Africa up to and including 20 April

2020 provided by Johns Hopkins University's Center for Systems Science and Engineering (CSSE) [36].

6.3 Results

SIR and SEIR model parameter inference was performed using confirmed cases data up to and including 20 April 2020 and MCMC samplers described in Chapter 2. Each of the samplers are run such that 5000 samples are drawn with 1000 burn-in and tuning steps. We use leave-one-out (LOO) cross-validation error of Vehtari et al. [127] to evaluate the goodness of fit of each model.

Table 6.2 shows the LOO validation errors of the various models. It can be seen that the SIR model with two change points as the best model fit with the lowest mean LOO of 447.91. The SEIR model with two change points showed a mean LOO of 453.82. We note that Dehning et al. [34] similarly find that the SIR model displayed superior goodness of fit to the SEIR on German data.

We now further present detailed results of the SIR and SEIR models with inference using HMC, the trace plots from these models indicating stationarity in the sampling chains are provided in appendix D figures D.2 and D.4. The trace plots for the SIR and SEIR models using MH are provided in figures D.3 and D.5. The trace plots largely indicate that the HMC sampler displays greater agreement between parallel chains.

Table 6.2 Leave-one out (LOO) Statistics comparing SEIR and SIR models with different number of change points.

Model	Change Points	LOO	Effective Parameters
SIR	2	447.91	10.36
SEIR	1	452.76	11.60
SEIR	0	453.47	15.90
SEIR	2	453.82	11.26
SIR	1	463.05	7.92
SIR	0	517.26	4.26

6.3.1 Posterior Parameter Distributions

Figure 6.3 shows the posterior distributions of the SIR model parameters. The parameter estimates are $\lambda_0 \approx 0.494$ (CI$[0.406, 0.594]$), $\lambda_1 \approx 0.098$ (CI$[0.063, 0.145]$), $\lambda_2 \approx 0.192$ (CI$[0.129, 0.256]$), $\mu \approx 0.149$ (CI$[0.096, 0.202]$) and reporting delay $(D) \approx 6.829$ (CI$[4.973, 8.596]$).

This corresponds to R_0 values of 3.315 (CI[2.940, 4.229]), 0.656 (CI[0.654, 0.673]) and 1.288 (CI[1.267, 1.343]) at the respective change points. Figure D.1 further shows the joint posterior distributions of λ_t and μ at each of the change points.

Time-varying spread rates allow for inference of the impact of various state and societal interventions on the spreading rate. Figure 6.4 shows the fit and projections based on SIR models with zero, one and two change points. As can be seen from the plot the two change point model best captures the trajectory in the development of new cases relative to the zero and one change point models. The superior goodness of fit of the two change point model is also illustrated in Table 6.2. The fit and projections showing similar behaviour on the SEIR model with various change points are shown in Figure 6.5.

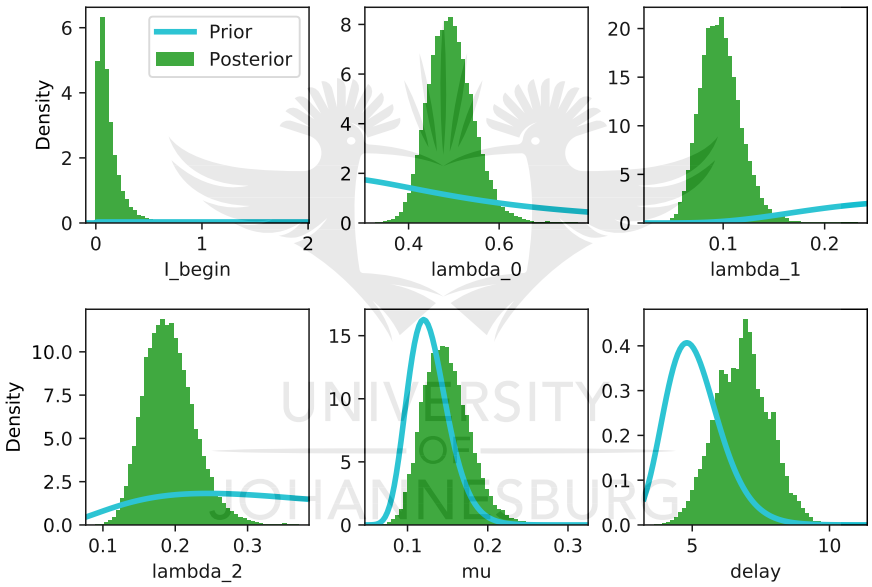


Figure 6.3 Posterior Parameter distributions for the SIR model with two change points.

6.3.2 Reporting Delays, Incubation and Infectious period

The mean reporting delay time in days was found to be 6.829 (CI[4.973, 8.596]), literature suggests this delay includes both the incubation period and the test reporting lags. The posterior distribution incubation period from the SEIR model in Figure 6.6 yields a median incubation period of 4.322 days (CI[2.395, 6.301]), Thus suggesting a mean laboratory reporting delay of approximately 2.507 days. A mean recovery rate $\mu \approx 0.151$ implies mean infectious period of 6.622 days which is in line with related literature [132, 72, 34].

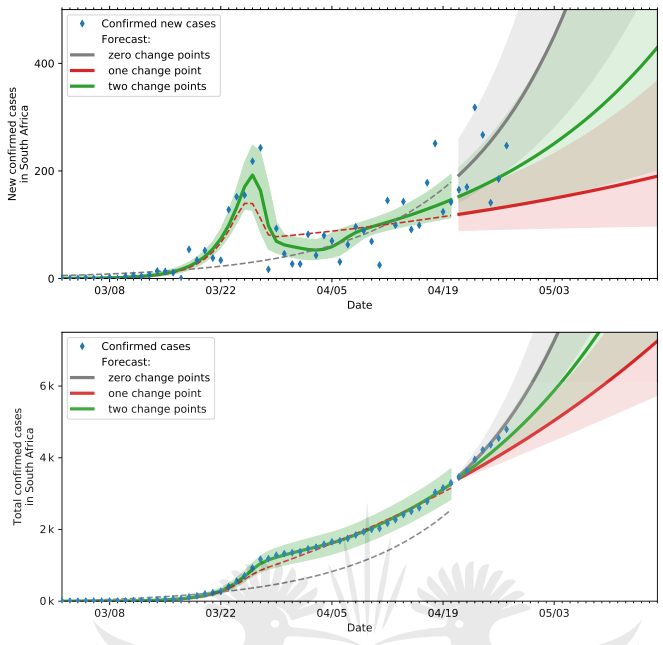


Figure 6.4 Predictions and actual data (until 20 April 2020) based on SIR models with various change points. The top plot indicates the actual and projected new cases while the bottom plot shows the actual and projected cumulative cases.

6.3.3 Timing and impact of interventions

Figure 6.7 depicts the posterior distributions of the spreading rates and times corresponding to each change point. We observe that the first change point is on a mean date of 18 March 2020 (CI:[16/03/2020, 20/03/2020]). This date is consistent with the travel ban, school closures and social distancing recommendations. This change point resulted in a substantial decrease in the spreading rate (80%) primarily due to the reduction in imported infections.

The second change point is observed on 28 March 2020 (CI:[26/03/2020, 30/03/2020]). This time point coincides with the announcement of mass screening and testing by the government on 30 March 2020. The resulting mean R_0 of 1.288 implies a 60% decrease from the initial value.

The inference of parameters is dependent on the underlying testing processes that generate the confirmed case data. The effect of the mass screening and testing campaign was to change the underlying confirmed case data generating process by widening the criteria of those eligible for testing. While initial testing focused on individuals that either had exposure to known cases or travelled to known COVID-19 affected countries, mass screening and

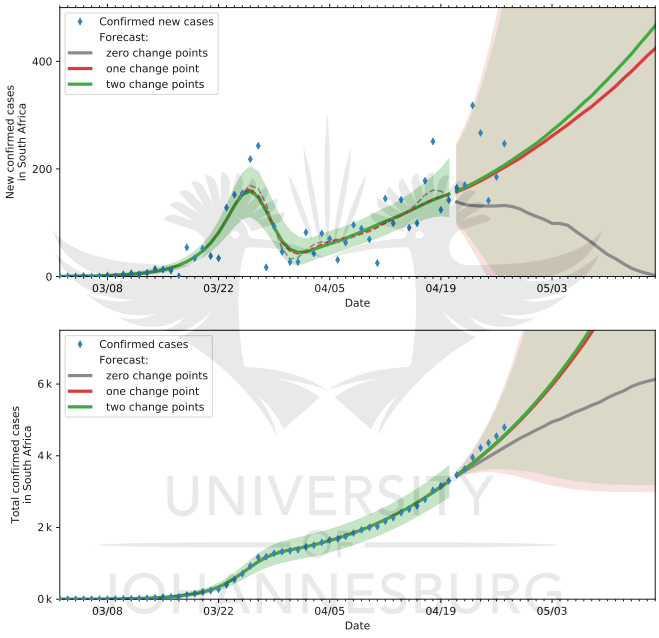


Figure 6.5 Predictions and actual data (until 20 April 2020) based on SEIR models with various change points. The top plot indicates the actual and projected new cases while the bottom plot shows the actual and projected cumulative cases.

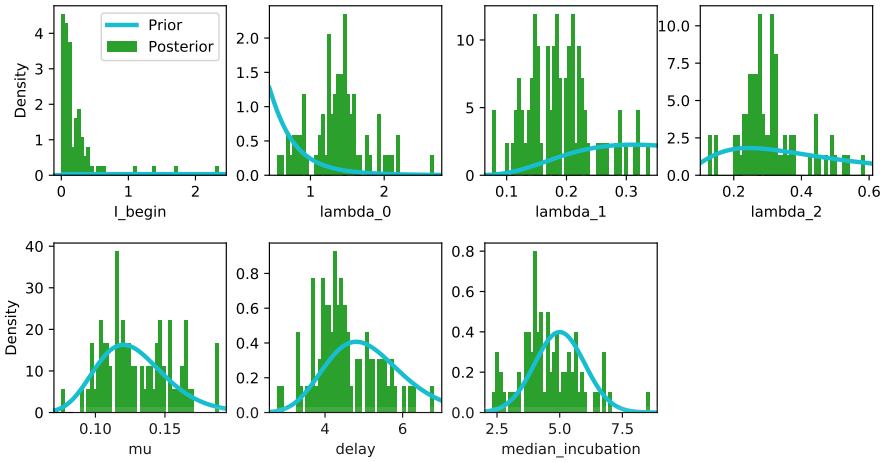


Figure 6.6 Posterior Parameter distributions under SEIR model with two change points.

testing further introduced detection of community level transmissions which may contain undocumented contact and exposure to COVID-19 positive individuals.

6.4 Discussion

We have performed Bayesian parameter inference of the SIR and SEIR models using MCMC and publicly available data as at 20 April 2020. The resulting parameter estimates fall in-line with the existing literature in-terms of mean baseline R_0 (before government action), mean incubation time and mean infectious period [34, 132, 43, 72].

We find that initial government action that mainly included a travel ban, school closures and stay-home orders resulted in a mean decline of 80% in the spreading rate. Further government action through mass screening and testing campaigns resulted in a second trajectory change point. This latter change point is mainly driven by the widening of the population eligible for testing, from travellers (and their known contacts) to include the generalised community who would have probably not afforded private lab testing which dominated the initial data. This resulted in an increase of R_0 to 1.288. The effect of mass screening and testing can also be seen in Figure 6.8 indicating a mean increase in daily tests preformed from 1639 to 4374.

The second change point illustrates the possible existence of "multiple pandemics", as suggested by Karim [67]. Thus testing after 28 March is more indicative of community-level transmissions that were possibly not as well documented in-terms of contact tracing and isolation relative to the initial imported infection driven pandemic. This is also supported

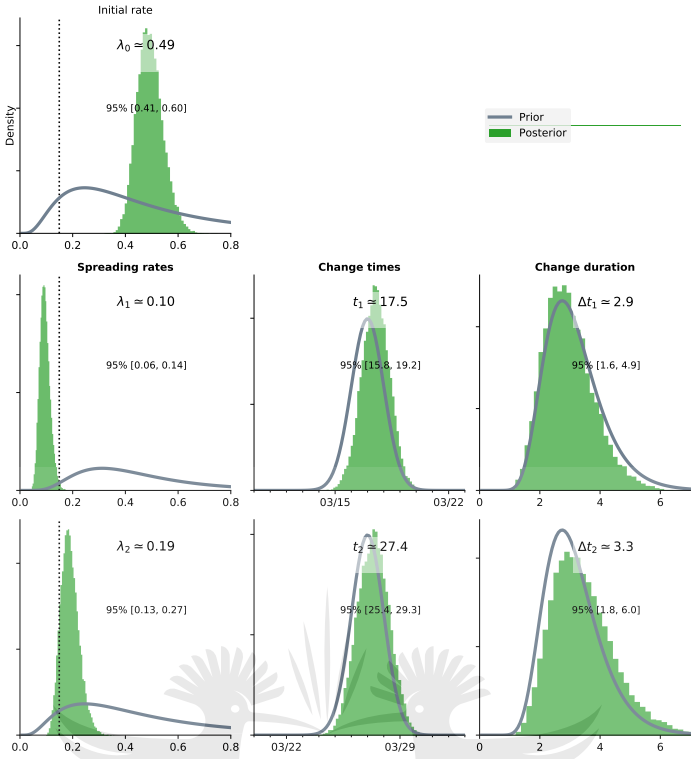


Figure 6.7 Posterior distributions of the spreading rates (λ_t) and the corresponding distributions of the time points.

by the documented increase in public laboratory testing (relative to private) past this change point, suggesting health care access might also play a role in the detection of community-level infections[1].

[1] Ministry of Health, Republic South Africa - Update on COVID-19 20th April 2020, https://sacoronavirus.co.za/2020/04/20/update-on-covid-19-20th-april-2020/

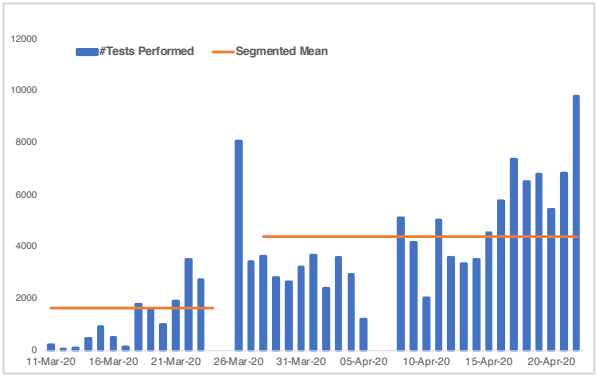


Figure 6.8 Daily COVID-19 tests performed in South Africa. The orange line indicates the segmented mean number of tests per day before and after the 28 March 2020 change point.

6.5 Conclusion

We have utilised a Bayesian inference framework to infer time-varying spreading rates of COVID-19 in South Africa. The time-varying spreading rates allow us to estimate the effects of government actions on the dynamics of the pandemic.

The results indicate a decrease in the mean spreading rate of 60%, which mainly coincides with the containment of imported infections, school closures and stay at home orders. The results also indicate the emergence of community-level infections which are increasingly being highlighted by the mass screening and testing campaign. The development of the community level transmissions ($R_0 \approx 1.288(CI[1.267,1.343])$) of the pandemic at the time of publication appears to be slower than that of the initial traveller based pandemic ($R_0 \approx 3.315(CI[2.940\ ,4.229])$).

Chapter 7

Conclusions and Future Research

7.1 Conclusion

In this thesis, we investigated probabilistic parameter inference in BNNs, GPs and compartmental pandemic forecasting models. The thesis showcases MCMC and GP based inference methods in the domains of credit prediction and wind speed forecasting, where the societal risk around incorrect predictions is high.

A modified equation-based S2HMC sampler is introduced for parameter inference in BNNs. S2HMC is found to yield higher effective sample sizes than the traditional HMC sampler. The predictive performance obtained via the two samplers is, however, found to be similar.

The S2HMC samples are augmented into a hierarchical ARD framework to include Gibbs sampling for hyperparameters in BNNs. A generalisable ARD committee approach is then introduced to add robustness to feature selections based on posterior variance estimates. This process of feature selection via a majority vote in the ARD committee is shown to select features with high information value. It is also shown that such an ARD committee based dimensional reduction improves the performance of uninformed samplers such as MH.

This thesis also makes contributions in scaling GPs through a PoE approach. Prediction calibration challenges due to the influence of weak experts are highlighted. Solutions based on Wasserstein barycenters and tempered softmax sparsity control are proposed. Empirical, experimental results show that such proposals outperform other PoE approaches in large-scale classification and regression tasks. In regression tasks these PoE proposals are shown to also outperform BNNs.

Finally, we propose a first in literature principled MCMC approach to change point determination in the spreading rates of COVID-19 in South Africa. This approach provides significant insights into the relative efficacy of various state-led public health policy interven-

tions during a period of high uncertainty. Similar studies in other jurisdictions reinforce the findings of this approach.

7.2 Future Work

As avenues for future research, other non-separable approaches to shadow HMC that are unexplored in BNN literature such as Mix & Match HMC of Radivojević and Akhmatskaya [102], Targeted Shadow HMC and Generalised Shadow HMC of Akhmatskaya and Reich [3, 4] can be presented alongside S2HMC given adequate computational resources. Variance reduction of these Shadow HMC methods using coupling also remains a possible area for future contributions [98, 90].

Investigations into improvements in computational efficiency of S2HMC to allow for speed up over HMC in BNNs can increase the attractiveness of the method. Such computational efficiencies could lead to greater adoption of MCMC methods in sampling larger scale deep NNs.

Importance samplers such as S2HMC facilitate the calculation of evidence via the importance weights [137]. The extension of S2HMC to evidence calculation can open up a multitude of applications in model selection problems.

Future work could include extensions to the proposed ARD committee feature selection approach to include other feature relevance metrics. A weighted committee voting scheme based on metrics such as predictive accuracy measures can also be explored. The inclusion of ARD committees into false discovery controlled feature selection paradigms such as the model-x knockoffs filter of Candes et al. [25] is already under investigation.

Extensions of the expert weighting methods presented in Chapter 5 to other experts based on other learning machines including BNNs can provide avenues to make the tempered softmax sparsity control framework adaptable to other probabilistic inference methods.

A broadening in the application scope of probabilistic methods is also of interest to include *inter alia* solar energy resource planning which suffers from similar intermittency issues to wind power [92]. Medicine, law enforcement and autonomous transportation are also application domains where such probabilistic techniques can yield significant benefit.

Future improvements to the COVID-19 inference work in Chapter 6 could include extensions to regional and provincial studies as current data suggests varied spreading rates both regionally and provincially. As more government interventions come to play priors on more change points might also be necessary.

Bibliography

[1] Abdel-Basset, M., Abdel-Fatah, L., and Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. In *Computational intelligence for multimedia big data on the cloud with engineering applications*, pages 185–231. Elsevier.

[2] Aguilar, E. and Brunet, M. (2001). *Seasonal Patterns of Air Surface Temperature and Pressure Change in Different Regions of Antarctica*, pages 215–228. Springer Berlin Heidelberg, Berlin, Heidelberg.

[3] Akhmatskaya, E. and Reich, S. (2006). The targeted shadowing hybrid monte carlo (tshmc) method. In *New Algorithms for Macromolecular Simulation*, pages 141–153. Springer.

[4] Akhmatskaya, E. and Reich, S. (2008). Gshmc: An efficient method for molecular simulation. *Journal of Computational Physics*, 227(10):4934–4954.

[5] Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive mcmc. *Statistics and computing*, 18(4):343–373.

[6] Angelini, E., di Tollo, G., and Roli, A. (2008). A neural network approach for credit risk evaluation. *The Quarterly Review of Economics and Finance*, 48(4):733 – 755.

[7] Antwarg, L., Shapira, B., and Rokach, L. (2019). Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

[8] Armaghani, D. J. and Asteris, P. G. (2020). A comparative study of ann and anfis models for the prediction of cement-based mortar materials compressive strength. *Neural Computing and Applications*, pages 1–32.

[9] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

[10] Baesens, B., Van Gestel, T., Viaene, S., STEPANOVA, M., Suykens, J., and Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54.

[11] Bai, C., Shi, B., Liu, F., and Sarkis, J. (2019). Banking credit worthiness: Evaluating the complex relationships. *Omega*, 83:26–38.

[12] Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. In *NeurIPS*.

[13] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637.

[14] Bertone, G., Deisenroth, M. P., Kim, J. S., Liem, S., Ruiz de Austri, R., and Welling, M. (2019). Accelerating the BSM interpretation of LHC data with machine learning. *Physics of the Dark Universe*.

[15] Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv: 1701.02434*.

[16] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[17] Bishop, C. M. and Svensén, M. (2003). Bayesian hierarchical mixtures of experts. In *UAI*.

[18] Blackwood, J. C. and Childs, L. M. (2018). An introduction to compartmental modeling for the budding infectious disease modeler. *Letters in Biomathematics*, 5(1):195–221.

[19] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

[20] Bonneel, N. and Pfister, H. (2013). Sliced Wasserstein barycenter of multiple densities. Technical Report TR-02-13, Havard University.

[21] Boulkaibet, I., Mthembu, L., Marwala, T., Friswell, M., and Adhikari, S. (2015). Finite element model updating using the shadow hybrid monte carlo technique. *Mechanical Systems and Signal Processing*, 52-53:115 – 132.

[22] Brauer, F. (2008). *Compartmental Models in Epidemiology*, pages 19–79. Springer Berlin Heidelberg, Berlin, Heidelberg.

[23] Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of markov chain monte carlo*. CRC press.

[24] Calderhead, B. (2011). *Differential geometric MCMC methods and applications*. PhD thesis, University of Glasgow.

[25] Candes, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold:'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577.

[26] Cao, Y. and Fleet, D. J. (2014). Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv:1410.7827*.

[27] Cao, Y. and Fleet, D. J. (2015). Transductive log opinion pool of Gaussian process experts. *arXiv:1511.07551*.

[28] Cárdenas, J. J., Garcia, A., Romeral, J., and Kampouropoulos, K. (2011). Evolutive anfis training for energy load profile forecast for an iems in an automated factory. In *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pages 1–8. IEEE.

[29] Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691.

[30] Cohen, S., Mbuvha, R., Marwala, T., and Deisenroth, M. (2020). Healing products of gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077. PMLR.

[31] Csató, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668.

[32] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

[33] Daniel, L. O., Sigauke, C., Chibaya, C., and Mbuvha, R. (2020). Short-term wind speed forecasting using statistical and machine learning methods. *Algorithms*, 13(6):132.

[34] Dehning, J., Zierenberg, J., Spitzner, F. P., Wibral, M., Neto, J. P., Wilczek, M., and Priesemann, V. (2020). Inferring covid-19 spreading rates and potential change points for case number forecasts. *arXiv preprint arXiv:2004.01105*.

[35] Deisenroth, M. P. and Ng, J. W. (2015). Distributed Gaussian processes. In *ICML*.

[36] Dong, E., Du, H., and Gardner, L. (2020). An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 20(5):533–534.

[37] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[38] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

[39] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

[40] Ernst, B., Oakleaf, B., Ahlstrom, M. L., Lange, M., Moehrlen, C., Lange, B., Focken, U., and Rohrig, K. (2007). Predicting the wind. *IEEE Power and Energy Magazine*, 5(6):78–89.

[41] Eseye, A., Zhang, J., Zheng, D., and Shiferaw, D. (2016). Short-term wind power forecasting using artificial neural networks for resource scheduling in microgrids. *International Journal of Science and Engineering Applications*, 5:144–151.

[42] Eseye, A. T., Zhang, J., Zheng, D., Ma, H., and Jingfu, G. (2017). A double-stage hierarchical anfis model for short-term wind power prediction. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 546–551.

[43] Ferguson, N., Laydon, D., Nedjati Gilani, G., Imai, N., Ainslie, K., Baguelin, M., Bhatia, S., Boonyasiri, A., Cucunuba Perez, Z., Cuomo-Dannenburg, G., et al. (2020). Report 9: Impact of non-pharmaceutical interventions (npis) to reduce covid19 mortality and healthcare demand.

[44] Fugon, L., Juban, J., and Kariniotakis, G. (2008). Data mining for wind power forecasting. In *European Wind Energy Conference & Exhibition EWEC 2008*. EWEC.

[45] Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*, 1(3).

[46] Gal, Y., van der Wilk, M., and Rasmussen, C. E. (2014). Distributed variational inference in sparse Gaussian process regression and latent variable models. In *NeurIPS*.

[47] Garcia-Chimeno, Y., Garcia-Zapirain, B., Gomez-Beldarrain, M., Fernandez-Ruanova, B., and Garcia-Monco, J. C. (2017). Automatic migraine classification via feature selection committee and machine learning techniques over imaging and questionnaire data. *BMC Medical Informatics and Decision Making*, 17(1):38.

[48] Garg, H. (2016). A hybrid pso-ga algorithm for constrained optimization problems. *Applied Mathematics and Computation*, 274:292 – 305.

[49] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT Press.

[50] Goodman, B. and Flaxman, S. (2016). Eu regulations on algorithmic decision-making and a right to explanation. In *ICML workshop on human interpretability in machine learning (WHI 2016), New York, NY. http://arxiv. org/abs/1606.08813 v1*.

[51] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *ICML*.

[52] Hahmann, A. N., Lennard, C., Badger, J., Vincent, C. L., Kelly, M. C., Volker, P. J., Argent, B., and Refslund, J. (2014). Mesoscale modeling for the wind atlas of south africa (wasa) project. *DTU Wind Energy*, 50:80.

[53] Hairer, E. (1999). Backward error analysis for multistep methods. *Numerische Mathematik*, 84(2):199–232.

[54] Hamori, S., Kawai, M., Kume, T., Murakami, Y., and Watanabe, C. (2018). Ensemble learning or deep learning? application to default risk analysis. *Journal of Risk and Financial Management*, 11(1):12.

[55] Hand, D. J. and Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541.

[56] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

[57] Hellewell, J., Abbott, S., Gimma, A., Bosse, N. I., Jarvis, C. I., Russell, T. W., Munday, J. D., Kucharski, A. J., Edmunds, W. J., working group, C. n., Funk, S., and Eggo, R. M. (2020). Feasibility of controlling 2019-ncov outbreaks by isolation of cases and contacts. *medRxiv*.

[58] Hensman, J., de G. Matthews, A. G., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In *AISTATS*.

[59] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *UAI*.

[60] Hinton, G. E. (1999). Products of experts. In *ICANN*.

[61] Hinton, G. E. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM.

[62] Hochberg, Y. and Tamhane, A. C. (2008). *Distribution-Free and Robust Procedures*, pages 234–273. John Wiley and Sons Inc.

[63] Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

[64] Izaguirre, J. A. and Hampton, S. S. (2004). Shadow hybrid monte carlo: an efficient propagator in phase space of macromolecules. *Journal of Computational Physics*, 200(2):581–604.

[65] Jang, J. S. R. (1993). Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

[66] Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., and Bennamoun, M. (2020). Hands-on bayesian neural networks–a tutorial for deep learning users. *arXiv preprint arXiv:2007.06823*.

[67] Karim, S. A. (2020). Sa's covid-19 pandemic trends and next steps. Presentation Prepared for Minister Zweli Mkhize.

[68] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[69] Lagazio, M. and Marwala, T. (2006). Assessing different bayesian neural network models for militarized interstate dispute: Outcomes and variable influences. *Social Science Computer Review*, 24(1):119–131.

[70] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[71] Liu, H., Cai, J., Wang, Y., and Ong, Y.-S. (2018). Generalized robust Bayesian committee machine for large-scale Gaussian process regression. *arXiv:1806.00720*.

[72] Lourenco, J., Paton, R., Ghafari, M., Kraemer, M., Thompson, C., Simmonds, P., Klenerman, P., and Gupta, S. (2020). Fundamental principles of epidemic spread highlight the immediate need for large-scale serological surveys to assess the stage of the sars-cov-2 epidemic.

[73] Lu, Y., Fan, Y., Lv, J., and Noble, W. S. (2018). Deeppink: reproducible feature selection in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 8676–8686.

[74] MacKay, D. J. (1992a). Bayesian model comparison and backprop nets. In *Advances in neural information processing systems*, pages 839–846.

[75] MacKay, D. J. (1995). Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505.

[76] MacKay, D. J. C. (1992b). A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472.

[77] Makarieva, A. M., Gorshkov, V. G., Sheil, D., Nobre, A. D., and Li, B.-L. (2013). Where do winds come from? a new theory on how water vapor condensation influences atmospheric pressure and dynamics. *Atmospheric Chemistry and Physics*, 13(2):1039–1056.

[78] Marivate, V., de Waal, A., Combrink, H., Lebogo, O., Moodley, S., Mtsweni, N., Rikhotso, V., Welsh, J., and Mkhondwane, S. (2020). Coronavirus disease (covid-19) case data-south africa. *Pretoria DrgatUo, ed2020*.

[79] Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

[80] Marwala, T. (2013a). *Economic modeling using artificial intelligence methods*. Springer.

[81] Marwala, T. (2013b). Flexibly-bounded rationality and marginalization of irrationality theories for decision making. *arXiv preprint arXiv:1306.2025*.

[82] Mbuvha, R. (2017). Bayesian neural networks for short term wind power forecasting. Master's thesis, KTH, School of Computer Science and Communication (CSC).

[83] Mbuvha, R., Boulkaibet, I., and Marwala, T. (2019a). Automatic relevance determination bayesian neural networks for credit card default modelling. *arXiv preprint arXiv:1906.06382*.

[84] Mbuvha, R., Boulkaibet, I., and Marwala, T. (2019b). Bayesian automatic relevance determination for feature selection in credit default modelling. In *International Conference on Artificial Neural Networks*, pages 420–425. Springer.

[85] Mbuvha, R., Boulkaibet, I., Marwala, T., and de Lima Neto, F. B. (2018). A hybrid ga-pso adaptive neuro-fuzzy inference system for short-term wind power prediction. In *International Conference on Swarm Intelligence*, pages 498–506. Springer.

[86] Mbuvha, R., Jonsson, M., Ehn, N., and Herman, P. (2017). Bayesian neural networks for one-hour ahead wind power forecasting. In *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 591–596. IEEE.

[87] Mbuvha, R. and Marwala, T. (2020a). Bayesian inference of covid-19 spreading rates in south africa. *PloS one*, 15(8):e0237126.

[88] Mbuvha, R. and Marwala, T. (2020b). On data-driven management of the covid-19 outbreak in south africa. *medRxiv*.

[89] McKight, P. E. and Najab, J. (2010). Kruskal-wallis test. *Corsini Encyclopedia of Psychology*.

[90] Mongwe, Wilson, T., Mbuvha, R., and Marwala, T. (2021). Antithetic magnetic and shadow hamiltonian monte carlo. *IEEE Access*.

[91] Morgan, J. and David, M. (1963). Education and income. *The Quarterly Journal of Economics*, 77(3):423–437.

[92] Mutavhatsindi, T., Sigauke, C., and Mbuvha, R. (2020). Forecasting hourly global horizontal solar irradiance in south africa using machine learning models. *IEEE Access*, 8:198872–198885.

[93] Neal, R. M. (1993). Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482.

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

[95] Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2.

[96] Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547.

[97] Pehlivanlı, A. Ç., Aşıkgil, B., and Gülay, G. (2016). Indicator selection with committee decision of filter methods for stock market price trend in ise. *Applied Soft Computing*, 49:792–800.

[98] Piponi, D., Hoffman, M., and Sountsov, P. (2020). Hamiltonian monte carlo swindles. In *International Conference on Artificial Intelligence and Statistics*, pages 3774–3783. PMLR.

[99] Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*.

[100] Potter, C. W. and Negnevitsky, M. (2006). Very short-term wind forecasting for tasmanian power generation. *IEEE Transactions on Power Systems*, 21(2):965–972.

[101] Quiñonero Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*.

[102] Radivojević, T. and Akhmatskaya, E. (2020). Modified hamiltonian monte carlo for bayesian inference. *Statistics and Computing*, 30(2):377–404.

[103] Rasmussen, C. E. and Ghahramani, Z. (2001). Infinite mixtures of Gaussian process experts. In *NeurIPS*.

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

[105] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[106] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[107] Rullière, D., Durrande, N., Bachoc, F., and Chevalier, C. (2018). Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing*, 28(4):849–867.

[108] Şahin, M. and Erol, R. (2017). A comparative study of neural networks and anfis for forecasting attendance rate of soccer games. *Mathematical and Computational Applications*, 22(4):43.

[109] Sariannidis, N., Papadakis, S., Garefalakis, A., Lemonakis, C., and Kyriaki-Argyro, T. (2020). Default avoidance on credit card portfolios using accounting, demographical and exploratory factors: decision making based on machine learning (ml) techniques. *Annals of Operations Research*, 294(1):715–739.

[110] Sibisi, S. (1989). *Regularization and Inverse Problems*, pages 389–396. Springer Netherlands, Dordrecht.

[111] Sideratos, G. and Hatziargyriou, N. (2007). Using radial basis neural networks to estimate wind power production. In *2007 IEEE Power Engineering Society General Meeting*, pages 1–7.

[112] Skeel, R. D. and Hardy, D. J. (2001). Practical construction of modified hamiltonians. *SIAM Journal on Scientific Computing*, 23(4):1172–1188.

[113] Skold, M. (2015). Computer intensive statistical methods. Lecture notes for FMS091/MAS221.

[114] Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *NeurIPS*.

[115] Sobhani, J., Najimi, M., Pourkhorshidi, A. R., and Parhizkar, T. (2010). Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and anfis models. *Construction and Building Materials*, 24(5):709–718.

[116] Song, P. X., Wang, L., Zhou, Y., He, J., Zhu, B., Wang, F., Tang, L., and Eisenberg, M. (2020). An epidemiological forecast model and software assessing interventions on covid-19 epidemic in china.

[117] Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In *AISTATS*.

[118] Sun, T. and Vasarhelyi, M. A. (2018). Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management*, 25(4):174–189.

[119] Sweet, C. R., Hampton, S. S., Skeel, R. D., and Izaguirre, J. A. (2009). A separable shadow hamiltonian hybrid monte carlo method. *The Journal of chemical physics*, 131(17):174106.

[120] Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*.

[121] Trapp, M., Peharz, R., Pernkopf, F., and Rasmussen, C. E. (2019). Deep structured mixtures of Gaussian processes. In *AISTATS*.

[122] Tresp, V. (2000a). A Bayesian committee machine. *Neural Computation*.

[123] Tresp, V. (2000b). Mixtures of Gaussian processes. In *NeurIPS*.

[124] Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications*, 37(4):3326–3336.

[125] UNFCCC (2015). Historic paris agreement on climate change. *United Nations Framework Convention on Climate Change (UNFCCC)*.

[126] Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337.

[127] Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432.

[128] Villani, C. (2008). *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media.

[129] Wang, G., Hao, J., Ma, J., and Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Syst. Appl.*, 38(1):223–230.

[130] Wang, K. A., Pleiss, G., Gardner, J. R., Weinberger, K. Q., and Wilson, A. G. (2019). Exact Gaussian processes on a million data points. In *NeurIPS*.

[131] Wang, Z., Mohamed, S., and Freitas, N. (2013). Adaptive hamiltonian and riemann manifold monte carlo. In *International Conference on Machine Learning*, pages 1462–1470.

[132] WHO (2020). Report of the who-china joint mission on coronavirus disease 2019 (covid-19).

[133] Wilson, A. G. and Nickisch, H. (2015). Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *ICML*.

[134] Xia, Y., Liu, C., Li, Y., and Liu, N. (2017). A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225 – 241.

[135] Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

[136] Yu, S., Wei, Y.-M., and Wang, K. (2012). A pso–ga optimal model to estimate primary energy demand of china. *Energy Policy*, 42:329 – 340.

[137] Yuan, C. and Druzdzel, M. J. (2006). Importance sampling algorithms for bayesian networks: Principles and performance. *Mathematical and Computer Modelling*, 43(9-10):1189–1207.

[138] Zamani, A., Sorbi, M. R., and Safavi, A. A. (2013). Application of neural network and anfis model for earthquake occurrence in iran. *Earth Science Informatics*, 6(2):71–85.

[139] Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for Gaussian processes. *Journal of Machine Learning Research*.

[140] Zhou, C., Yuan, W., Wang, J., Xu, H., Jiang, Y., Wang, X., Wen, Q. H., and Zhang, P. (2020). Detecting suspected epidemic cases using trajectory big data.

[141] Álvarez Esteban, P. C., del Barrio, E., Cuesta-Albertos, J. A., and Matrán, C. (2016). A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*.

Appendix A

Adaptive Neuro-Fuzzy inference systems

A.1 Adaptive Neuro-Fuzzy inference systems

Adaptive Neuro-Fuzzy inference system (ANFIS) is a class of the fuzzy Inference Systems (FIS) that adaptively adjust membership functions and consequent parameters based on training data. Figure A.1 shows an ANFIS architecture as proposed by Jang [65]. This method is established by five consecutive layers that sequentially process the information from inputs towards outputs. These five layers operate as follows:

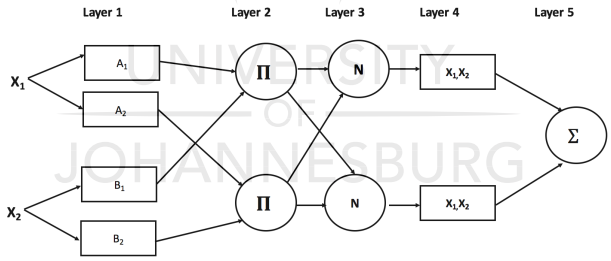


Figure A.1 Simple ANFIS architecture with two inputs and two rules.

Layer 1 is a fuzzification layer, where crisp inputs are converted into fuzzy set membership values. This is done using membership functions (MFs) which are bounded in range the [0,1]. The output of the j^{th} node in this layer will be of the form:

$$O_j^1 = \mu_{A_j}(x) \quad \text{j=1,2} \tag{A.1}$$

where $\mu_{A_j}(x)$ is the MF. In this work, a Gaussian MF, as described in equation A.2, are selected for the modelling process.

$$\mu_{A_j}(x) = \exp - \left(\frac{(x - p_j)}{\alpha_j} \right) \tag{A.2}$$

Layer 2 combines the incoming signals from the the fuzzy sets in the previous layer using a T-norm operator. The result of this operation is the combined firing strength of each rule. If the chosen T-norm operator is multiplication then the output of the j^{th} node in this layer is:

$$O_j^2 = w_j = \mu_{A_j}(x) \times \mu_{A_j}(x) \quad \text{j=1,2} \tag{A.3}$$

Layer 3 is a normalisation node where the relative firing strength of each rule is calculated as ratio of its firing strength w_j to the sum of the firing strengths of all rules. The normalised firing strength of the j^{th} in this layer will be:

$$O_j^3 = \bar{w}_j = \frac{w_j}{w_1 + w_2} \quad \text{j=1,2} \tag{A.4}$$

Layer 4 calculates the consequent part of a Tagaki-Sugeno type FIS. The result is a linear combination of the inputs for each rule weighted by its respective normalised firing strength $\bar{w}_j$. This weighted linear combination is of the form:

$$O_j^4 = \bar{w}_j f_j = \bar{w}_j (a_j x_1 + b_j x_2 + c_j) \tag{A.5}$$

where a_j, b_j, c_j are unknown consequent parameters

Layer 5 performs an aggregation of the consequent values evaluated in the previous layer as a weighted average. The final output is therefore:

$$O_j^5 = \bar{w}_j = \sum_i \bar{w}_j f_i \tag{A.6}$$

A.2 Parameter Settings for ANFIS Training

A population size of 40 is used of the GA, GAPSO and GAPSO-I. Table A.1 shows a list of the additional parameters settings.

Table A.1 List of additional parameters

	Parameter	Value
GAPSO/GAPSO-I	Inertia Weight	1
	Personal Learning Coefficient	1.6
	Global Learning Coefficient	2
	GAPSO random number Threshold (T)	0.75

A.3 Norwegian wind farm dataset

The Norwegian wind farm dataset consists of 7384 records covering the period from January 2014 to December 2016 [82]. The dataset features include the windfarm online capacity, one and two hour lagged historical power production values as well as NWP estimates of humidity, temperature and wind speed.

Table A.2 Input variables used for model training.

Feature name	Source	Time Delay
Online Capacity (%)	SCADA	1
Online Capacity (%)	SCADA	2
Power Production	SCADA	1
Power Production	SCADA	2
Relative Humidity	NWP	1
Temperature	NWP	-1
Temperature	NWP	0
Wind speed	NWP	-1
Wind speed	NWP	1

Appendix B

Gaussian Approximation and HMC Approaches to ARD

B.1 Gaussian Approximation to the Posterior

MacKay [75] proposed a Gaussian Approximation to the posterior based on a second order Taylor Expansion of the posterior around MAP estimate $\mathbf{w_{MP}}$ as follows [82, 75]:

$$P(\mathbf{w}|\alpha,\beta,H,D) \approx \frac{1}{Z'_M(\alpha,\beta)} \exp\left(-\left(E_W(\mathbf{w_{MP}}) + E_D(\mathbf{w_{MP}})\right) - \frac{1}{2}(\mathbf{w}-\mathbf{w_{MP}})^T \mathbf{A} (\mathbf{w}-\mathbf{w_{MP}}) \right) \tag{B.1}$$

The matrix $\mathbf{A}$ is defined by the weighted sum of the second derivatives of E_W and E_D with respect to the model weights as follows [82, 75]:

$$-\bigtriangledown\bigtriangledown log P(\mathbf{w}|\alpha,\beta,H,D) = \bigtriangledown\bigtriangledown M(\mathbf{w}) \tag{B.2}$$

$$= \alpha\mathbf{I} + \beta\mathbf{H} \tag{B.3}$$

$\mathbf{H}$ contains the second derivatives of the network error with respect to the weights and is known as the Hessian. The normalizing constant Z' is now a Gaussian Integral which can be evaluated using the functional form of the Multivariate Gaussian as [82]:

$$Z' = \int exp\left(-\left(\alpha E_W(\mathbf{w_{MP}}) + \beta E_D(\mathbf{w_{MP}})\right) - \frac{1}{2}(\mathbf{w}-\mathbf{w_{MP}})^T \mathbf{A} (\mathbf{w}-\mathbf{w_{MP}}) \right) d\mathbf{w} \tag{B.4}$$

$$= exp\left(-\left(\alpha E_W(\mathbf{w_{MP}}) + \beta E_D(\mathbf{w_{MP}})\right) \right) (2\pi)^{\frac{-k}{2}} det(\mathbf{A})^{\frac{-1}{2}} \tag{B.5}$$

where k is the number of weights in the network.

B.2 Hyperparameter Estimation

The equations above assume that the hyperparameters α and β are fixed. These hyperparameters can be estimated using the evidence framework [82, 75]. The evidence $P(D|\alpha,\beta,H)$ as defined in equation 2.1 is the marginal distribution of the data given the model can be obtained by marginalising over the weights. Re-arranging the terms in equation 2.1 it can be seen that the evidence can be expressed in terms ratios of the normalizing constants of posterior, prior and the likelihood. The log evidence given the Gaussian approximation in equation B.1 and corresponding normalizing constant Z' in equation B.5 can therefore be written down as:

$$log P(D|\alpha,\beta,\mathcal{H}) = log \frac{Z'}{Z_W(\alpha)Z_D(\beta)} \tag{B.6}$$

$$= -\alpha E_W(\mathbf{w_{MP}}) - \beta E_D(\mathbf{w_{MP}}) - \frac{1}{2} det\left(\frac{\mathbf{A}}{(2\pi)^k}\right) - log Z_W(\alpha) - log Z_D(\beta) \tag{B.7}$$

$$= -\alpha E_W(\mathbf{w_{MP}}) - \beta E_D(\mathbf{w_{MP}}) - \frac{1}{2} det(\mathbf{A}) + \frac{k}{2} log(2\pi) - log Z_W(\alpha) - log Z_D(\beta) \tag{B.8}$$

where $\beta E_D(\mathbf{w_{MP}})$ represents the likelihood's contribution to the evidence [82], while the terms $-\alpha E_W(\mathbf{w_{MP}}) - \frac{1}{2} det(\mathbf{A}) - log Z_W(\alpha)$ are referred to as the log 'Occam factor' which gives low evidence to small values of α [82].

The optimal α and β values can be obtained by maximising the log evidence. When differentiating with respect to α we need to evaluate $\frac{\partial \log\det(\mathbf{A})}{\partial \alpha}$ using Jacobi's formula and equation B.1 as [82]:

$$\begin{aligned} \frac{\partial \log\det(\mathbf{A})}{\partial \alpha} &= \text{Trace}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \alpha}\right) \\ &= \text{Trace}\left(\mathbf{A}^{-1}\mathbf{I}\right) \\ &= \text{Trace}\left(\mathbf{A}^{-1}\right) \end{aligned}$$

Setting the derivative to zero results in the following expression in terms of α [82, 75]:

$$2\alpha E_W(\mathbf{w_{MP}}) = k - \alpha \ \text{Trace}(A^{-1}) = \gamma \tag{B.9}$$

This quantity is referred to as the true number of effective parameters γ which ranges from 0 to k. Using the definition of E_W in equation 2.5, the update equation for α becomes [82, 75]:

$$\alpha_{MP} = \frac{\gamma}{\mathbf{w}_{MP}^T \mathbf{w}_{MP}} \tag{B.10}$$

Similarly by differentiating the log evidence in terms of β and setting the derivative to zero, we obtain the following expression in terms of β:

$$2\beta E_D^{MP} = N - \gamma \tag{B.11}$$

Using the definition of E_D in equation 2.3 the update equation for β becomes:

$$\beta_{MP} = \frac{N - \gamma}{\sum_{i=1}^{N} \left(t^{(i)} - y(X^{(i)}; \mathbf{w})\right)^2} \tag{B.12}$$

In the training procedure of such BNNs we therefore have to alternate between optimizing the hyperparameters (equations B.10, B.12) and optimizing the posterior distribution (equation 1.9). The alogrithm below shows this training procedure.

Data: Training dataset $\{\mathbf{X}^{(i)}, \mathbf{t}^{(i)}\}$
Result: Trained Network Weights $\mathbf{w}_{MP}$ and tuned hyperparameters α, β
begin

1. Randomly initialize hyperparameters α and β
2. Optimize the loss function $\alpha E_W(\mathbf{w}) + \beta E_D(\mathbf{w})$ for $\mathbf{w}_{MP}$ given α and β using an optimizer of choice e.g. gradient descent in equation 1.6
3. Use the evidence framework to estimate hyperparameters using equations B.10 and B.12
4. 2 and 3 until convergence or stopping criterion

end

B.2.1 Predictive Distribution

If the posterior is approximated by the Gaussian as in equation B.1 and the network output $y(x^{N+1}, \mathbf{w}_{MP})$ can be linearized by a first order Taylor expansion around $\mathbf{w}_{MP}$ as [75, 82]:

$$y(\mathbf{x}^{N+1}, \mathbf{w}) \simeq y(\mathbf{x}^{N+1}, \mathbf{w}_{MP}) + \mathbf{g}(\mathbf{w} - \mathbf{w}_{MP}) \tag{B.13}$$

where $\mathbf{g}$ is derivative of the network output with respect to the weights evaluated at the new input and optimal weights

$$\mathbf{g} = \frac{\partial y}{\partial \mathbf{w}}\bigg|_{\mathbf{x}^{N+1}, \mathbf{w}_{MP}}$$

meaning therefore from our assumed noise model in equation 2.3:

$$P(t^{(N+1)}|\mathbf{w}, \beta, \mathcal{H}) \simeq \mathcal{N}(y(\mathbf{x}^{N+1}, \mathbf{w}_{MP}) + \mathbf{g}(\mathbf{w} - \mathbf{w}_{MP}), \beta^{-1}) \tag{B.14}$$

B.2.2 Automatic Relevance Determination

The formulation above assumes that all network weights have the same prior. However in principle weights can come from distinct groups with a unique regularization parameter for each class α_c. The evidence framework for hyperparameter estimation applies as before with [82]:

$$\alpha_c^{MP} = \frac{\gamma_c}{\mathbf{w}_{MP}^T \mathbf{w}_{MP}} \quad \text{where } \mathbf{w}_{MP} \in c \tag{B.15}$$

and

$$\gamma_c = k_c - \alpha_c \text{Trace}_c(\mathbf{A}^{-1}) \tag{B.16}$$

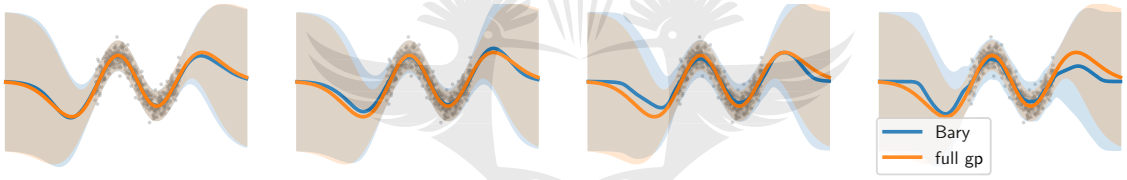
B.3 HMC with Gibbs Sampling Algorithm

HMC with alternating Gibbs sampling depicted by the algorithm below.

Data: Dataset $\{\mathbf{X}, \mathbf{y}\}$
Result: N samples of model parameters $\mathbf{w}$ and $\frac{N}{n_{\text{Gibbs}}}$ samples of parameters α_c.
initialise the network weights $\mathbf{w}$ **for** $n \leftarrow 1$ **to** N **do**
  if $mod(n, n_{Gibbs}) = 0$ **then**
    Sample hyper-parameters α_c from Gamma$(\tau + N_c, \theta + E_{W_c})$
  end
  sample the auxiliary momentum variables $\mathbf{p}$
  $\mathbf{p} \sim \mathcal{N}(0, \mathbf{M})$
  Use leapfrog steps to generate proposals for $\mathbf{w}$
  for $t \leftarrow 1$ **to** L **do**
    $\mathbf{p}(t + \varepsilon/2) \leftarrow \mathbf{p}(t) + (\varepsilon/2)\frac{\partial H}{\partial \mathbf{w}}\big(\mathbf{w}(t)\big)$
    $\mathbf{w}(t + \varepsilon) \leftarrow \mathbf{w}(t) + \varepsilon \frac{\mathbf{p}(t+\varepsilon/2)}{\mathbf{M}}$
    $\mathbf{p}(t + \varepsilon) \leftarrow \mathbf{p}(t + \varepsilon/2) + (\varepsilon/2)\frac{\partial H}{\partial w}\big(\mathbf{w}(t + \varepsilon)\big)$
  end
  Metropolis Update step:
  $(\mathbf{p}, \mathbf{w})_n \leftarrow (\mathbf{p}(L), \mathbf{w}(L))$ with probability:
  $min\big(1, \frac{P(\mathbf{w}(L)|D,H)}{P(\mathbf{w}_{(n-1)}|D,H)}\big)$
end

Appendix C

Products of Gaussian Processes Appendix



(a) soft-var, 100 pts/exp (b) soft-var, 20 pts/exp (c) soft-var, 4 pts/exp (d) soft-var, 2 pts/exp

Figure C.1 Full GP baseline (orange) and barycenter of GPs model (blue) trained on synthetic data with a decreasing number of points per experts (Left to Right), using softmax-variance weighting [30].

Dataset	BAR_var	BCM	gPoE_unif	gPoE_var	PoE	rBCM_diff_entr	rBCM_var
WM01 Alexander Bay	0.540 (1.605)	0.556 (1.620)	0.546 (1.618)	0.538 (1.602)	0.555 (1.618	0.554 (1.617)	0.551 (1.648)
WM05 Napier	0.414 (1.502)	0.424 (1.508)	0.418 (1.510)	0.413 (1.500)	0.426 (1.510)	0.424 (1.508)	0.424 (1.554)
WM13 Jozini	0.752 (1.426)	0.767 (1.435)	0.756 (1.437)	0.751 (1.425)	0.769 (1.437)	0.767 (1.434)	0.760 (1.572)

Table C.1 Average NLPD (RMSE) on the three weather stations for the regression datasets using random partitioning

	BAR_var	BCM	gPoE_unif	gPoE_var	PoE	rBCM_diff_entr	rBCM_var	$SVGP_{500}$
Top-1-accur.	0.911	0.894	0.894	0.910	0.894	0.895	0.396	0.862
Top-2-accur.	0.964	0.955	0.954	0.964	0.955	0.956	0.411	0.939
Top-3-accur.	0.981	0.976	0.975	0.982	0.976	0.976	0.418	0.967
NLPD	0.312	0.853	0.384	0.313	0.851	0.879	2.474	0.497

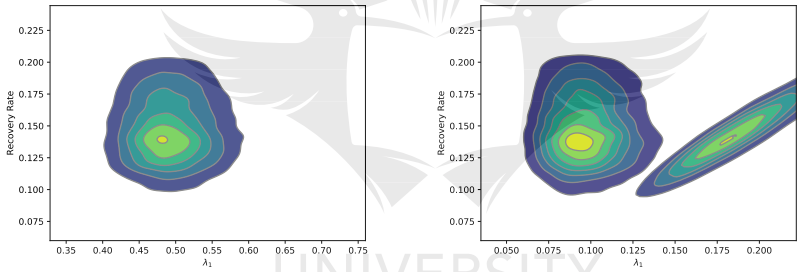
Table C.2 Top-*n* accuracy and NLPDs on the MNIST dataset (PCA features) using clustering partitioning.

	BAR_var	BCM	gPoE_unif	gPoE_var	PoE	rBCM_diff_entr	rBCM_var	$SVGP_{500}$
Top-1-accur.	0.821	0.822	0.822	0.820	0.822	0.822	0.820	0.818
NLPD	0.812	1.181	0.843	0.775	1.176	1.172	0.981	0.709

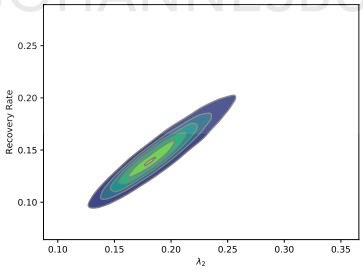
Table C.3 Top-1 accuracy and NLPDs on the Taiwan credit dataset using clustering partitioning.

Appendix D

Additional Information: COVID-19 Inference



(a) λ_0 vs μ (b) λ_1 vs μ



(c) λ_2 vs μ

Figure D.1 Two dimensional heat maps of the posterior distributions of the spreading rate (λ) and the recovery rate (μ) at various change points of the SIR model. The high joint density areas (in yellow) indicate likely values of R_0. The baseline mean R_0 estimate in D.1(a) is 3.315, the first change point estimate in figure D.1(b) is 0.657 while the second change point in figure D.1(c) has resulted in a mean R_0 estimate of 1.288.

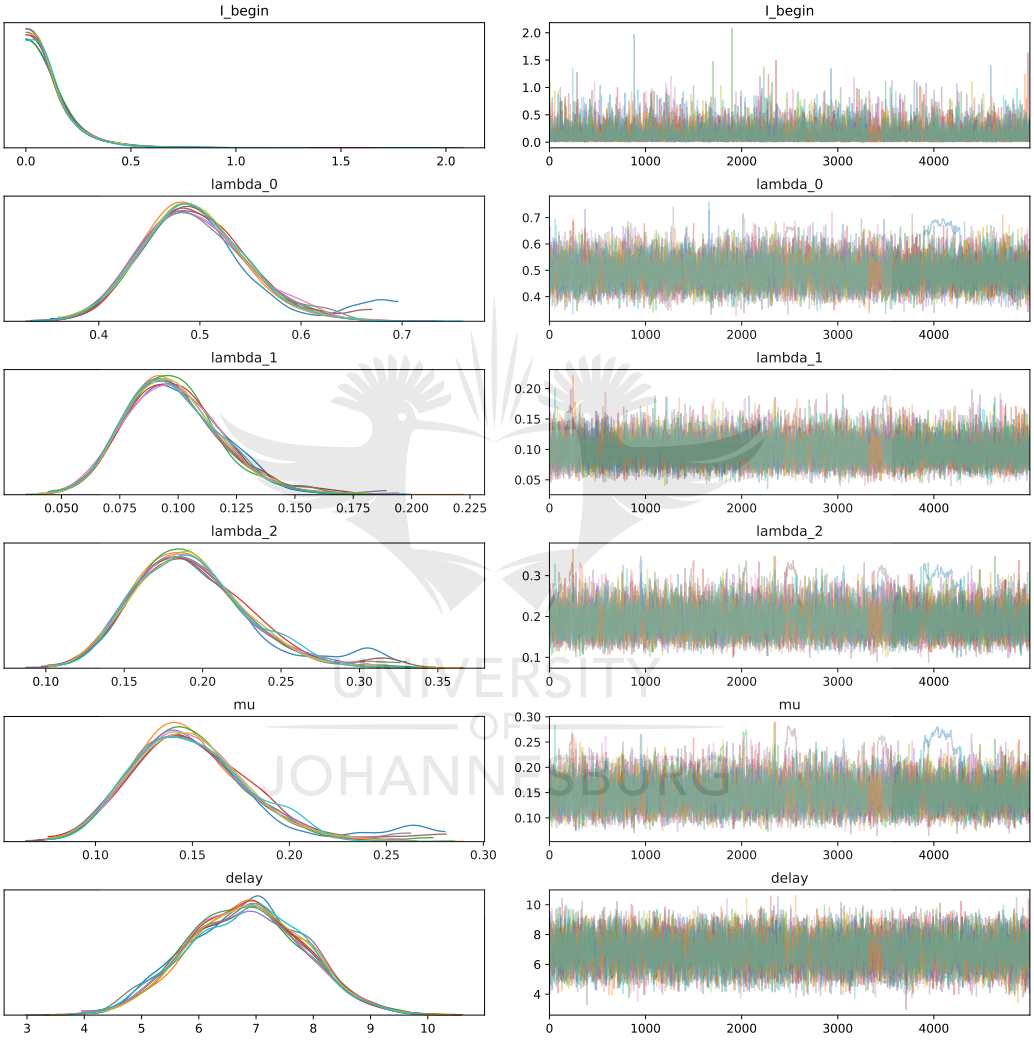


Figure D.2 Diagnostic trace plots for the SIR model inferred using HMC.

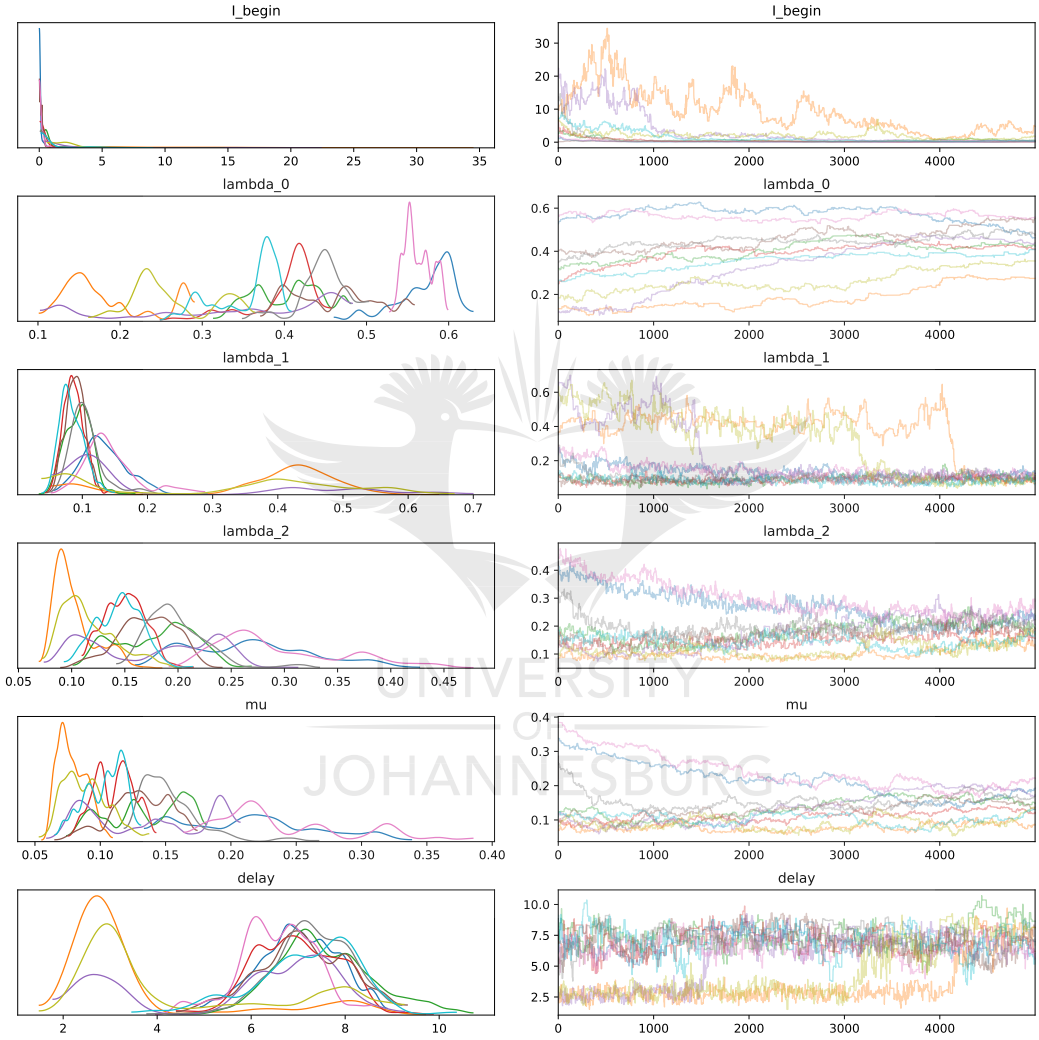


Figure D.3 Diagnostic trace plots for the SIR model inferred using MH.

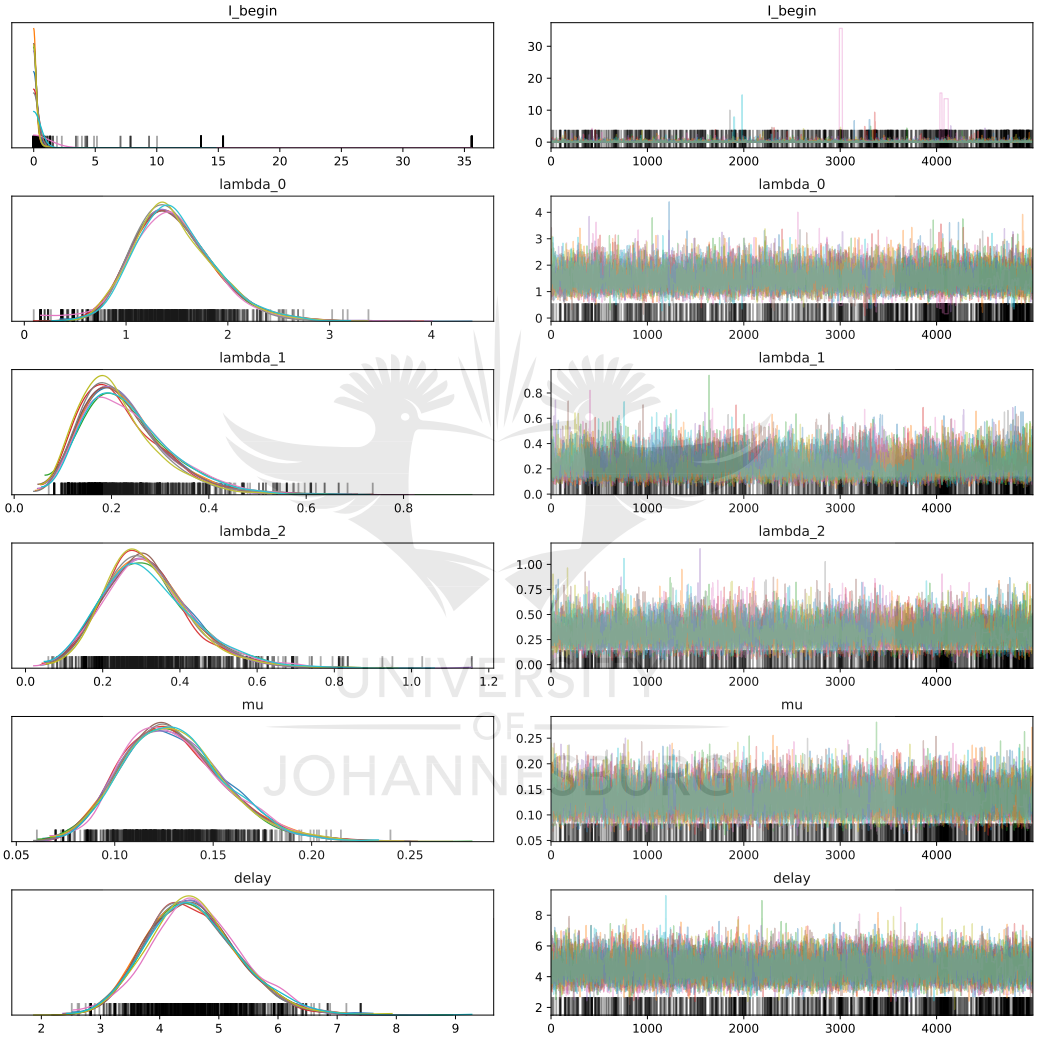


Figure D.4 Diagnostic trace plots for the SEIR model inferred using HMC.

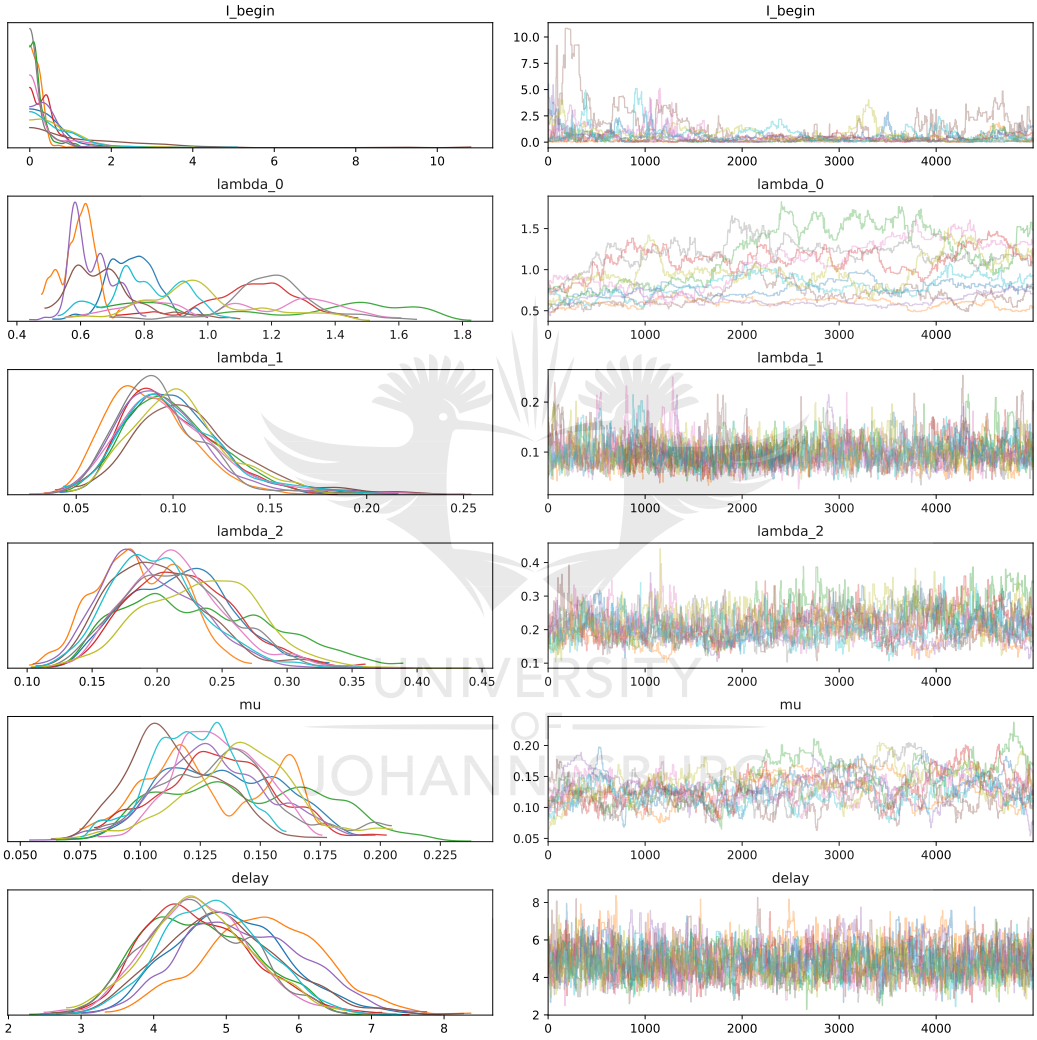


Figure D.5 Diagnostic trace plots for the SEIR model inferred using MH.